

Erkki Mäkinen (toim.)

**Tietojenkäsittelytieteellisiä
tutkielmia
Syksy 2009**



TIETOJENKÄSITTELYTIETEIDEN LAITOS
TAMPEREEN YLIOPISTO

D-2010-1

TAMPERE 2010

TAMPEREEN YLIOPISTO
TIETOJENKÄSITTELYTIETEIDEN LAITOS
JULKAISUSARJA D – VERKKOJULKAISUT
D-2010-1, TAMMIKUU 2010

Erkki Mäkinen (toim.)

**Tietojenkäsittelytieteellisiä tutkielmia
Syksy 2009**

TIETOJENKÄSITTELYTIETEIDEN LAITOS
33014 TAMPEREEN YLIOPISTO

ISBN 978-951-44-7981-6
ISSN 1795-4274

Sisällysluettelo

Hiukkasparviontimointi ja kauppatuottajan ongelma	1
<i>Ville Autio</i>	
Laskennallinen näkökulma Riemannin zeta-funktioon.....	24
<i>Henry Joutsijoki</i>	
Kasvojen tunnistukseen perustuvan mobiilikäyttöliittymän arviointi.....	43
<i>Kari Jussila</i>	
Sosiaalinen media opetuskäytössä.....	55
<i>Maiju Karhunen</i>	
Mainossoikeus – katsaus tutkimustuloksiin	67
<i>Tony Keisala</i>	
Assosiaatiosäännöt ja louhittujen sääntöjoukkojen karsiminen	80
<i>Sami Kiviharju</i>	
GATE Toolkit esimerkkinä nykyaikaisesta luonnollisen kielen käsittelyohjelmasta	95
<i>Salla Kuisma</i>	
Älykkäät agentit oppimisen apuvälineenä.....	115
<i>Karoliina Käki</i>	
Sähköisen vaalijärjestelmän kokeilusta kunnallisvaaleissa 2008 äänestäjän tietoturvan näkökulmasta.....	130
<i>Jorma Laajamäki</i>	
Virtuaalinen tuotekokemus.....	147
<i>Taina Lempiäinen</i>	
Second Life liiketoiminnallisena toimintaympäristönä	160
<i>Mervi Ollikainen</i>	
Kohti mukautuvia käyttöliittymiä	172
<i>Pasi Paunu</i>	
Mediakasvatus peruskoulussa.....	196
<i>Jenni Päckilä</i>	

Ajax ja sen vaikutukset web-sovelluksen käytettävyyteen.....	207
<i>Tommi Ritola</i>	
Massiivisesti hajautetut avain-arvo -tietokannat.....	217
<i>Timo Sirainen</i>	
Päätöspuut hahmontunnistuksessa	227
<i>Nina Tyni</i>	
Muurahaisyhdyskuntaoptimointi.....	245
<i>Ilkka Virolainen</i>	
Social Human-Robot Interaction: Review of Existing Literature	266
<i>Jakub Zlotowski</i>	

Hiukkasparviontimointi ja kauppamatkustajan ongelma

Ville Autio

Tiivistelmä

Hiukkasparviontimointi on suhteellisen tuore stokastinen, populaatiopohjainen optimointimenetelmä. Se on esittelynsä jälkeen kehittynyt nopeasti mielenkiintoiseksi menetelmäksi, jota on sovellettu useilla eri alueilla. Se voidaan helposti risteyttää muiden menetelmien kanssa, ja siksi sitä voidaan soveltaa muun muassa klassisen kauppamatkustajan ongelman ratkaisemiseen, vaikkei se luontevasti kyseiseen tehtävään sovellukaan. Risteytyksen lopputuloksena on usein menetelmä, joka tuottaa parempia tuloksia kuin kumpikaan osapuolista yksinään.

Avainsanat ja -sanonnat: Hiukkasparviontimointi, kauppamatkustajan ongelma, paikallinen optimointi

CR-luokat: G.1.6, G.2.2, I.2.8

1. Johdanto

Hiukkasparviontimointi (particle swarm optimization, PSO) on Kennedyn ja Eberhartin [1995] esittelemä stokastinen, populaatiopohjainen optimointialgoritmi. Inspiraationa olivat Reynoldsin [1987] ja Heppnerin ja Grenanderin [1990] tutkimukset, joissa pyrittiin mallintamaan lintuparven toimintaa. Eberhart ja Kennedy [1995] havaitsivat nopeasti PSO:n olevan menetelmä, jolla on monia suosiollisia piirteitä, kuten yksinkertainen periaate, helppo toteutettavuus, matemaattinen yksinkertaisuus ja vähäinen sekä muistin että suoritinajan tarve. PSO on pieni osa laajempaa *luontolaskennan* (natural computing) kokonaisuutta, joka käsittelee luonnollisista prosesseista inspiraationsa saaneita algoritmeja [Banks *et al.*, 2007]. Kennedy ja Eberhart [1995] totesivat, että PSO:lla on yhteyksiä sekä *geneettisiin algoritmeihin* (genetic algorithms) että *evolutionääriseen ohjelmointiin* (evolutionary programming).

Lukuisien muiden algoritmien tapaan PSO:ta voidaan käyttää klassisen *kauppamatkustajan ongelman* (traveling salesman problem, TSP) ratkaisemiseen. Kauppamatkustajan ongelma kuuluu kombinatoristen optimointiongelmien luokkaan ja sitä käytetään usein uusien ratkaisumenetelmien testauksessa mitattikkuna [Shi *et al.*, 2007]. PSO ei perusmuodossaan sovellu erityisen hyvin tämälntyyppisten ongelmien ratkaisemiseen, joten lukuisia eri lähestymistapoja on kehitetty ja tutkittu [Shi *et al.*, 2007; Liu *et al.*, 2007; Banks *et al.*, 2008]. Mene-

telmien risteytyksillä saavutetaan jopa parempia tuloksia kuin klassisilla TSP:n ratkaisuun käytetyillä heuristiikoilla [Goldbarg *et al.*, 2006].

Luvussa 2 esitellään alkuperäinen PSO-algoritmi ja tarkastellaan sen kehittymistä nykyiseen muotoonsa. Tämän lisäksi havainnollistetaan algoritmin toimintaa, tarkastellaan toiminnan seuraamista sekä visuaalisesti että matemaattisesti. Lopuksi esitellään yleisimpiä muunnelmia ja tarkastellaan PSO:n suhdetta geneettisiin ja evolutionäärisiin algoritmeihin.

Luvussa 3 esitellään kauppamatkustajan ongelma sekä yleisellä tasolla että matemaattisesti formalisoituna. Tämän jälkeen esitellään paikalliseksi optimoinniksi kutsuttu menetelmä ja tarkastellaan sen toimintaa TSP:n ratkaisemisessa. Lopuksi käydään läpi yksi keino soveltaa PSO:ta yhdessä paikallisen optimoinnin kanssa TSP:n ratkaisemiseen ja esitellään saatuja tutkimustuloksia.

Luvussa 4 pohditaan PSO:n soveltuvuutta erilaisen ongelmien ratkaisemiseen ja esitellään sovellusalueita. Lopuksi katsastetaan avoimia tutkimuskysymyksiä.

2. Hiukkasparvioptimointi

2.1. Alkuperäinen versio

Hiukkasparvioptimoinnissa joukko yksinkertaisia olioita (hiukkasia) asetetaan jonkin ongelman tai funktion *hakuavaruuteen* (search space), ja jokainen niistä määrittää *hyvyysfunktion* (fitness function) arvon sen hetkessä sijainnissaan. Tämän jälkeen jokainen hiukkanen päätelee seuraavan liikkeensä hakuvavuudessa käyttäen hyväkseen tietoa sekä omasta parhaasta että parven yhden tai useamman hiukkasen parhaasta sijainnista; liikkeeseen kohdistetaan myöskin hieman satunnaista häiriötä. Seuraava iteraatio suoritetaan, kun kaikki hiukkasparvet ovat liikkuneet. Lopulta parvi liikkuu, hieman kuin yhdessä ruokaa etsivä lintuparvi, lähelle hyvyysfunktion optimiarvoa. [Poli *et al.*, 2007]

Jokainen hiukkasparven yksilö i koostuu kolmesta D-ulotteisesta vektorista. Nämä ovat tämänhetkinen paikka \bar{x}_i , edellinen paras paikka \bar{p}_i ja nopeus \bar{v}_i . [Poli *et al.*, 2007] Vaikka kyseessä on joukko massattomia ja tilavuudettomia pisteitä avaruudessa, Kennedy ja Eberhart [1995] päätyivät käyttämään termiä hiukkanen, koska nopeus ja kiihtyvyydet ovat paremmin niihin sopivia käsitteitä.

Tämänhetkistä paikkaa \bar{x}_i voi ajatella joukkona pisteen paikan avaruudessa määrittäviä koordinaatteja. Algoritmin jokaisella iteraatiolla tämänhetkistä paikkaa arvioidaan ongelman ratkaisuna. Mikäli tämä paikka on parempi kuin mikään tähän asti löydetyistä, tallennetaan koordinaatit vektoriin \bar{p}_i . Hyvyysfunktion toistaiseksi paras arvo tallennetaan muuttujaan $pbest_i$ (sanoista "previous best") myöhemmissä iteraatioissa tehtävää vertailua varten. Tavoitteena on etsiä parempia paikkoja ja \bar{p}_i :n sekä $pbest_i$:n päivittäminen. Uudet pisteet

valitaan lisäämällä \bar{v}_i :n arvoja \bar{x}_i :hin. Algoritmi toimii säätelämällä \bar{v}_i :tä, joka voidaan mieltää askelkooksi. [Poli *et al.*, 2007]

Hiukkasparvi on muutakin kuin vain joukko yksittäisiä hiukkasia. Käsiteltävän ongelman ratkaisu löytyy, kun hiukkaset ovat vuorovaikutuksessa keskenään. Tätä tarkoitusta varten hiukkaset järjestetään jonkinlaisella topologiaalla, jonka voi mieltää sosiaalisesti verkostoksi. [Poli *et al.*, 2007] Topologioita käsitellään tarkemmin kohdassa 2.3.

Algoritmi 1 esittää alkuperäisen PSO:n mukailleen Polin ja muiden [2007] esitystä.

Algoritmi 1	Alkuperäinen PSO
1.	Alusta hiukkaspopulaation yksilöt satunnaisilla D-ulotteisilla pisteillä ja nopeuksilla hakuavaruuteen.
2.	silmukka
3.	Evaluoi jokaiselle hiukkaselle i hyvyysfunktion arvo.
4.	Vertaa hiukkasen hyvyysarvoa sen $pbest_i$ -arvoon. Jos tämänhetkinen arvo on parempi kuin $pbest_i$, aseta $pbest_i$:n arvoksi tämänhetkinen arvo ja \bar{p}_i :n arvoksi \bar{x}_i .
5.	Tunnista naapurustosta hiukkanen, jolla on toistaiseksi paras arvo ja aseta sen indeksi muuttujaan g .
6.	Muuta hiukkasen nopeus ja paikka käyttäen seuraavia yhtälöitä:
	$\begin{cases} \bar{v}_i \leftarrow \bar{v}_i + \bar{U}(0, \phi_1) \otimes (\bar{p}_i - \bar{x}_i) + \bar{U}(0, \phi_2) \otimes (\bar{p}_g - \bar{x}_i) \\ \bar{x}_i \leftarrow \bar{x}_i + \bar{v}_i \end{cases} \quad (1)$
7.	Jos lopetuskriteeri on saavutettu (tyypillisesti riittävän hyvä hyvyysarvo tai maksimimäärä iteraatioita), poistu silmukasta.
8.	lopeta silmukka

Huomautuksia:

- $\bar{U}(0, \phi_i)$ esittää joka iteraatiossa jokaiselle hiukkaselle luotavaa vektoria, jonka alkiot ovat välille $[0, \phi_i]$ tasaisesti jakautuneita satunnaislukuja.
- \otimes on komponenttikohtainen kertolasku eli vektorien vastinkomponenttien kertominen keskenään.
- PSO:n alkuperäisessä versiossa \bar{v}_i :n jokainen komponentti pidetään välillä $[-V_{max}, V_{max}]$. Katso alakohta 2.2.3.

2.2. Kontrolliparametrit

PSO:n edellä kuvatussa versiossa ei ole montaa parametria, jotka käyttäjän tarvitsee kiinnittää tiettyyn arvoon [Poli *et al.*, 2007]. Van der Bergh ja Engelbrecht

[2006] nimeävät neljä perusparametria, jotka ovat: *kiihtyvyyškertoimet* (acceleration coefficients), *jatkavuus* (inertia), *nopeuden rajoittaminen* (velocity clamping) ja populaation koko. PSO on osoittautunut herkäksi varsinkin ensimmäisen kolmen valinnalle. Näiden väärä valinta voi johtaa parven hajaantuvaan tai sykliseen käyttäytymiseen. Ohjausparametrien valinta on myös ongelmakoh- taista. [van der Bergh and Engelbrecht, 2006]

2.2.1. Populaation koko

Populaation koko eli hiukkasten lukumäärä valitaan usein empiirisesti ongel- man ulottuvuuden ja arvioidun vaativuuden perusteella. Arvot ovat tyypillises- ti välillä 20–50. [Poli *et al.*, 2007] Kennedy ja Eberhart [2001] suosittelevat arvoa väliltä 10–50 ja toteavat, että kysymys on jossain määrin myös käyttäjän miel- tymyksistä.

2.2.2. Kiihtyvyyškertoimet

Yhtälössä (1) esiintyvät termit ϕ_1 ja ϕ_2 ilmaisevat satunnaisien voimien suu- ruutta hiukkasen oman parhaan arvon \bar{p}_i ja naapuruston parhaan arvon \bar{p}_g suuntaan. Niitä kutsutaan usein kiihtyvyyškertoimiksi, ja niiden arvot vaikut- tavat huomattavasti PSO:n käytökseen. [Poli *et al.*, 2007]

2.2.3. Nopeuden rajoittaminen

Varhaisessa PSO:n tutkimuksessa ϕ_1 ja ϕ_2 saivat arvon 2,0. Tämä johti hiuk- kasien nopeuden kasvavan nopeasti niin suureksi, että ne saattoivat lentää ulos hakuavaruudesta. Siksi PSO:n alkuperäisessä versiossa hiukkasen nopeuden jokainen komponentti tuli rajata määrätylle välille $[-V_{max}, V_{max}]$. [Banks *et al.*, 2007; Poli *et al.*, 2007] Van der Bergh ja Engelbrecht [2006] toteavat kuitenkin, ettei tämäkään välttämättä estänyt hiukkasen lentämistä hakuavaruuden rajo- jen ulkopuolelle.

Nopeuden rajoittaminen aiheuttaa kuitenkin ongelmia: V_{max} :n optimaalinen arvo on ongelmakohtainen, mutta mitään järkevää nyrkkisääntöä sen valitsemi- seksi ei ole. Tämän lisäksi nopeuden rajoittaminen johti siihen, etteivät hiuk- kasien liikeradat välttämättä koskaan yhtyneet, sillä V_{max} saattoi yksinkertai- sesti katkaista hiukkasen värähtelyt, ja siirtyminen hakuavaruuden laajamittai- sesta tutkimisesta tarkempaan lähiympäristön tutkimiseen ei onnistunut [Poli *et al.*, 2007]. Van der Bergh ja Engelbrecht [2006] todistivat, että alkuperäisen algo- ritmin parametrit johtivat aina eriäviin liikeratoihin, jos nopeutta ei rajoitettu edellä mainitulle välille.

2.2.4. Jatkavuuskerroin

Jotta nopeuden rajoittamisesta päästäisiin eroon ja hakuprosessin ohjaaminen helpottuisi, Shi ja Eberhart [1998a] esittelivät jatkavuuskertoimen ω , joka nope- utta laskiessa määrää edellisen nopeuden painoarvon. [van der Bergh and En-

gelbrecht, 2006; Poli *et al.*, 2007]. Tällöin algoritmissa 1 esiintyvä yhtälöpari voidaan kirjoittaa muotoon

$$\begin{cases} \bar{v}_i \leftarrow \omega \bar{v}_i + \bar{U}(0, \phi_1) \otimes (\bar{p}_i - \bar{x}_i) + \bar{U}(0, \phi_2) \otimes (\bar{p}_g - \bar{x}_i) \\ \bar{x}_i \leftarrow \bar{x}_i + \bar{v}_i \end{cases} \quad (2)$$

Fysikaalisesti ω voidaan mieltää hiukkasia ympäröivän väliaineen *juoksevuudeksi* (fluidity) [Poli *et al.*, 2007]. Shi ja Eberhart [1998b] tutkivat jatkavuuskertoimen eri arvojen vaikutusta algoritmin käyttäytymiseen ja totesivat, että paras tulos saavutettiin, kun ω :n lähtöarvoksi asetettiin 0,9 ja sitten asteittain vähennettiin arvoon 0,4. Tällöin hiukkaset aluksi tutkivat hakuavaruutta kattavasti ja sen jälkeen alkavat hakeutua entistä hanakammin kohti paikallisia optimiarvoja. Käyttämällä yhtälöä (2) ja valitsemalla arvot ω , ϕ_1 ja ϕ_2 hyvin, PSO käyttäytyy paljon vakaammin ja hiukkasten nopeuden rajoittaminen ei ole enää välttämätöntä tai V_{max} :n arvoksi voidaan asettaa huomattavasti suurempi luku. [Poli *et al.*, 2007]

2.2.5. Rajoituskertoimet

Clerc'in ja Kennedyn [2002] (huom. idea kehitettiin jo vuonna 1999) teoreettiset tarkastelut johtivat ideaan rajoituskertoimien käytöstä. Rajoituskertoimilla saadaan estettyä parven räjähtäminen, taataan yhdentymisen ja päästään eroon mielivaltaisesta V_{max} :n arvosta. Nämä kertoimet voidaan toteuttaa usealla tavalla, joista seuraavassa yksi:

$$\begin{cases} \bar{v}_i \leftarrow \chi(\bar{v}_i + \bar{U}(0, \phi_1) \otimes (\bar{p}_i - \bar{x}_i) + \bar{U}(0, \phi_2) \otimes (\bar{p}_g - \bar{x}_i)) \\ \bar{x}_i \leftarrow \bar{x}_i + \bar{v}_i \end{cases}, \quad (3)$$

missä

$$\chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \quad (4)$$

ja edelleen $\phi = \phi_1 + \phi_2 > 4$. [Poli *et al.*, 2007]

Tässä rajoitusmenetelmässä ϕ :n arvoksi asetetaan tyypillisesti 4,1 – siis $\phi_1 = \phi_2 = 2,05$ – ja χ :n arvoksi 0,7298, jolloin edellinen nopeus kerrotaan luvulla 0,7298 ja termit $(\bar{p} - \bar{x}_i)$ kerrotaan luvun $0,7298 \times 2,05 \approx 1,49618$ rajoittamalla satunnaisluvulla. [Poli *et al.*, 2007]

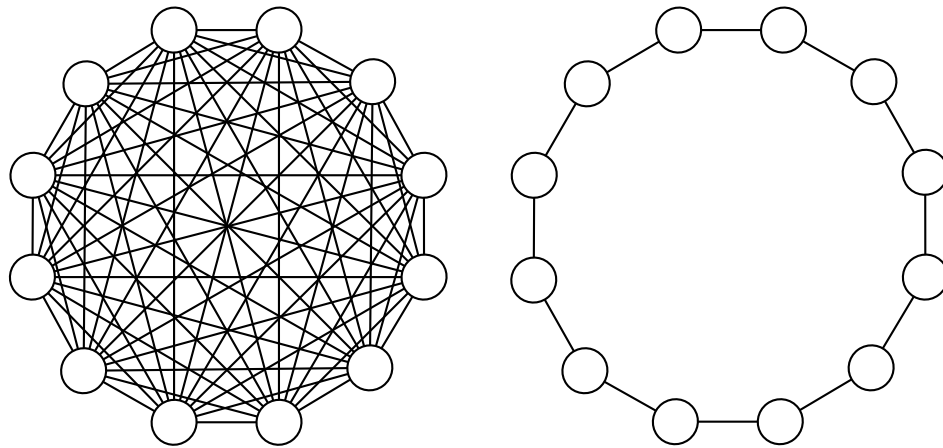
Nyt hiukkasien liikeradat yhtenevät ilman V_{max} :n käyttöä. Eberhart ja Shi [2000] havaitsivat kokeellisesti, että parempi lähestymistapa käyttäen yhtälöitä (3) ja (4) on kuitenkin asettaa $V_{max} = X_{max}$, missä X_{max} on kunkin muuttujan *dynaaminen alue* (dynamic range) kussakin ulottuvuudessa. Lopputuloksena on

tällä hetkellä kanoninen PSO, jossa ei ole enää ongelmakohtaisia parametreja, mikäli populaation koko ja naapurustotopologia oletetaan vakioiksi, kuten käytännössä on tapana. [Poli *et al.*, 2007]

Voidaan todistaa, että rajoituskertoimen käyttö on yhtäpitävä jatkavuuskertoimen ω käytön kanssa [Eberhart and Shi, 2000]. Yhtälöt (2) ja (3) voidaan muuntaa toisikseen asettamalla $\omega \leftrightarrow \chi$ ja $\phi_i \leftrightarrow \chi\phi_i$. Tällöin yhtälössä (2) käytetään arvoja $\omega = 0,7298$ ja $\phi_1 = \phi_2 = 1,49618$ [Poli *et al.*, 2007].

2.3. Naapurustotopologiat

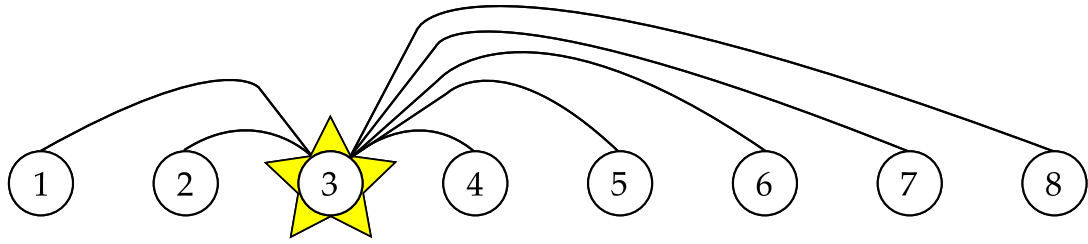
Kuten kohdassa 2.1 todettiin, ongelman ratkaisemiseksi hiukkaset ovat vuorovaikutuksessa keskenään. *Naapurustotopologia* (neighborhood topology) määrää hiukkasten naapurit indeksien eikä niiden välisen euklidisen etäisyyden perusteella [Parsopoulos, 2009]. Varhaiset topologiat käyttivät jälkimmäistä periaatetta, mutta se hylättiin nopeasti laskennallisen raskauden ja epämieluisien yhdenmisysominaisuuksien takia [Poli *et al.*, 2007]. Seuraavassa käsitellään tarkemmin kahta tavallisinta topologiaa, *gbest*- ja *lbest*-naapurustoa, ja katsastetaan myös vaihtoehtoisia ratkaisuja.



Kuva 1. *Gbest*- ja *lbest*-naapurustot [Kennedy and Mendes, 2002].

2.3.1. *Gbest*-naapurusto

Gbest-naapurusto (sanoista "global best", kuvassa 1 vasemmalla) oli Eberhartin ja Kennedyn [1995] alkuperäisessä PSO:ssa käyttämä topologia. *Gbest*-naapurustossa kaikki hiukkaset ovat yhteydessä toisiinsa, jolloin jokaisen hiukkasen käyttäytymiseen vaikuttaa minkä tahansa hiukkasen löytämä paras arvo [Kennedy and Eberhart, 2001]. Se voidaan mieltää täydellisesti yhdistetyksi graafiksi [Poli *et al.*, 2007].

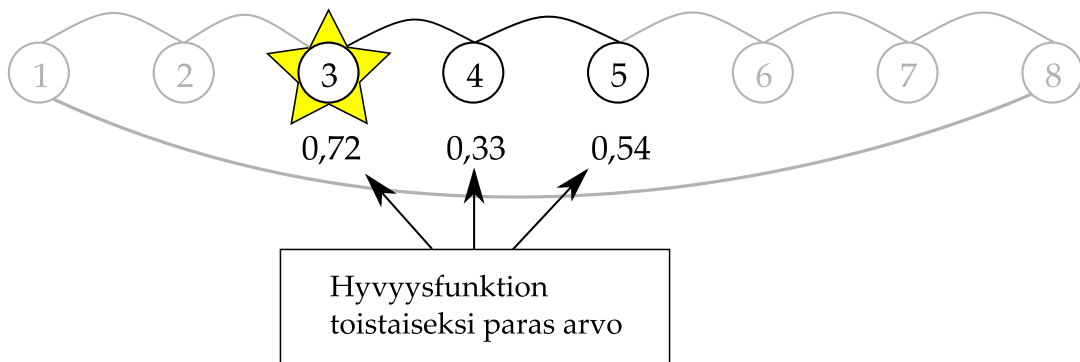


Kuva 2. Parhaan arvon löytäneen hiukkasen vaikutus *gbest*-naapurustossa [Kennedy and Eberhart, 2001].

Kuvassa 2 hiukkanen 3 on löytänyt koko populaatiossa parhaan arvon hyvyysfunktiolle, joten se vetää kaikkien muiden hiukkasten nopeuden kohti sen edellistä parasta sijaintia [Kennedy and Eberhart, 2001]. *Gbest*-naapurusto on staattinen eli naapurit ja naapurustot eivät muutu algoritmin suorituksen aikana [Poli *et al.*, 2007].

2.3.2. *Lbest*-naapurusto

Eberhart ja Kennedy [1995] esittelivät myös toisen staattisen topologian, *lbest*-naapuruston (sanoista "local best", kuvassa 1 oikealla), jossa hiukkasilla on tieto vain omasta ja k :n lähimmän naapurin parhaista arvoista. Esimerkiksi tyypillisessä tapauksessa arvolla $k = 2$ hiukkaseen i vaikuttaa hiukkasien $i - 1$, i ja $i + 1$ paras arvo [Kennedy and Eberhart, 2001]. *Lbest* sallii rinnakaisen haun, sillä nämä alipopulaatiot saattavat yhdentyä eri puolelle hakuavaruutta, mutta yhden paikan ollessa muita parempi se luultavasti houkuttelee muutkin luokseen. Täten hakuavaruus käydään kattavammin läpi, ja vaikka *lbest* yhdentyy hitaammin kuin *gbest*, se ei kuitenkaan jää niin herkästi paikalleen paikallisiin optimiarvoihin. [Poli *et al.*, 2007]



Kuva 3. Yksi alipopulaatio *lbest*-naapurustossa [Kennedy and Eberhart, 2001].

Kuvassa 3 on käytössä *lbest*-naapurusto arvolla $k = 2$. Hiukkasen 4 naapurustossa oleva hiukkanen 3 on löytänyt toistaiseksi parhaan arvon, joten hiukkanen 4 lähtee nyt kohti hiukkasen 3 edellistä parasta sijaintia. [Kennedy and Eberhart, 2001]

2.3.3. Muut topologiat

Edellä mainittujen lisäksi monia muita topologiavaihtoehtoja on tutkittu. Kennedy ja Mendes [2002] testasivat algoritmin toimintaa viidellä testifunktiolla sekä luomalla tietyt ehdot täyttäviä satunnaisia graafeja että erinäisillä tietoisesti suunnitelluilla ratkaisuilla ja vertailivat tuloksia kolmella muuttujalla. Suunnitelluista ratkaisuista selkeästi parhaaksi havaittiin von Neumannin topologia, jossa jokaisen hiukkasen naapurustoon kuului hiukkasen itsensä lisäksi hiukkaset sekä molemmilta sivuilta että ylä- ja alapuolelta. Se siis muistuttaa hilamaisista *lbest*-naapurustoa arvolla $k = 4$. *Gbest*- ja *lbest*-naapurustot pärjäsivät huonosti heidän vertailussaan.

Staattisista topologioista tehty tutkimus vihjailee, että sopeutuva topologia voisi olla hyödyllinen. Perusideana monessa lähestymistavassa on suosia paremman tuloksen saaneita hiukkasia joko priorisoimalla niitä tai karsimalla parvesta huonoimpia. Aiheesta on tehty jonkin verran tutkimusta ja tulokset ovat olleet lupaavia. [Poli *et al.*, 2007; Banks *et al.*, 2007]

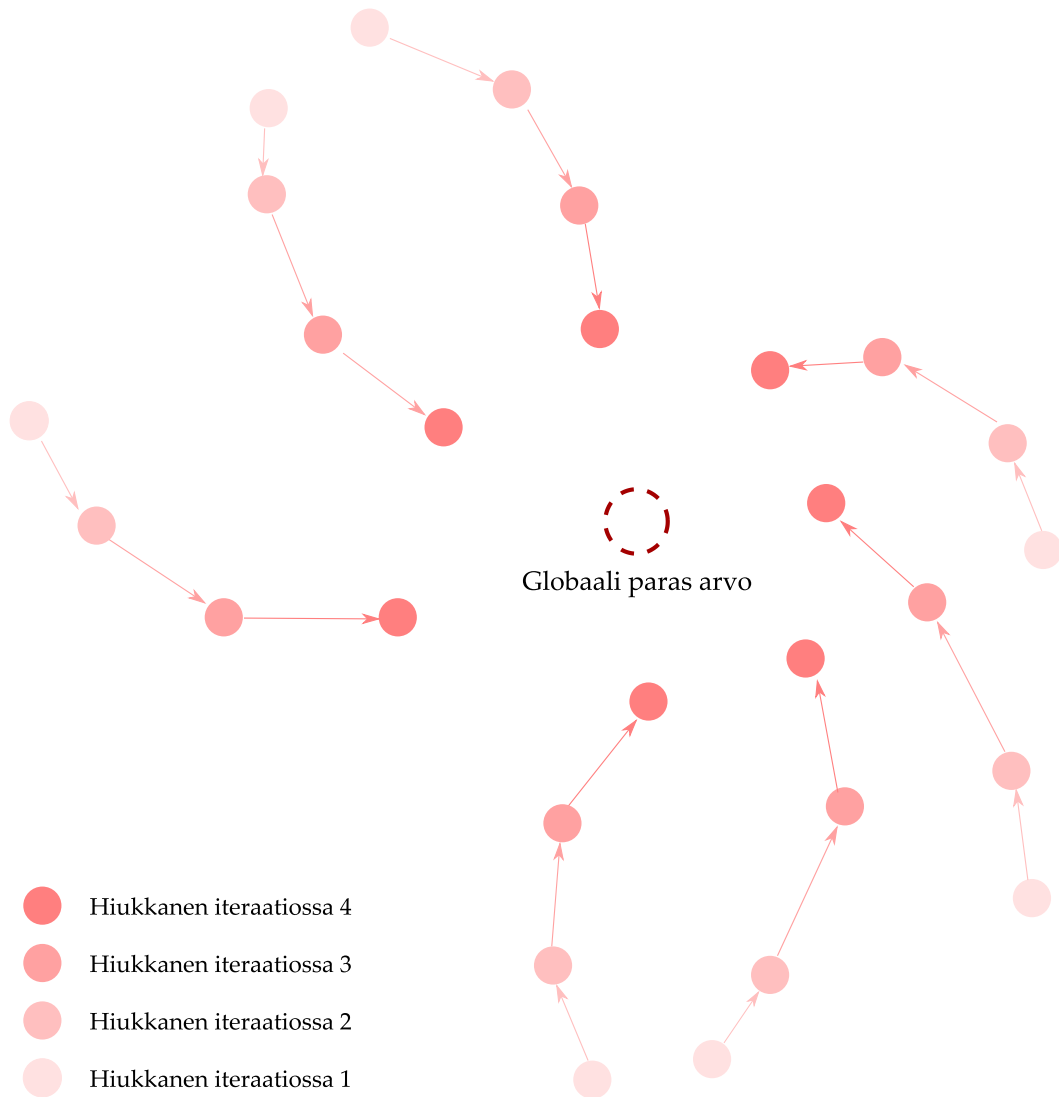
2.4. Hakuprosessin seuraaminen

Algoritmin toiminnan ymmärtämistä helpottaa huomattavasti hakuprosessin etenemisen seuraaminen. Perinteisesti evolutionäärisen prosessin etenemistä on visualisoitu piirtämällä kuvaaja, jossa x-akselilla on iteraatioiden lukumäärä ja y-akselilla hyvyysfunktion arvo [Kim *et al.*, 2009; katso myös Angeline, 1998]. Seuraavassa esitellään kaksi vaihtoehtoista keinoa seurata algoritmin toimintaa: parven visualisointi graafisesti ja sen *monimuotoisuuden* (diversity) tutkiminen.

2.4.1. Hakuprosessin visualisointi

Kim ja muut [2009] toteavat, että kaikista intuitiivisin tapa visualisoida PSO:ta on jokaisessa iteraatiossa merkitä jokaisen hiukkasen paikka, jolloin lopputuloksena syntyy animaatio hiukkasten liikkeistä. Tämä on suoraviivaista 2- ja 3-ulotteisessa avaruudessa, mutta tyypillisesti ongelmat ovat huomattavasti korkeaulotteisempia. Kim ja muut [2009] tarjoavat ratkaisuksi Sammonin [1969] kartoitusta, jonka avulla D-ulotteinen syöte voidaan muuntaa M-ulotteiseksi tulosteeksi, missä $M < D$.

Perusideana Sammonin kartoituksessa on järjestää pisteet M-ulotteiseen avaruuteen siten, että vääristymä on mahdollisimman pieni ja hiukkasten väliset etäisyydet muistuttavat mahdollisimman paljon syöteavaruuden etäisyyksiä. Sammonin kartoitus on $n:n$ hiukkasen järjestelmässä vaativuudeltaan suuruusluokkaa $\mathcal{O}(n^2D)$. Kuvaamalla hiukkaset 2-ulotteiseen avaruuteen parven yhdyntymistä voidaan jatkuvasti seurata. [Kim *et al.*, 2009]



Kuva 4. Visualisoinnin perusidea mukailien Kimin ja muiden [2009] esitystä.

Kuvassa 4 seurataan neljän iteraation ajan seitsemän hiukkasen yhdentymistä kohti globaalia parasta arvoa. Tätä perusideaa noudattaen parven toimintaa voidaan seurata erilaisissa tilanteissa. Menetelmä on myös sovellettavissa muunneltuihin PSO-algoritmeihin. Esimerkkiajoista on tehty videot [Lee]. [Kim *et al.*, 2009]. Vaihtoehtoisia visualisointikeinoja ovat käsitelleet esimerkiksi Khemka ja Jacob [2009].

2.4.2. Parven monimuotoisuuden seuraaminen

Toinen lähestymistapa algoritmin toiminnan seuraamiseen on tutkia eri muuttujien monimuotoisuutta. Suoraviivainen tapa on tutkia hyvyysarvojen keskihajontaa, mutta se antaa vain epäsuoraa tietoa hiukkasten tilasta. Muita vaihtoehtoja ovat paikkojen ja nopeuksien monimuotoisuus sekä *kognitiivinen monimuotoisuus* (cognitive diversity), joka kertoo, kuinka laajalti hiukkasiin kohdealueet ovat hajautuneet. [Shi and Eberhart, 2009]

Paikkojen ja nopeuksien monimuotoisuus voidaan luokitella yläkäsitteen *populaation monimuotoisuus* (population diversity) alle. Paikkojen monimuotoisuutta voidaan tarkastella esimerkiksi euklidisten etäisyyksien perusteella mutta parempi lähestymistapa on tarkastella ulottuvuuskohtaisia koordinaatteja, sillä joissain ulottuvuuksissa parvi saattaa olla yhtymässä ja joissain eriytymässä. Nopeutta voidaan tarkastella sekä tutkimalla hiukkasten vauhtien että suuntien monipuolisuutta. Nämä populaation monimuotoisuutta kuvaavat arvot kertovat siis parven senhetkisestä tilanteesta. [Shi and Eberhart, 2009]

Kognitiivinen monimuotoisuus käyttää hyödykseen hiukkasten muistia eli vektorin \bar{p}_i arvoa, jota tarkkailemalla saadaan tietoa siitä, missä päin hakuvaruutta kohdealueet näyttäisivät olevan. Myös \bar{p}_i :n arvot voivat vaihdella voimakkaasti ulottuvuuskohtaisesti, joten myös kognitiivista monimuotoisuutta laskettaessa vertaillaan ulottuvuuskohtaisia arvoja. [Shi and Eberhart, 2009] Yhdistämällä nämä kaksi mittaria saamme taulukon 1.

	Korkea populaation monimuotoisuus	Matala populaation monimuotoisuus
Korkea kognitiivinen monimuotoisuus	Globaali haku → globaali haku	Paikallinen haku → globaali haku
Matala kognitiivinen monimuotoisuus	Globaali haku → paikallinen haku	Paikallinen haku → paikallinen haku

Taulukko 1. Populaation monimuotoisuuden ja kognitiivisen monimuotoisuuden yhdistelmät [Shi and Eberhart, 2009]

Parven hakukäyttäytymiseen vaikuttaa molemmat monimuotoisuuden arvot ja taulukko 1 listaa niiden yhdistelmät. Korkea monimuotoisuus molemmilla mittareilla tarkoittaa, että parvi on hajallaan ja hiukkaset lentävät kohti useita eri alueita. Jos populaation monimuotoisuus on korkea mutta kognitiivinen monimuotoisuus matala, parvi alkaa tutkia paikallista ympäristöään eli yhden tyä. Päinvastaisessa tilanteessa eli populaation ollessa kasassa ja kognitiivisen monimuotoisuuden ollessa suuri hiukkaset alkavat hajaantua tutkiakseen hakuvaruutta kattavin. Viimein, molempien saadessa matalan arvon, parvi jatkaa lähiympäristönsä tarkkaa tutkimista. [Shi and Eberhart, 2009]

Hyvän hakukyvyn omaava PSO kykenee vaihtamaan tilaansa johonkin toiseen. Yksi algoritmin suurimpia ongelmia on helppo juuttuminen paikallisiin minimeihin, ja käyttämällä hyväksi tietoa parven monimuotoisuudesta tämä voidaan havaita aikaisemmin ja tarvittaessa voidaan tehdä korjauksia esimerkiksi mutaatioita käyttämällä. [Shi and Eberhart, 2009; Banks *et al.*, 2007]

2.5. Muunnelmat

Jo aiemmin alakohdissa 2.2.1–2.2.5 kuvatus kehityksen lisäksi algoritmista on tehty lukuisia eri versioita sekä yleisen suorituskyvyn että tietyille sovellusalueille sopivuuden parantamiseksi [Poli *et al.*, 2007]. Seuraavassa käsitellään hieman tarkemmin binääristä PSO:ta, täydellisesti informoitua parvea ja yleisemmällä tasolla risteytyksiä ja muita muunnelmia.

2.5.1. Binäärinen PSO

Kanoninen PSO toimii jatkuvassa hakuavaruudessa, joten se ei luontevasti sovellu diskreettien ongelmien ratkaisemiseen. Kennedy ja Eberhart [1997] esittelivät diskreetin, binäärisiä bittijonoja käyttävän version, jossa siis paikkavektorin \bar{x}_i jokainen komponentti x_{id} voi saada vain arvon 0 tai 1. Arvo määräytyy seuraavasti: nopeusvektorin \bar{v}_i d. komponentti v_{id} litistetään välille [0,1] käyttämällä logistista sigmoidifunktiota

$$S(v_{id}) = \frac{1}{1+e^{-v_{id}}}, \quad (5)$$

jonka jälkeen luodaan satunnaisluku r väliltä [0,1] ja sovelletaan yhtälöä

$$x_{id} = \begin{cases} 1, & r < S(v_{id}) \\ 0, & r \geq S(v_{id}) \end{cases}. \quad (6)$$

Nopeuden komponentti v_{id} voidaan siis mieltää todennäköisyydeksi, jolla x_{id} saa arvon 1. Tätä binääristä versioita voidaan soveltaa luonteeltaan diskreettien ongelmien ratkaisuun. Kennedy ja Spears [1998] vertasivat binääristä PSO:ta kolmeen erilaiseen geneettiseen algoritmiin satunnaisilla multimodaalisilla ongelmilla (hakuavaruuden multimodaalisuus tarkoittaa huippujen lukumäärää). PSO havaittiin ainoaksi algoritmiksi, joka löysi globaalin optimiarvon joka testissä, ja se oli myös nopein kaikissa paitsi aivan yksinkertaisimmissa testeissä. Diskreetistä PSO:sta on myös tehty tuoreempaa tutkimusta, joissa on luotu algoritmista erilaisia variaatioita. [Poli *et al.*, 2007; Banks *et al.*, 2007]

2.5.2. Täydellisesti informoitu parvi

Kanonisessa PSO:ssa käytetään hyväksi vain hiukkasen omia ja parhaan naapurin tietoja [Poli *et al.*, 2007]. *Täydellisesti informoitu hiukkasparvi* (fully informed particle swarm, FIPS) on Mendesin ja muiden [2004] esittelemä muunnelma, jossa jokaiseen hiukkaseen ei vaikuta pelkästään naapuruston paras hiukkanen vaan koko naapurusto. Poli ja muut [2007] muotoilevat FIPSin yhtälöillä

$$\begin{cases} \bar{v}_i \leftarrow \chi(\bar{v}_i + \frac{1}{K_i} \sum_{n=1}^{K_i} \bar{U}(0, \phi) \otimes (\bar{p}_{nbr_n} - \bar{x}_i)), \\ \bar{x}_i \leftarrow \bar{x}_i + \bar{v}_i \end{cases}, \quad (7)$$

missä K_i on hiukkasen i naapurien lukumäärä ja nbr_n on i :n n . naapuri. Poli ja muut [2007] toteavat, että tämä on sama kuin perinteinen versio, jos otetaan huomioon vain hiukkanen itse ja naapuruston paras tapauksessa $K_i = 2$. Mendesin ja muiden [2004] testeissä FIPSin suorituskyky oli parempi kuin kanonisen algoritmin kaikilla mittareilla, mutta he ja Poli ja muut [2007] toteavat, että suorituskyky on huomattavan riippuvainen valitusta naapurustotopologiasta.

2.5.3. Risteytykset

Banks ja muut [2008] toteavat, että PSO:n populaatiopohjaisen luonteen vuoksi se on helposti risteytettävissä evolutionääristen ja geneettisten algoritmien kanssa. Tästä johtuen kaikkien tehtyjen risteytyksien käsittely ei ole tämän työn piirissä mahdollista, joten seuraavassa käsitellään yleisiä periaatteita.

PSO:n suhdetta evolutionääriiseen laskentaan on käsitelty esimerkiksi Angeline [1998]. Evolutionäärisissä menetelmissä käytetään valintafunktiota karsimaan populaatiosta huonosti menestyviä yksilöitä. Hiukkasparvionnoinnissa ei vastaavaa funktiota ole, mutta jokaisen hiukkasen tieto omasta parhaasta sijainnistaan on verrattavissa vanhempien rooliin evolutionäärisissä algoritmeissa. Menetelmät eroavat myös haun suuntaamisessa: hiukkasparvionnoinnissa nopeusvektori osoittaa aina tiettyyn suuntaan ja siten sillä on huomattava vaikutus siihen, miten laajalti hakuavaruutta tutkitaan; evolutionäärisessä laskennassa mutaatio-operaatio kohdistuu kaikkiin suuntiin. Kennedy ja Eberhart [1995] totesivat, että yhtäläisyyksiä ovat muun muassa prosessien stokastisuus ja hyvyysarvon käsite.

Hiukkasten tapa suuntautua kohti oman parhaan ja naapuruston parhaan arvon painotettua keskiarvoa muistuttaa Kennedyn ja Eberhartin [1995] mukaan käsitteellisesti geneettisten algoritmien risteytysoperaatiota (crossover). Eberhart ja Shi [1998] toteavat myös, että paikallisten tai globaalien optimien välimaastoa (ainakin hetkellisesti) tutkivat hiukkaset vastaavat risteytystä josain määrin. Mutaatioon ja valintaan pätevät samat erot kuin muissakin evolutionäärisissä menetelmissä [Eberhart and Shi, 1998].

Algoritmien välisiä risteytyksiä voidaan toteuttaa lukemattomilla eri tavoilla, joita ovat esimerkiksi turnausvalinnan soveltaminen hiukkasparveen, hyvien yksilöiden risteyttäminen keskenään, toisen algoritmin käyttö tiettyjen ehtojen ollessa voimassa ja hiukkasparven tiettyjen ominaisuuksien, esimerkiksi nopeuden ja sosiaalisuuden, hyödyntäminen muissa menetelmissä. Edellä mainitut keinot pyrkivät usein parantamaan yleistä suorituskykyä; näiden lisäksi toinen lähestymistapa on mukauttaa algoritmia ongelmakohtaisesti. Seikkaperäi-

sempiä katsauksia aiheesta tehdystä tutkimuksesta ovat kirjoittaneet esimerkiksi Poli ja muut [2007] ja Banks ja muut [2008].

3. Kauppamatkustajan ongelma

3.1. Kauppamatkustajan ongelmasta yleisesti

Kauppamatkustajan ongelma (TSP) on klassinen, vaikea kombinatorisen optimoinnin ongelma. Applegate ja muut [1998] toteavat, että ongelman alkuperä on jokseenkin epämääräinen: ensin sen esitteli matemaatikko ja taloustieteilijä Karl Menger Wienissä 1920-luvulla, sitten se ilmaantui Princetonin yliopiston matemaatikkipiireistä 1930-luvulla ja lopulta 1940-luvulla tilastotieteilijät käsitelivät aihetta.

TSP on muotoiltavissa seuraavasti mukaillen Applegaten ja muiden [1998] esitystä: Otetaan äärellinen joukko kaupunkeja ja tiedot kaikkien kaupunkien välisistä etäisyyksistä; etsi lyhin kaikissa kaupungeissa käyvä reitti siten, että palaat lähtöpaikkaan. Tästä yksinkertaisesta muotoilusta huolimatta kyseessä on hankala ongelma, joten siitä on tullut suosittu koealusta algoritmiikassa [Applegate *et al.*, 1998].

Matemaattisesti TSP voidaan esittää seuraavasti mukaillen Heldin ja Karpin [1970] esitystä: Olkoon G_n täydellinen suuntaamaton graafi, jonka solmut ovat joukko $\{1, 2, \dots, n\}$. Solmujen i ja j välillä olevaa kaarta merkitään $\{i, j\}$ ja sen painoa c_{ij} . Graafi on symmetrinen eli $c_{ij} = c_{ji}$. Edelleen aligraafin paino on sen kaarien painojen summa. Kauppamatkustajan ongelmassa etsitään *matkaa* (tour) eli jokaisessa solmussa täsmälleen kerran käyvää sykliä, jolla on pienin mahdollinen paino.

Kauppamatkustajan ongelma kuuluu vaativuusluokkaan *NP-kova* (NP-hard), joten pidetään epätodennäköisenä, että löytyy polynomiaalisessa ajassa optimaalisen ratkaisun antavaa algoritmia. Erilaisia ratkaisuvaihtoehtoja n kaupungin joukossa on $(n - 1)!$ kappaletta, joten TSP:n ratkaiseminen raa'an voiman haulla eli käymällä kaikki mahdolliset vaihtoehdot läpi on suoritusai-kavaativuudeltaan luokkaa $\mathcal{O}(n!)$. Täten tämä ratkaisu on epäkäytännöllinen jo 20 kaupungin joukolla, ja siksi on kehitetty lukuisia erilaisia algoritmeja, jotka pyrkivät ratkaisemaan ongelman tehokkaammin. [Johnson, 1990; Held and Karp, 1962]

3.2. Ratkaisun etsiminen hiukkasparioptimoinnilla

Kuten luvussa 1 todettiin, kanoninen PSO soveltuu melko huonosti kombinatoristen ongelmien ratkaisemiseen. Tämän takia niiden ratkaisemiseen on kehitetty lukuisia muunnelmia kanonisesta algoritmista [Banks *et al.*, 2008]. Tässä työssä käytetään esimerkkinä Goldbargin ja muiden [2006] esitystä, jossa PSO:hon yhdistettiin TSP:n ratkaisemisessa yleisesti käytetty Lin-Kernighan-

algoritmi. Tästä syystä seuraavassa käsitellään ensiksi paikallista optimointia, johon Lin-Kernighan-algoritmi kuuluu.

3.2.1. Paikallinen optimointi ja Lin-Kernighan-algoritmi

TSP:n ratkaisemiseen on kehitetty lukuisia approksimointimenetelmiä (heuristiikoita) lähes optimaalisten ratkaisujen löytämiseksi. Usein nämä algoritmit pohjautuvat *paikalliseen optimointiin* (local optimization), jossa ongelman ratkaisua parannetaan iteratiivisesti tekemällä paikallisia muutoksia. [Johnson, 1990] Seuraavassa käsitellään lyhyesti TSP:n ratkaisemista tavallisimmilla paikallisilla optimointimenetelmillä.

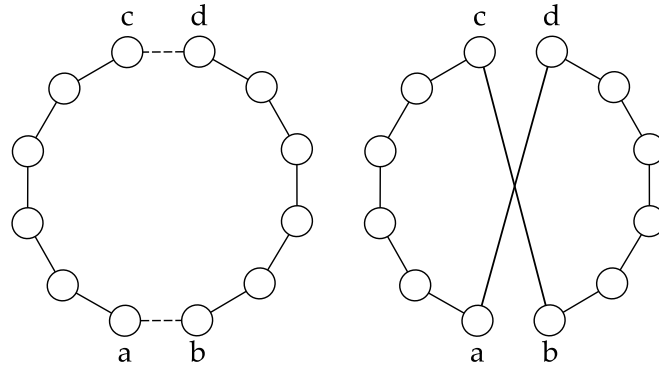
Paikallisessa optimoinnissa naapuruston käsite on keskeinen. Matkat T ja T' ovat naapurustossa eli relaatioissa (T, T') , jos ja vain jos T' on T :n legitiimi muunnos (perturbaatio) eli naapuri. Tällaiselle rakenteelle pohjautuva paikallinen optimointialgoritmi koostuu kahdesta aliohjelmasta: algoritmista A , joka tuottaa syötteestä I alkuarvon eli matkan T_0 , ja algoritmista B , joka käyttäen hyväkseen I :tä ja annettua matkaa T selvittää, onko olemassa sellaista naapuria, jonka pituus on lyhyempi. Jos tällainen matka T' löytyy, se palautetaan, jos ei, niin T :tä sanotaan paikalliseksi optimiksi. Huomattavaa on, että täten saatavan paikallisen optimin ei tarvitse olla globaali optimi tai edes lähellä sitä, mutta tietenkin toivottavaa on, että lopputulos olisi kohtuullisen hyvä. [Johnson, 1990] Paikalliseen optimointiin perustuva TSP:n ratkaiseminen voidaan esittää algoritmia 2 mukailleen Johnsonin [1990] esitystä.

Algoritmi 2 TSP:n ratkaiseminen paikallisella optimoinnilla

1. Kutsu aliohjelmaa A syötteellä I ja saadaksesi alkuarvon $T_0 \rightarrow T$.
 2. **while** T ei ole paikallinen optimi **do**
 - 2.1 Kutsu aliohjelmaa B syötteillä I ja T .
 - 2.2 Jos B palauttaa paremman ratkaisun T' , sijoita $T' \rightarrow T$.
 3. Palauta T .
-

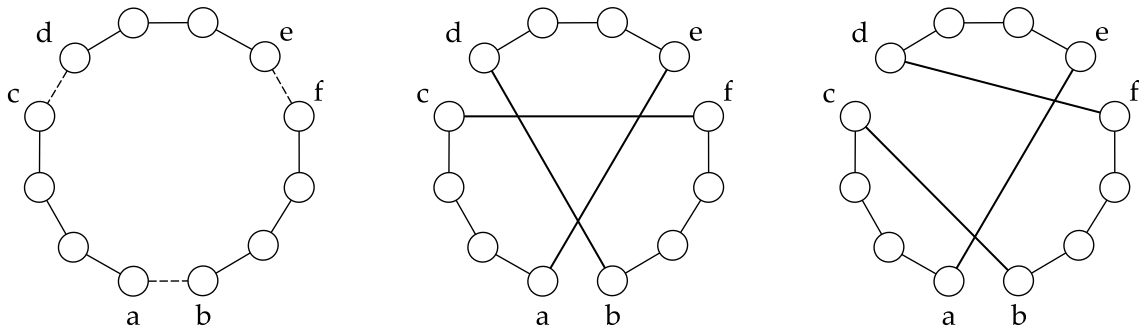
Tunnetuimmat TSP:n ratkaisemisessa käytetyt paikalliset optimointialgoritmit ovat *2-Opt*, *3-Opt* ja *Lin-Kernighan*. Johnson ja McGeoch [1997] toteavat, että vuosina 1973–1989 Lin-Kernighania pidettiin TSP-heuristiikkojen maailmanmestarina. Näitä menetelmiä vastaavat naapurustot voidaan muotoilla seuraavasti mukailleen Johnsonin [1990] esitystä:

2-Opt: Kaksi matkaa ovat naapureita, mikäli toinen saadaan toisesta poistamalla kaksi kaarta, kääntämällä toinen syntyvistä poluista ja yhdistämällä nämä polut sen jälkeen uudestaan. Kuvassa 5 on esimerkki tämän tyyppisestä muunnoksesta.



Kuva 5. 2-Opt-muutos, vasemmalla alkuperäinen matka ja oikealla lopputulos [Johnson and McGeoch, 1997].

3-Opt: Kaksi matkaa ovat naapureita, mikäli toinen saadaan toisesta poistamalla kolme kaarta ja yhdistämällä muodostuneet polut uudestaan jollain uudella tavalla, mahdollisesti yhden tai useamman polun kääntämällä. Kuvassa 6 on kaksi esimerkkiä tämän tyyppisestä muunnoksesta.



Kuva 6. Kaksi mahdollista 3-Opt-muutosta, alkuperäinen matka vasemmalla ja vaihtoehtoiset lopputulokset oikealla [Johnson and McGeoch, 1997].

Lin-Kernighan on Linin ja Kernighanin [1973] esittelemä heuristiikka, jota voidaan pitää edellä mainittujen yleistyksenä eli *k-optina*. Johnson [1990] kuitenkin toteaa sen olevan huomattavasti monimutkaisempi, asymmetrinen rakenne, joten seuraavassa esitetään vain Johnsonin [1990] luonnehdinta.

Lin-Kernighan: Jotta T' olisi T :n naapuri, sen täytyy olla lyhyempi, se täytyy saada T :stä kolme kaarta rikkomalla, uudelleenjärjestelmällä ja sen jälkeen suorittamalla vielä ylimääräisiä *ahneella rajatun levyisellä haulla* (greedy bounded-width search) rakennettuja *2-opt*-siirtoja.

Johnson ja McGeoch [1997] huomauttavat, että Lin-Kernighan-algoritmin maine on levinnyt laajemmalti kuin tiedot sen yksityiskohtaisesta toteuttamisesta, ja jotkut kirjoittajat jopa pitävät sitä 3-Optin tai jopa 2-Optin synonyymina. Tämän työn puitteissa paikallista optimointia ei käsitellä tämän enempää, asiaa ovat kattavammin käsitelleet esimerkiksi Johnson [1990] ja Johnson ja McGeoch [1997].

3.2.2. PSO-Lin-Kernighan -risteytys

Goldbargin ja muiden [2006] PSO-LK-toteutuksessa TSP:n ratkaisuvaihtoehdot esitetään n kaaren permutaatioina, kuten evolutionäärisissä ratkaisumalleissa on yleensä tapana. Tässä toteutuksessa hiukkasella on kolme liikkumisvaihtoehtoa:

1. Jatkaa omaa reittiään (Lin-Kernighan)
2. Palata takaisin edelliseen omaan parhaaseen paikkaan
3. Liikkua kohti globaalia parasta ratkaisua (hiukkasta).

Naapurustossa on siis vain globaali paras ratkaisu, ja menetelmä poikkeaa tavallisesta myös siinä, miten nopeus määritellään. [Goldbarg *et al.*, 2006]

Oman reitin kulkeminen toteutetaan käyttämällä paikallista optimointia eli tässä tapauksessa Lin-Kernighan-algoritmia. Toiset kaksi liikkumisvaihtoehtoa toteutetaan käyttämällä *polkujen uudelleenlinkittämistä* (path relinking), mikä tässä tapauksessa toteutetaan siirtämällä paikkavektorin elementtejä yksi kerrallaan vaihtamalla niiden paikkaa vasemmalla olevan elementin kanssa. Esimerkiksi ratkaisusta (1 2 3 4 5) päästään ratkaisuun (3 5 1 2 4) siirtämällä ensin elementtiä 3 vasemmalle, kunnes se on paikallaan ja sitten siirtämällä vastaavasti elementtiä 5. Näitä kolmea liikkumisvaihtoehtoa vastaavat muuttujat p_1 , p_2 ja p_3 , jotka kuvaavat todennäköisyyttä, jolla hiukkanen valitsee kyseisen vaihtoehdon. [Goldbarg *et al.*, 2006] Algoritmissa 3 esitetään Goldbargin ja muiden [2006] ratkaisumenetelmä.

Algoritmi 3 PSO-LK-risteytys TSP:n ratkaisuun

1. Määritä todennäköisyyksien p_1 , p_2 ja p_3 alkuarvot. $p_1 + p_2 + p_3 = 1$.
 2. Alusta hiukkaspopulaation yksilöt (kuvaus jäljempänä).
 3. **silmukka**
 4. Evaluoi jokaiselle hiukkaselle i ratkaisu.
 5. Vertaa hiukkasen ratkaisua sen $pbest_i$ -arvoon. Jos tämänhetkinen ratkaisu on parempi kuin $pbest_i$, aseta $pbest_i$:n arvoksi tämänhetkinen ratkaisu ja \bar{p}_i :n arvoksi \bar{x}_i .
 6. Aseta parhaan ratkaisun löytänyt hiukkanen globaalisti parhaaksi ratkaisuksi.
 7. Valitse jokin liikkumisvaihtoehto (1, 2 tai 3) ja päivitä hiukkasen paikka.
 8. Päivitä todennäköisyydet seuraavasti:
 $p_1 = p_1 \times 0,95; p_2 = p_2 \times 1,01; p_3 = 1,00 - (p_1 + p_2)$
 9. Jos lopetuskriteeri on saavutettu, poistu silmukasta.
 10. **lopeta silmukka**
-

Alussa todennäköisyys p_1 asetetaan suureksi ja vastaavasti p_2 ja p_3 pieniksi, jotta ensimmäisillä iteraatioilla tapahtuu enemmän yksilöllisiä siirtymiä. Suorituksen edetessä p_3 saa lopulta suurimman arvon, jotta viimeisillä kierroksilla haku keskittyisi hakuavaruuden hyville alueille. [Goldberg *et al.*, 2006]

Hiukkaspopulaatio alustetaan käyttämällä mukautettua lähimmän naapurin heuristiikkaa. Lähtökaupunki valitaan satunnaisesti ja ratkaisuun lisätään kaupungeja seuraavasti: edellisen valitun kaupungin lähimmistä naapureista valitaan 5 % ehdokaslistalle, josta vuorostaan valitaan satunnaisesti yksi. Tätä menetelmää jatketaan, kunnes saadaan jokin ratkaisu TSP:lle. [Goldberg *et al.*, 2006]

3.3. Tulokset

Goldberg ja muut [2006] käyttivät testiaineistona Reineltin [1991] yleisesti käytettyä TSPLIB-kirjastoa. Applegate ja muut [2007] löysivät optimaalisen ratkaisun 85 900 kaupungin joukolle, mikä oli viimeinen ratkaisematon TSPLIB-ongelma, joten kaikkien TSPLIB-ongelmien optimaaliset ratkaisut tiedetään. Kyseinen ratkaisu on myös tällä hetkellä suurin optimaalisesti ratkaistu symmetrinen graafi, ja kuriositeettina kerrottakoon, että ratkaisun löytämiseen käyttäen 64 kappaletta AMD Opteron 250 (2,4 GHz) ja 192 kappaletta Intel Xeon (2,66 GHz) -suorittimia kului yli 136 tietokonevuotta [Applegate *et al.*, 2007].

Testiajoissaan Goldberg ja muut [2006] asettivat lopetuskriteereiksi optimin löytymisen lisäksi ylärajat sekä suoritusajalle että iteraatioiden määrälle. Näiden lisäksi suoritus lopetettiin, mikäli parannusta ei tapahtunut 20 iteraation aikana. PSO-LK-algoritmia verrattiin pelkkään Lin-Kernighan-algoritmiin 30 eri symmetrisessä TSPLIB-tapauksessa ja 23 tapauksessa Cookin ja Seymourin [2003] Tourmerge-algoritmiin sekä kahteen eri iteroitavaan Lin-Kernighan-algoritmiin. Ensimmäisessä tapauksessa kaikki testit ajettiin itsenäisesti 20 kertaa ja jälkimmäisessä 20 PSO-LK-ajon tuloksia verrattiin sekä Tourmerge-algoritmin että iteroitavien LK-algoritmien NYYY (Nguyen-Yoshihara-Yamamori-Yasunaga) ja JM (Johnson-McGeoch) testiajojen tuloksiin [Johnson]. Kaupunkien lukumäärä näissä testeissä oli väliltä 439–7397. [Goldberg *et al.*, 2006]

Seuraavat tulokset ilmaisevat tulosten prosentuaalista poikkeavuutta optimiarvosta. Lin-Kernighanin minimi- ja keskipoikkeaman keskiarvot olivat 0,1115 % ja 0,3431 % ja PSO-LK-algoritmin vastaavasti 0,0022 % ja 0,0154 %. Yksittäisiä testejä vertailemalla PSO-LK paransi parasta tulosta 23 tapauksessa ja keskiarvoa jokaisessa. Keskimääräinen suoritus aika PSO-LK-algoritmillä (noin 421 sekuntia) oli kuitenkin noin 84-kertainen suhteessa pelkkään Lin-Kernighaniin (noin 5 sekuntia). [Goldberg *et al.*, 2006]

Tourmerge-algoritmin minimi- ja keskipoikkeamien keskiarvot olivat 0,00461 % ja 0,0467 % ja PSO-LK:n vastaavasti 0,0013 % ja 0,00186 %. Tourmerge

ei kuitenkaan onnistunut ratkaisemaan kahta testiongelmaa, joten vertailu tehtiin vain onnistuneiden testien osalta. Iteroitavien Lin-Kernighan-algoritmien kanssa vertailu tehtiin pelkäästään parhaiden tulosten osalta. NYYY-algoritmin parhaiden tulosten keskipoikkeama oli 0,0199 % ja JM-algoritmin vastaavasti 0,0569 %, mitkä ovat korkeammat kuin PSO-LK:n 0,0028 % poikkeama. PSO-LK tuotti myös lukumäärissä mitaten selkeästi eniten parhaita tuloksia verrattuna kaikkiin muihin. [Goldbarg *et al.*, 2006]

Luonnollisesti aiheesta on tehty lukuisia muita tutkimuksia, joista seuraavassa lyhyesti. Clercin [2000] diskreetti PSO-variantti oli lienee varhaisin yritys ratkaista TSP, ja tulokset eivät tuolloin olleet erityisen hyviä, mutta hän totesi, että algoritmi on kyllä melko helposti mukautettavissa kombinatorisien ongelmien ratkaisuun. Li ja muut [2006] yhdistivät diskreettiin PSO:hon 3-Opt paikallisen haun, millä saavutettiin hieman huonompia tuloksia kuin *muurahaisyhdyskuntaoptimoinnin* (ant colony optimization, ACO) *muurahaisyhdyskuntajärjestelmä*-muunnelmalla (ant colony system, ACS). Liu ja muut [2007] käyttivät kahta itsenäisesti ratkaisua etsivää parvea, jotka risteyttivät joka kierroksella globaalisti parhaat hiukkaset keskenään ja mutaatiota hyväksikäyttäen paransivat globaalia hakukykyä. Shi ja muut [2007] käyttivät diskreettiä PSO:ta, jossa hiukkasia risteytettiin keskenään 2-Opt-menetelmää muistuttavalla menetelmällä. Heidän algoritminsä tuotti 19 testitapauksessa keskimäärin 4,25 % optimista poikkeavan ratkaisun [Shi *et al.*, 2007].

Shuang ja muut [2009] yhdistivät PSO:n muurahaisyhdyskuntaoptimointiin liittämällä PSO:n globaalin ja lokaalin hakumekanismin feromonien päivittämissä sääntöihin, jolloin sekä koko yhdyskunta hakeutuu kohti parempia ratkaisuja että toisaalta tutkii kyseisiä alueita kattavasti. Heidän testissään tämä PS-ACOristeytys tuotti selkeästi parempia tuloksia kuin verrokkeina käytetyt geneettinen algoritmi, tavallinen ACO sekä *MAX-MIN -muurahaisjärjestelmä* -muunnos (MAX-MIN ant system) [Shuang *et al.*, 2009].

4. Lopuksi

4.1. Muista sovelluksista

Wolpertin ja Macreadyn [1995] *No Free Lunch* -lauseen mukaan emme voi kaikkien ongelmien joukossa sanoa, että jokin algoritmi olisi parempi kuin toinen. Kennedy ja Eberhart [2001] kuitenkin toteavat, kyseisestä joukosta vain pieni osajoukko on oikeasti kiinnostava ja eri tilanteissa eri menetelmien välille löytyy selviä eroja. Yksi keino löytää nämä erot on tutkia algoritmien käyttäytymistä vaikeaksi tunnettujen ongelmien ratkaisemisessa [Kennedy and Eberhart, 2001]. Hiukkasparvioptimointi on ottanut paikkansa muiden tämänkaltaisten ongelmien ratkaisemiseen soveltuvien algoritmien joukossa ja on osoittautunut erityisen lupaavaksi ratkaistaessa muun muassa ongelmia, jotka ovat

multimodaalisia tai sellaisia, joiden ratkaisemiseen ei ole olemassa erikoistuneita menetelmiä tai ne eivät anna tyydyttäviä tuloksia. [Kennedy and Eberhart, 2001; Poli *et al.*, 2007].

Kennedyn ja Eberhartin [1995] alkuperäinen sovellus oli kolmitasoisen XOR-ongelman ratkaisevan *neuroverkon* (neural network) kouluttaminen. Kennedy ja Eberhart [2001] kertovat myös reaali maailman sovelluksesta, jossa algoritmia sovellettiin samaan ongelmaan menestyksekkäästi. Eräs työryhmä, jossa Eberhart työskenteli, koulutti neuroverkkoa sähköauton akkujärjestelmän lataustason arviointia varten, ja aluksi käytössä olleella backpropagation-menetelmällä kesti noin 3,5 tuntia käydä kaikki noin 2 500 opetusaineistoa läpi. Vaihtamalla opetusalgoritmiksi PSO vastaavan tuloksen tuottavan suorituksen aika putosi 2,2 minuuttiin. [Kennedy and Eberhart, 2001]

Hiukkasparviontimointia on toki sovellettu myös lukuisien muiden reaali maailman ongelmien ratkaisemiseen. Niiden lukumäärän ja monimuotoisuuden vuoksi niitä ei tämän työn puitteissa esitellä, katsauksia aiheeseen ovat tehneet esimerkiksi lähteistä Poli ja muut [2007] ja Banks ja muut [2008]. Polin ja muiden [2007] katsauksessa korostuu varsinkin erilaiset sähkötekniikkaan enemmän tai vähemmän liittyvät sovellukset, sillä he analysoivat IEEE Xplore -tietokannasta löytyviä julkaisuja. Banks ja muut [2008] toteavat, että PSO:ta on sovellettu muun muassa erilaisissa lääketieteellisen laskennan ja taloudellisen ennustamisen tehtävissä.

4.2. Avoimia kysymyksiä

Hiukkasparviontimointi on suhteellisen uusi menetelmä ja siten siihen liittyy edelleen monia avoimia kysymyksiä. Käytännön sovellusten puolella ongelmia PSO:lle tuottavat muun muassa dynaamisesti muuttuvat ympäristöt, monta tavoitetta sisältävät ongelmat, ongelmakohtaiset rajoitteet sekä jumittuminen paikallisiin optimiarvoihin [Banks *et al.*, 2008]. Poli ja muut [2007] esittelevät sekä algoritmin käytännön toteuttamiseen että teoreettisiin tarkasteluihin liittyviä kysymyksiä.

Käytännön toteutuksissa parven alustus on vielä jossain määrin problemaattinen kysymys, sillä parvella väitetään olevan taipumus hakeutua lähellä alustuspaikkaa oleviin optimiarvoihin. On myös pohdittu, kannattaako joka iteraatiolla päivittää jokaisen hiukkasen arvo ja pitäisikö hyviä hiukkasia suosia enemmän. Erilaisien muistimekanismien vaikutus on myös vielä varsin vähän tutkittu aihepiiri. Suorituksen aikana sopeutuvat eli muuttuvat parvet ovat myös avoin tutkimuskohde, jossa tavoitteena voisi olla täysin parametrivapaa itsenäisesti sopeutuva hiukkasparvi. [Poli *et al.*, 2007]

Teoreettisempien kysymysten osalta Poli ja muut [2007] toteavat, että kysymykset ovat pitkälti samankaltaiset muita stokastisia algoritmeja koskevat kysymykset. PSO:ta koskevia avoimia kysymyksiä ovat muun muassa liikeyhtä-

löiden ja naapurustotopologian vaikutukset PSO:n dynaamiikkaan, eri muunnelmien soveltuvuus tietyn tyyppisten ongelmien ratkaisuun ja sellaisten parametrien kuin populaation koko, suoritus aika, alustusstrategia, satunnaislukujen jakauma ja naapurustotopologia vaikutus algoritmin toimintaan.

Viiteluettelo

- [Angeline, 1998] Peter J. Angeline, Evolutionary optimization versus particle swarm optimization: philosophy and performance differences. In: *Proc. of 7th International Conference on Evolutionary Programming VII, Lecture Notes in Computer Science* **1447** (1998), Springer, 601–610.
- [Applegate *et al.*, 1998] David Applegate, Robert Bixby, Vašek Chvátal and William Cook, On the solution of traveling salesman problems. *Documenta Mathematica, Extra Volume ICM III*, (Aug. 1998), 645–656.
- [Applegate *et al.*, 2007] David L. Applegate, Robert E. Bixby, Vašek Chvátal and William J. Cook, *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2007.
- [Banks *et al.*, 2007] Alec Banks, Jonathan Vincent and Chukwudi Anyakoha, A review of particle swarm optimization. Part I: background and development. *Natural Computing* **6**, 4 (Dec. 2007), 467–484.
- [Banks *et al.*, 2008] Alec Banks, Jonathan Vincent and Chukwudi Anyakoha, A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Natural Computing* **7**, 1 (Mar. 2008), 109–124.
- [Clerc, 2000] Maurice Clerc, Discrete particle swarm optimization illustrated by the traveling salesman problem. February 2000. Available as: http://clerc.maurice.free.fr/ps0/ps0_tsp/Discrete_PSO_TSP.htm. Checked 9.12.2009.
- [Clerc and Kennedy, 2002] Maurice Clerc and James Kennedy, The particle swarm–explosion, stability and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* **6**, 1 (Feb. 2002), 58–73.
- [Cook and Seymour, 2003] William Cook and Paul Seymour, Tour merging via branch-decomposition. *Inform Journal on Computing* **15**, 3 (Jul. 2003), 233–248.
- [Eberhart and Kennedy, 1995] Russell Eberhart and James Kennedy, A new optimizer using particle swarm theory. In: *Proc. of the Sixth International Symposium on Micro Machine and Human Science*, 39–43.
- [Eberhart and Shi, 1998] Russell C. Eberhart and Yuhui Shi, Comparison between genetic algorithms and particle swarm optimization. In: *Proc. of 7th*

International Conference on Evolutionary Programming VII, Lecture Notes in Computer Science **1447** (1998), Springer, 611–616.

- [Eberhart and Shi, 2000] Russell Eberhart and Yuhui Shi, Comparing inertia weights and constriction factors in particle swarm optimization. In: *Proc. of the 2000 Congress on Evolutionary Computation*, 84–88.
- [Goldbarg *et al.*, 2006] Elizabeth F. Gouvêa Goldbarg, Givanaldo R. de Souza and Marco César Goldbarg, Particle swarm for the travelling salesman problem. In: *Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science* **3906** (2006), Springer, 99–110.
- [Held and Karp, 1962] Michael Held and Richard M. Karp, A dynamic approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics* **10**, 1 (Mar. 1962), 196–210.
- [Held and Karp, 1970] Michael Held and Richard M. Karp, The traveling-salesman problem and minimum spanning trees. *Operations Research* **18**, 6 (Nov.–Dec. 1970), 1138–1162.
- [Heppner and Grenander, 1990] F. Heppner and U. Grenander, A stochastic nonlinear model for coordinated bird flocks. In: S. Krasner (ed.), *The Ubiquity of Chaos*. American Association for the Advancement of Science, 1990.
- [Johnson] David S. Johnson, TSP Challenge: Results page, [www-page, http://www2.research.att.com/~dsj/chtsp/results.html](http://www2.research.att.com/~dsj/chtsp/results.html). August 2002 Checked 9.12.2009.
- [Johnson, 1990] David S. Johnson, Local optimization and the traveling salesman problem. In: *Automata, Languages and Programming, Lecture Notes in Computer Science* **443** (1990), Springer, 446–461.
- [Johnson and McGeoch, 1997] David S. Johnson and Lyle A. McGeoch, The traveling salesman problem: a case study in local optimization. In: E. H. L. Aarts and J. K. Lenstra (eds.), *Local Search in Combinatorial Optimization*. John Wiley & Sons, 1997, 215–310.
- [Kennedy and Eberhart, 1995] James Kennedy and Russell Eberhart, Particle swarm optimization. In: *Proc. of the IEEE International Conference on Neural Networks, 1995*, 1942–1948.
- [Kennedy and Eberhart, 1997] James Kennedy and Russell Eberhart, A discrete binary version of the particle swarm algorithm. In: *Proc. of the 1997 IEEE International Conference on Systems, Man, and Cybernetics*, 4104–4108.
- [Kennedy and Eberhart, 2001] James Kennedy and Russell Eberhart, *Swarm Intelligence*. Morgan Kaufmann, 2001.
- [Kennedy and Mendes, 2002] James Kennedy and Rui Mendes, Population structure and particle swarm performance. In: *Proc. of Congress on Evolutionary Computation 2*, 1671–1676.
- [Kennedy and Spears, 1998] James Kennedy and William M. Spears, Matching algorithms to problems: an experimental test of the particle swarm and

- some genetic algorithms on the multimodal problem generator. In: *Proc. of the 1998 IEEE International Conference on Evolutionary Computation*, 78–83.
- [Khemka and Jacob, 2009] Namrata Khemka and Christian Jacob, VISPLORE: A toolkit to explore particle swarms by visual inspection. In: *Proc. of the 11th Annual Conference on Genetic and Evolutionary Computation*, 41–48.
- [Kim *et al.*, 2009] Yong-Huik Kim, Kang Hoon Lee and Yourim Yoon, Visualizing the search process of particle swarm optimization. In: *Proc. of the 11th Annual Conference on Genetic and Evolutionary Computation*, 49–56.
- [Lee] Kang Hoon Lee, Visualizing the search process of particle swarm optimization, www-page, <http://cg.kw.ac.kr/kang/pso/>. Checked 9.12.2009.
- [Li *et al.*, 2006] Xiangyong Li, Peng Tian, Jing Hua and Ning Zhong, A hybrid discrete particle swarm optimization for the traveling salesman problem. In: *Simulated Evolution and Learning, Lecture Notes in Computer Science 4247* (2006), Springer, 181–188.
- [Lin and Kernighan, 1973] S. Lin and B. W. Kernighan, An effective heuristic algorithm for the traveling-salesman problem. *Operations Research* **21**, 2 (Mar.–Apr. 1973), 498–516.
- [Liu *et al.*, 2007] Xinmei Liu, Jinrong Su and Yan Han, An improved particle swarm optimization for traveling salesman problem. In: *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, Lecture Notes in Computer Science 4682* (2007), Springer, 803–812.
- [Mendes *et al.*, 2004] Rui Mendes, James Kennedy and José Neves, The fully informed particle swarm: simpler, maybe better. *IEEE Transactions on Evolutionary Computation* **8**, 3 (Jun. 2004), 204–210.
- [Parsopoulos, 2009] Konstantinos E. Parsopoulos, Cooperative micro-particle swarm optimization. In: *Proc. of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 467–474.
- [Poli *et al.*, 2007] Riccardo Poli, James Kennedy and Tim Blackwell, Particle swarm optimization – An overview. *Swarm Intelligence* **1**, 1 (Jun. 2007), 33–57.
- [Reinelt, 1991] Gerhard Reinelt, TSPLIB–A traveling salesman problem library. *Orsa Journal on Computing* **3**, 4 (1991), 376–384.
- [Reynolds, 1987] Craig W. Reynolds, Flocks, herds and schools: A distributed behavioral model. In: *SIGGRAPH '87: Proc. of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, 25–34.
- [Sammon, 1969] John W. Sammon Jr., A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers* **18**, 5 (May 1969), 401–409.
- [Shi and Eberhart, 1998a] Yuhui Shi and Russell Eberhart, A modified particle swarm optimizer. In: *Proc. of the 1998 IEEE International Conference on Evolutionary Computation*, 69–73.

- [Shi and Eberhart, 1998b] Yuhui Shi and Russell C. Eberhart, Parameter selection in particle swarm optimization. In: *Proc. of the 7th International Conference on Evolutionary Programming*, 591–600.
- [Shi and Eberhart, 2009] Yuhui Shi and Russ Eberhart, Monitoring of particle swarm optimization. *Frontiers of Computer Science in China* **3**, 1 (Mar. 2009), 31–37.
- [Shi *et al.*, 2007] X.H. Shi, Y.C. Liang, H.P. Lee, C. Lu and Q.X. Wang, Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information Processing Letters* **103**, 5 (Aug. 2007), 169–176.
- [Shuang *et al.*, 2009] Bing Shuang, Jiapin Chen and Zhenbo Li, Study on hybrid PS-ACO algorithm. Accepted for publication in *Applied Intelligence*, 2009.
- [van der Bergh and Engelbrecht, 2006] F. van den Bergh, A.P. Engelbrecht, A study of particle swarm optimization particle trajectories. *Information Sciences* **176**, 8 (Apr. 2006), 937–971.
- [Wolpert and Macready, 1995] David H. Wolpert and William G. Macready, No Free Lunch Theorems for Search. Santa Fe Institute, Technical Report **SFI-TR-95-02-010**, February 1995. Also available as <http://www.santafe.edu/research/publications/workingpapers/95-02-010.pdf>

Laskennallinen näkökulma Riemannin ζ -funktioon

Henry Joutsijoki

Tiivistelmä

Riemannin ζ -funktio on yksi analyyttisen lukuteorian ja kompleksianalyysin keskeisimpiä funktioita. Tässä tutkielmassa esitellään ζ -funktion perusominaisuuksia. Lisäksi tutkielmassa käydään läpi laskentamenetelmiä $\zeta(n)$:n laskemiseksi, kun n on kokonaisluku.

Avainsanat: Kompleksianalyysi, Riemannin ζ -funktio, analyyttinen lukuteoria, laskentamenetelmät.

CR-luokat: F.2.1

1 Johdanto

Riemannin ζ -funktio on yksi harvoista matemaattisista funktioista, jotka ovat nousseet miltei myyttiseen maineeseen. Siitä on kirjoitettu populaari-teoksia ja se on esiintynyt myös televisiosarjassa. Kaiken tämän julkisuuden takana on Riemannin vuonna 1859 esittämä ζ -funktioita koskeva otaksuma Riemannin hypoteesi, joka on pysynyt ratkaisemattomana jo 150 vuotta ja se valittiin yhdeksi Millennium-ongelmaksi, jonka ratkaisemisesta saa miljoona dollaria. Hypoteesin esittämisen jälkeen ζ -funktion nollakohtia tutkittiin aluksi kynä-ja-paperi-menetelmällä, ja tietokoneiden keksimisen ja kehittymisen myötä laskennallinen tarkastelu on nopeutunut huomattavasti.

Nykyään nollakohtien laskeminen on hyvin suosittua, mutta sen rinnalla tutkitaan ahkerasti myös uusia tehokkaita laskentatapoja $\zeta(n)$:n arvoille, kun n on kokonaisluku. Tässä tutkielmassa on kaksi suurempaa kokonaisuutta. Esitieto-luvun jälkeen kolmannessa luvussa käydään läpi ζ -funktioon liittyviä perustuloksia kuten Riemannin funktionaaliyhtälö ja ζ -funktion yhtenä sovelluksena alkulukulause. Lisäksi hahmotamme ζ -funktion ja alkulukujen välistä yhteyttä. Toinen kokonaisuus muodostuu neljännestä luvusta, jossa tarkastelemme $\zeta(n)$:n laskentatapoja kokonaislukutapauksissa. Tällöin esitämme algoritmeja ζ -funktion arvojen laskemiseen ja näemme kiinnostavan ζ -funktion ja Bernoullin lukujen välillä.

Lukijalta vaaditaan tutkielman sisällön ymmärtämiseksi vahvaa maattista perusosaamista niin kompleksianalyysin kuin analyttisen luku-teorian saroilta. Yksittäistä päälähdettä ei tutkielmassa ole, vaan tutkielma koostuu usean lähteen tiedoista.

2 Esitiedot

ζ -funktion teoria on kiinteässä yhteydessä alkulukujen teorian kanssa, joten ensimmäiseksi määrittelemme alkuluvun ja yhdistetyn luvun käsitteet. Näiden jälkeen esittelemme Eulerin keksimän Γ -funktion, joka esiintyy usein ζ -funktion liittyvissä tuloksissa.

Määritelmä 2.1. Luku ($p > 1$) on *alkuluku*, jos sen ainoat positiiviset tekijät ovat 1 ja p .

Määritelmä 2.2. Positiivinen kokonaisluku, joka on lukua 1 suurempi ja ei ole alkuluku, on *yhdistetty luku*.

Lause 2.1 (Weierstrassin M-testi). *Oletetaan, että sarjan $\sum_{n=1}^{\infty} f_n$ jokainen termi on määritelty joukossa A . Jos on olemassa sellainen reaali-termien jono (M_n) , että $|f_n(z)| \leq M_n$ pätee kaikilla $z \in A$ ja sarja $\sum_{n=1}^{\infty} M_n$ suppenee, niin $\sum_{n=1}^{\infty} f_n$ suppenee itseisesti ja tasaisesti A :ssa.*

Todistus. Ks. [8, ss. 253-254].

Määritelmä 2.3. Määrittelemme *Eulerin gammafunktion* $\Gamma(z)$ kaavalla

$$\Gamma(z) = \frac{e^{-\gamma z}}{z} \prod_{n=1}^{\infty} \left(1 + \frac{z}{n}\right)^{-1} e^{\frac{z}{n}},$$

kun $\gamma = \lim_{n \rightarrow \infty} (1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} - \log n) \approx 0.577$ on *Eulerin vakio*.

Lause 2.2. *Jos $z \neq 0, -1, -2, \dots$, niin $\Gamma(z + 1) = z\Gamma(z)$.*

Todistus. Ks. [5, s. 10].

Lause 2.3. *Eulerin Γ -funktiolle pätee*

$$\Gamma(z)\Gamma(1 - z) = \frac{\pi}{\sin(\pi z)}.$$

Todistus. Ks. [8, ss. 504-505].

Residylause on erinomainen apuväline vaikeiden integraalien laskemiseen. Lisäksi sitä soveltamalla pystytään määrittämään tiettyjen päättymättömien sarjojen arvoja. Erityisesti sitä käytetään Riemannin funktionaaliyhtälön todistamisessa. Lisätietoa residylauseesta ja sen lukuisista sovelluksista löytää pro gradu -tutkielmastani [4].

Määritelmä 2.4. Olkoon U avoin joukko. Silmukka σ on *nollahomologinen* joukossa U , jos silmukan kierrosluku $n(\sigma, z) = 0$ kaikilla $z \notin U$.

Lause 2.4 (Residylause). *Olkoon funktio f holomorfinen avoimessa joukossa U paitsi eristetyissä erikoispisteissä. Olkoot σ silmukka joukossa $U \setminus E$ ja nollahomologinen joukossa U . Tällöin*

$$\int_{\sigma} f(z) dz = 2\pi i \sum_{z \in E} n(\sigma, z) \operatorname{Res}(z, f).$$

Todistus. Ks. [4, ss. 16-17].

3 ζ -funktioon liittyviä perustuloksia

Toiston välttämiseksi sovimme, että $\Re s = \sigma$ ja $\Im s = t$.

Määritelmä 3.1. Määrittelemme *Riemannin ζ -funktion* sarjan

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

summana niillä arvoilla s , joilla sarja suppenee. Näin on, kun $\sigma > 1$.

Määritelmässä esiintyvä suppenemisarve pystytään perustelemaan hyvin helposti (vrt. [6, s. 315]). Oletetaan, että $\sigma > 1$. Näin ollen

$$\left| \frac{1}{n^s} \right| = \frac{1}{|e^{s \log n}|} = \frac{1}{e^{\sigma \log n}} = \frac{1}{n^{\sigma}}.$$

Sarja $\sum_{n=1}^{\infty} \frac{1}{n^{\sigma}}$ suppenee, kun $\sigma > 1$. Näin ollen sarja $\sum_{n=1}^{\infty} \frac{1}{n^s}$ suppenee itseisesti. Sarja suppenee tasaisesti puolitasossa $\sigma \geq \sigma_0 > 1$, koska $\frac{1}{n^{\sigma}} \leq \frac{1}{n^{\sigma_0}}$ ja sarja $\sum_{n=1}^{\infty} \frac{1}{n^{\sigma_0}}$ suppenee. Täten Weierstrassin M -testistä seuraa sarjan tasainen suppeneminen.

Lause 3.1 (Eulerin tulokaava). *Olkoon (p_n) kaikkien alkulukujen muodostama jono. Jos $\sigma > 1$, niin*

$$\frac{1}{\zeta(s)} = \prod_{n=1}^{\infty} (1 - p_n^{-s}).$$

Todistus (vrt. [6, ss. 315-317]). Oletamme aluksi, että $s \in \mathbb{R}$ ja $s > 1$. Tarkastelemme lauseketta $\frac{1}{1-\frac{1}{p^s}}$, kun p on alkuluku. Pystymme esittämään lausekkeelle sarjakehitelmän $\frac{1}{p}$:n potenssien mukaan. Siis

$$\frac{1}{1-\frac{1}{p^s}} = 1 + \frac{1}{p^s} + \frac{1}{p^{2s}} + \dots$$

Soveltamalla tätä yhteyttä alkulukujen arvoilla $p_1 = 2, p_2 = 3, p_3 = 5, \dots$ ja muodostamalla tulon

$$\prod_{\nu=1}^m \frac{1}{1-\frac{1}{p_\nu^s}}$$

saamme Cauchyn kertosäännön perusteella, kun $m = 2$,

$$(3.1) \quad \frac{1}{1-\frac{1}{p_1^s}} \frac{1}{1-\frac{1}{p_2^s}} = \sum_{\nu_1, \nu_2=0}^{\infty} \frac{1}{(p_1^{\nu_1} p_2^{\nu_2})^s},$$

kun ν_1 ja ν_2 saavat toisistaan riippumatta kaikki ei-negatiiviset kokonaislukuarvot. Täydellisellä induktiolla m :n suhteen voimme yleistää kaavan (3.1), jolloin

$$(3.2) \quad \prod_{\nu=1}^m \frac{1}{1-\frac{1}{p_\nu^s}} = \sum_{\nu_1, \nu_2, \dots, \nu_m=0}^{\infty} \frac{1}{(p_1^{\nu_1} p_2^{\nu_2} \dots p_m^{\nu_m})^s},$$

kun vastaavasti $\nu_1, \nu_2, \dots, \nu_m$ saavat kaikki ei-negatiiviset kokonaislukuarvot.

Koska jokaisella kokonaisluvulla on yksikäsitteinen alkutekijäesitys, voimme esittää yhtälön (3.2) muodossa

$$(3.3) \quad \prod_{\nu=1}^m \frac{1}{1-\frac{1}{p_\nu^s}} = \sum \frac{1}{n^s},$$

missä n saa ne positiiviset kokonaislukuarvot, joilla ei ole suurempaa tekijää kuin p_m . Kaava (3.3) on osasumma $\zeta(s)$:n kehitelmästä, joten se on pienempi kuin $\zeta(s)$. Toisaalta summassa on mukana termit $1, \frac{1}{2^s}, \frac{1}{3^s}, \dots, \frac{1}{p_m^s}$, jolloin jokaisella m :n arvolla on voimassa kaksoisepähtälö

$$\sum_{n=1}^{p_m} \frac{1}{n^s} < \prod_{\nu=1}^m \frac{1}{1-\frac{1}{p_\nu^s}} < \zeta(s).$$

Kun $m \rightarrow \infty$, niin myös $p_m \rightarrow \infty$, joten vasen puoli lähenee raja-arvoa $\zeta(s)$.

Näin ollen tulo

$$\prod_{\nu=1}^{\infty} \frac{1}{1-\frac{1}{p_\nu^s}}$$

suppenee ja sen arvo on $\zeta(s)$. Siis

$$(3.4) \quad \zeta(s) = \prod_p \frac{1}{1 - \frac{1}{p^s}},$$

kun p käy läpi kaikki alkuluvut.

Olemme todistaneet väitteen nyt siinä erikoistapauksessa, kun s on reaalinen ja > 1 . Seuraavaksi tutkimme kaavan (3.4) suppenemista, kun s on kompleksinen. Aluksi esitämme tulon muodossa

$$(3.5) \quad \prod_p \left(1 + \frac{1}{p^s - 1}\right).$$

Eräs välttämätön ja riittävä ehto päättymättömän tulon $\prod_{n=1}^{\infty} (1 + a_n)$ suppenemiselle on se, että sarja $\sum_{n=1}^{\infty} |a_n|$ suppenee (ks. todistus [1, ss. 192-193]). Koska kolmioepäyhtälöä käyttämällä on voimassa

$$\sum_p \left| \frac{1}{p^s - 1} \right| \leq \sum_p \frac{1}{p^\sigma - 1},$$

voimme päätellä, että tulo suppenee, jos sarja

$$\sum_p \frac{1}{p^\sigma - 1}$$

suppenee. Tämä sarja suppenee tasaisesti, kun $\sigma \geq \sigma_0 > 1$. Näin ollen kaava (3.5) määrittelee säännöllisen analyttisen funktion puolitasossa $\sigma > 1$. Koska yhtälön (3.4) kumpikin puoli on säännöllinen analyttinen funktio puolitasossa $\sigma > 1$ ja ne ovat reaalilla s :n arvoilla yhtäsuuret, ovat ne koko puolitasossa yhtäsuuret.

Laskennallisesta näkökulmasta tarkasteltuna ζ -funktion määritelmä ja Eulerin tulokaava ovat varsin hankalia, sillä kumpikin niistä sisältää äärettömän määrän termejä. Koska tietokoneen laskentatarkkuus ja -kapasiteetti ovat äärellisiä, ei tarkkoja numeerisia arvoja pystytä ζ -funktiolle määritelmän avulla saamaan. Näin ollen tietokoneella laskettuna numeerisiin arvoihin jää väkisinkin pyöristysvirhettä. Sen sijaan teoreettisesti pystytään todistamaan useille ζ -funktion arvoille tarkka arvo. Esimerkiksi, jos $s = 2$ tai $s = 4$, on voimassa, että $\zeta(2) = \frac{\pi^2}{6}$ ja $\zeta(4) = \frac{\pi^4}{90}$.

Eulerin tulokaavassa nähdään mielenkiintoinen yhteys ζ -funktion ja alkulukujen kesken. Mietittäessä, miten voitaisiin laskea ζ -funktion arvoja Eulerin tulokaavan avulla, on ensin haettava keino alkulukujen seulomiseen.

Ensimmäisen algoritmin alkulukujen seulomiseen esitti antiikin aikana Eratosthenes, jonka kehittämää metodia sanotaan Eratostheneen seulaksi. Esitämme seuraavaksi tämän algoritmin [7].

Algoritmi 1. *Etsitään alkuluvut Eratostheneen seulan avulla. Syötteenä on luku n ($n \geq 2$).*

```

Eratosthenes( $n$ )
 $a[1] := 0$ 
for  $i := 2$  to  $n$  do
     $a[i] := 1$ 
end for
while  $p^2 \leq n$  do
     $j := p^2$ 
    while  $j \leq n$  do
         $a[j] := 0$ 
         $j := j + p$ 
    end while
    repeat
         $p := p + 1$ 
    until  $a[p] = 1$ 
end while
return ( $a$ )

```

Algoritmi on aikakompleksisuudeltaan $O(n(\log n) \log \log n)$ [7], mutta huonona puolena on tilavaatimus, mikä on $O(n)$. Eratostheneen seulan toimintaperiaate on se, että ensin listataan positiiviset kokonaisluvut väliltä $[2, n]$. Ensimmäinen luku eli 2 on alkuluku, joten sitä ei merkitä. Tämän jälkeen haetaan kaikki luvun 2 monikerrat, jotka ovat pienempiä tai yhtä suuria kuin n ja ne merkitään. Toisessa vaiheessa ensimmäinen tutkimatton luku on kolme, joka on myös alkuluku, joten sitä ei merkitä, mutta sen monikerrat merkitään. Tätä prosessia jatketaan, kunnes kaikki taulukon luvut on käyty läpi. Kaikki merkkeamattomat luvut ovat alkulukuja. Esitettyssä algoritmisssa alkuluvut ovat ne taulukon indeksit, joita vastaavat taulukon arvot ovat 1.

Jatketaan vielä alkulukuteeman parissa. Seuraavaksi esitämme kaksi tulosta, joiden perusteella saadaan arvio sille, kuinka monta operaatiota tarvitaan sen selvittämiseen, onko luku alkuluku vai yhdistetty luku. Todistetaan

ensin pieni lukuteoreettinen tulos, jonka perusteella luku todistetaan yhdistetyksi luvuksi, kun tutkitaan, onko luku n jaollinen lukua \sqrt{n} pienemmillä alkuluvuilla.

Lause 3.2. *Jos n on yhdistetty luku, niin n :llä on alkulukutekijä, joka on $\leq \sqrt{n}$.*

Todistus (vrt. [11, s. 67]). Koska n on yhdistetty luku, voimme kirjoittaa $n = ab$, kun $1 < a \leq b < n$. Näin ollen $a \leq \sqrt{n}$, sillä muutoin olisi voimassa $b \geq a > \sqrt{n}$ ja $ab > \sqrt{n}\sqrt{n} = n$. Koska jokaisella lukua 1 suuremmalla kokonaisluvulla on alkulukutekijä, a :n täytyy olla luvun n alkulukutekijä.

Lause 3.3. *Olkoon $n \in \mathbb{N}$ ja $n \geq 2$. Jos n on yhdistetty luku, niin se voidaan osoittaa ajassa $O((\log_2 n)^2)$.*

Todistus (vrt. [11, s. 343]). Jos n on yhdistetty luku, niin on olemassa sellaiset kokonaisluvut a ja b , että $1 < a < n$ ja $1 < b < n$ ja lisäksi $n = ab$. Tällöin kertomalla luvut a ja b ja todentamalla, että $n = ab$ kuuluu $O((\log_2 n)^2)$ todistaaksemme n :n yhdistetyksi luvuksi.

Lause 3.4. *Olkoon $n \in \mathbb{N}$ ja $n \geq 2$. Jos n on alkuluku, niin se voidaan todistaa ajassa $O((\log_2 n)^4)$.*

Todistus. Ks. [11, ss. 343-344].

Tähän mennessä olemme käsitelleet Riemannin ζ -funktiota puolitasossa $\sigma > 1$. Pystymme laajentamaan määrittelyjoukon aina koko kompleksitasoon lukuun ottamatta pistettä $s = 1$, missä ζ -funktiolla on yksinkertainen napa. Tämä tapahtuu analyyttisen jatkamisen periaatteen avulla.

Lause 3.5. *Funktio $\zeta(s)$ voidaan jatkaa analyyttisesti puolitasoon $\sigma \leq 1$.*

Todistus. Ks. [6, ss. 319-321].

Nyt, kun ζ -funktio on jatkettu analyyttisesti koko kompleksitasoon, pystytään laskemaan ilman rajoituksia ζ -funktion arvoja koko kompleksitasossa. Eulerin tulokaavan perusteella on mahdollista laskea puolitasossa $\sigma > 1$ ζ -funktion arvoja. Sen sijaan puolitasossa $\sigma \leq 1$ meillä ei ole vielä minkäänlaista kaavaa tai keinoa, jonka perusteella voitaisiin laskea ζ -funktion arvoja. Tämän ongelman poistamiseksi Riemann kehitti kuuluisan funktionaaliryhtälönsä, jonka avulla saatiin yhteys arvojen $\zeta(s)$:n ja $\zeta(1-s)$:n välille.

Samalla muodostettiin kiinnekohta puolitasojen $\sigma > 1$ ja $\sigma \leq 0$ välille. Lisäksi alueelle $0 \leq \sigma \leq 1$ päästiin käsiksi funktionaaliyhtälön perusteella. Tämän alueen merkitystä selvitämme myöhemmin. Seuraava todistus löytyy tarkastuksessa olevasta liseniaattitutkielmastani [5], jossa esitän muun muassa funktionaaliyhtälölle viisi todistusta. Todistus on peräisin Titchmarshin kirjasta [12], jossa Titchmarsh esittää funktionaaliyhtälölle peräti kahdeksan todistusta.

Lause 3.6 (Riemannin funktionaaliyhtälö). *Funktio $\zeta(s)$ on säännöllinen kaikilla $s \in \mathbb{C}$ lukuun ottamatta pistettä $s = 1$, missä sillä on yksinkertainen napa ja residy on 1. Se toteuttaa funktionaaliyhtälön*

$$(3.6) \quad \zeta(s) = 2^s \pi^{s-1} \sin \frac{\pi s}{2} \Gamma(1-s) \zeta(1-s).$$

Todistus (vrt. [12, ss. 18-20]). Olkoon

$$I(s) = \int_C \frac{z^{s-1}}{e^z - 1} dz.$$

Polku C alkaa positiivisesta äärettömyydestä reaaliakselilta. Reaaliakselia pitkin tullaan mielivaltaisen lähelle origoa, jonka jälkeen kierretään origo keran positiiviseen kulkusuuntaan käymättä minkään joukon $\{\pm 2\pi i, \pm 4\pi i, \dots\}$ pisteiden kautta. Lopuksi palataan takaisin reaaliakselilla positiiviseen äärettömyyteen. Integraalissa kompleksiluvun z^{s-1} määrittelemme logaritmin avulla $e^{(s-1)\log z}$, kun logaritmi on reaalinen integrointipolun alussa. Täten $\Im(\log z)$ on välillä $[0, 2\pi]$. Voimme valita C :n ulottumaan positiivisesta äärettömyydestä ρ :hon, kun $0 < \rho < 2\pi$, sekä ympyränkaareksi $|z| = \rho$ ja lopuksi reaaliakselilla palataan ρ :sta kohti äärettömyyttä. Siis

$$\int_C \frac{z^{s-1}}{e^z - 1} dz = \int_{-\infty}^{\rho} \frac{x^{s-1}}{e^x - 1} dx + \int_{|z|=\rho} \frac{z^{s-1}}{e^z - 1} dz + \int_{\rho}^{\infty} \frac{x^{s-1}}{e^x - 1} dx.$$

Ympyrällä on voimassa

$$|z^{s-1}| = e^{(\sigma-1)\log|z| - t \arg z} \leq |z|^{\sigma-1} e^{2\pi|t|}.$$

Lisäksi, koska

$$\frac{z}{e^z - 1} = \sum_{n=0}^{\infty} \frac{B_n}{n!} z^n \quad (\text{ks. [10, s. 19]}),$$

on $|e^z - 1| > A|z|$, kun A on vakio ja sarjan suppenemissäde on 2π . Näin ollen

$$\int_{|z|=\rho} \frac{z^{s-1}}{e^z - 1} dz \rightarrow 0,$$

kun $\rho \rightarrow 0$ ja $\sigma > 1$. Siis

$$\begin{aligned}
 I(s) &= - \int_0^\infty \frac{x^{s-1}}{e^x - 1} dx + \int_0^\infty \frac{(xe^{2\pi i})^{s-1}}{e^x - 1} dx \\
 &= (e^{2\pi i s} - 1)\Gamma(s)\zeta(s) = \frac{\pi(e^{2\pi i s} - 1)\zeta(s)}{\sin \pi s \Gamma(1 - s)} \\
 &= \frac{2\pi i \zeta(s)}{\Gamma(1 - s)} \cdot \frac{e^{2\pi i s} - 1}{e^{\pi s i} - e^{-\pi s i}} \\
 (3.7) \quad &= \frac{2\pi i e^{\pi i s}}{\Gamma(1 - s)} \zeta(s).
 \end{aligned}$$

Täten

$$(3.8) \quad \zeta(s) = \frac{e^{-i\pi s} \Gamma(1 - s)}{2\pi i} \int_C \frac{z^{s-1}}{e^z - 1} dz,$$

kun $\sigma > 1$. Voimme siis esittää integraalin $I(s)$ kahden epäoleellisen integraalin summana. Tämä esitys on tasaisesti suppeneva jokaisessa kompleksitason äärellisessä alueessa, joten kaava (3.8) jatkaa ζ -funktion analyyttisesti koko kompleksitasoon. Ainoat erikoispisteet ovat funktion $\Gamma(1 - s)$ yksinkertaiset navat, kun $s = 1, 2, 3, \dots$. Koska $\zeta(s)$ on säännöllinen pisteissä $s = 2, 3, \dots$, joissa $I(s) = 0$ Cauchyn integraalilauseen perusteella, ainoaksi erikoispisteeksi jää yksinkertainen napa pisteessä $s = 1$. Nyt

$$I(1) = \int_C \frac{dz}{e^z - 1} = 2\pi i$$

ja

$$\Gamma(1 - s) = -\frac{1}{s - 1} + \dots$$

Täten kaavan (3.8) nojalla ζ -funktion residy pisteessä $s = 1$ on 1. Jos $s \in \mathbb{Z}$, integraalin $I(s)$ integroitava on yksikäsitteinen ja $I(s)$ voidaan laskea residylauseen perusteella.

Johtaaksemme funktionaaliyhtälön kaavasta (3.8) tulee integraali integroida pitkin polkua C_n , joka kulkee positiivisesta äärettömyydestä reaaliakselia pitkin pisteeseen $(2n + 1)\pi$ ja kiertää janat $(2n + 1)\pi(\pm 1 \pm i)$, jonka jälkeen se palaa takaisin positiiviseen äärettömyyteen pitkin reaaliakselia. Kaarimonikulmioiden C ja C_n välissä integroitavalla on navat pisteissä $\pm 2\pi i, \dots, \pm 2n\pi i$. Pisteissä $2m\pi i$ ja $-2m\pi i$ residyjen summa on

$$\begin{aligned}
 (2m\pi e^{\frac{\pi i}{2}})^{s-1} + (2m\pi e^{\frac{3\pi i}{2}})^{s-1} &= (2m\pi)^{s-1} e^{i\pi(s-1)} 2 \cos \frac{1}{2}\pi(s-1) \\
 &= -2(2m\pi)^{s-1} e^{i\pi s} 2 \sin \frac{1}{2}\pi s.
 \end{aligned}$$

Näin ollen residylauseen perusteella

$$I(s) = \int_{C_n} \frac{z^{s-1}}{e^z - 1} dz + 4\pi i e^{i\pi s} \sin \frac{\pi s}{2} \sum_{m=1}^n (2m\pi)^{s-1}.$$

Olkoot $\sigma < 0$ ja $n \rightarrow \infty$. Funktio $\frac{1}{e^z - 1}$ on rajoitettu polun C_n sisällä ja $z^{s-1} = O(|z|^{\sigma-1})$. Täten integraali pitkin C_n :ää lähenee nollaa, jolloin saamme

$$\begin{aligned} I(s) &= 4\pi i e^{i\pi s} \sin \frac{\pi s}{2} \sum_{m=1}^{\infty} (2m\pi)^{s-1} \\ &= 4\pi i e^{i\pi s} \sin \frac{\pi s}{2} (2\pi)^{s-1} \zeta(1-s). \end{aligned}$$

Nyt funktionaaliyhtälö seuraa välittömästi yhdistämällä saatu tulos kaavan (3.7) kanssa.

Seurauslause 3.1. ζ -funktiolla on voimassa $\zeta(-2n) = 0$, kun $n \in \mathbb{N}$.

Todistus. Funktionaaliyhtälön perusteella

$$\zeta(-2n) = 2^{-2n} \pi^{-2n-1} \sin(-\pi n) \Gamma(2n+1) \zeta(2n+1).$$

Koska $\sin(-\pi n) = 0$ kaikilla $n \in \mathbb{Z}$, väite seuraa välittömästi.

Pisteitä $s = -2n$ sanotaan ζ -funktion triviaaleiksi nollakohdiksi. Nämä pisteet eivät herätä suurempaa mielenkiintoa, vaan pääpaino kohdistuu ζ -funktion ei-triviaaleihin nollakohtiin. Riemannin ζ -funktio on ollut jo pitkän aikaa laskennallisen tutkimuksen kohteena. Erityisesti suora $\sigma = \frac{1}{2}$, jota kutsutaan myös *kriittiseksi suoraksi*, on ollut tarkemman tarkastelun alaisena, koska se on *Riemannin hypoteesin* kannalta keskeisin osa-alue. Itse asiassa Riemannin hypoteesi on juuri se asia, minkä vuoksi ζ -funktio on ollut jo 150 vuotta aktiivisen tutkimuksen alaisena. ζ -funktion ei-triviaaleja nollakohtia on haettu tietokoneen avulla jo pitkän aikaa ja kaikki löydetty nollakohdat ovat olleet suoralla $\sigma = \frac{1}{2}$. Siis laskennallinen tutkimus vahvistaa käsitystä Riemannin hypoteesin paikkansapitävyydestä. Kuitenkin on todistettu, että ζ -funktiolla on kriittisellä suoralla ääretön määrä nollakohtia. Näin ollen laskennallisesti ei voida koskaan todistaa täydellisesti Riemannin hypoteesia, vaan se pitää teoreettisesti todistaa oikeaksi tai vääräksi.

Otaksuma 1 (Riemannin hypoteesi). *Kaikki Riemannin ζ -funktion ei-triviaalit nollakohdat sijaitsevat suoralla $\sigma = \frac{1}{2}$.*

Määritelmä 3.2. Alkulukujen kertymäfunktio $\pi(x)$ ilmaisee alkulukujen määrän, jotka ovat $\leq x$.

Ennen kuin seuraavassa luvussa ryhdymme tarkastelemaan ζ -funktiota kokonaislukuarvoilla, on syytä vielä palata ζ -funktion ja alkulukujen väliseen yhteyteen. Yksi kuuluisimmista ja merkittävimmistä tuloksista analyttisen lukuteorian saralla on *alkulukulause*. Sen todistaminen oli Riemannin yksi päätavoitteista, mutta hän ei sitä saanut todistettua, vaan sen todistivat Hadamard ja de la Vallée Poussin vuonna 1896. Alkulukulauseessa todistaan asymptoottinen yhtäsuuruus alkulukujen kertymäfunktiolle $\pi(x)$ ja lausekkeelle $\frac{x}{\log x}$. Käytämme asymptoottiselle yhtäsuuruudelle merkintää \sim eli

$$f(x) \sim g(x), \quad \text{jos} \quad \frac{f(x)}{g(x)} \rightarrow 1, \quad \text{kun} \quad x \rightarrow \infty.$$

Funktiolle $\pi(x)$ pystytään esittämään myös toinen asymptoottinen yhtäsuuruus, sillä Gauss esitti, että funktio

$$\text{Li}(x) = \int_0^x \frac{dt}{\log t}$$

on asymptoottisesti yhtäsuuri $\pi(x)$:n kanssa. Huomattavaa on, että kirjallisuudessa joskus $\text{Li}(x)$:n määrittelyssä esiintyy integraalin alarajana luku 2 nollan asemesta. Tällöin $\text{Li}(x)$:n integraalien arvot eroavat toisistaan vakioilla. Näin ollen saadaan yhteys

$$\pi(x) \sim \text{Li}(x) \int_0^x \frac{dt}{\log t} \sim \frac{x}{\log x}.$$

Lause 3.7 (Alkulukulause). *Kun $x \rightarrow \infty$, niin*

$$\pi(x) \sim \frac{x}{\log x}.$$

Todistus. Ks. [5, ss. 43-45].

Erastotheneen seulan yhteydessä jo huomattiin, että alkulukujen seulon ja yleisesti ottaen luvun tutkiminen alkuluvuksi on haastava tehtävä. Näin ollen alkulukulauseen perusteella saammme yksinkertaisen tavan arvioida $\pi(x)$:n arvoja suurilla x :n arvoilla. Toinen tapa $\pi(x)$:n arviointiin on laskea funktion $\text{Li}(x)$ integraali. Funktion $\text{Li}(x)$ integraalissa on se huono puoli, että sitä voidaan integroida vain numeerisesti, sillä funktion $\frac{1}{\log t}$ integraalifunktiolle ei ole olemassa eksplisiittistä kaavaa. Näin ollen määrätystä integraalista

saadaan vain likiarvoja. Tosin nämä likiarvot ovat erittäin hyviä. Taulukossa 1 näemme approksimaatioita $\pi(x)$:lle niin alkulukulauseen kuin $\text{Li}(x)$:n avulla laskettuna. $\text{Li}(x)$:n arvot on pyöristetty lähimpään kokonaislukuun.

x	$\pi(x)$	$\frac{x}{\log x}$	$\text{Li}(x)$
10^3	168	144.8	178
10^4	1229	1085.7	1246
10^5	9592	8685.9	9630
10^6	78498	72382.4	78628
10^7	664579	620420.7	664918
10^8	5761455	5428681.0	5762209
10^9	50847534	48254942.4	50849235
10^{10}	455052512	434294481.9	455055614

Taulukko 1. Approksimaatioita funktion $\pi(x)$ arvoille [11, s. 72]

4 $\zeta(n)$, kun $n \in \mathbb{Z}$

Tässä luvussa paneudumme tarkastelemaan $\zeta(s)$:n arvoja, kun s on kokonaisluku. Kokonaislukutapauksille on kehitetty monenlaisia esitysmuotoja. Näitä esitysmuotoja käymme läpi ja esitämme myös joitakin algoritmeja näiden arvojen laskemiseen. Funktionaaliyhtälön seurauksena saimme todistettua seurauslauseessa 3.1, että $\zeta(-2n) = 0$, kun $n \in \mathbb{N}$. Ensimmäiseksi todistamme, että $\zeta(0) = -\frac{1}{2}$. Tätä varten tarvitsemme yleisesti käytetyn laskennallisen apuvälineen eli *Eulerin-MacLaurinin summakaavan*. Sen määritelmässä esiintyy *Bernoullin lukuja*, jotka määritellään funktion $\frac{t}{e^t-1}$ kautta. Tälle funktiolle muodostetaan sarjakehitelmä, jonka kertoimina Bernoullin luvut B_n esiintyvät. Siis

$$\frac{t}{e^t - 1} = \sum_{n=0}^{\infty} \frac{B_n}{n!} t^n.$$

Bernoullin luvuista tarkempaa tietoa löytyy esimerkiksi Rademacherin kirjasta [10].

Määritelmä 4.1. Olkoon $f(x)$ välillä $[a, b]$ jatkuvasti derivoituva funktio. *Eulerin-MacLaurinin summakaava* määritellään kaavalla

$$(4.1) \quad \sum_{n=a+1}^b f(n) = \int_a^b f(x) dx + \sum_{r=1}^q (-1)^r \frac{B_r}{r!} \{f^{(r-1)}(b) - f^{(r-1)}(a)\} + R_q,$$

kun B_r :t ovat *Bernoullin lukuja* ja jäännöstermi on

$$(4.2) \quad R_q = \frac{(-1)^{q-1}}{q!} \int_a^b B_q(x - [x]) f^{(q)}(x) dx.$$

Lause 4.1. $\zeta(0) = -\frac{1}{2}$.

Todistus (vrt. [10, ss. 80-81]). Kirjoitamme aluksi ζ -funktion määritelmän muodossa

$$\zeta(s) = 1 + \lim_{N \rightarrow \infty} \sum_{n=2}^N \frac{1}{n^s}.$$

Nyt voimme soveltaa summalausekkeeseen kaavaa (4.1), kun asetamme $f(x) = x^{-s}$, jolloin $f^{(k)}(x) = (-1)^k s(s+1) \cdots (s+k-1) x^{-s-k}$. Näin ollen saamme

$$\begin{aligned} \zeta(s) - 1 &= \lim_{N \rightarrow \infty} \left\{ \int_1^N x^{-s} dx + \frac{1}{2}(N^{-s} - 1) \right. \\ &\quad - \sum_{r=2}^q \frac{B_r}{r!} s(s+1) \cdots (s+r-2) (N^{-s-r+1} - 1) \\ &\quad \left. - \frac{1}{q!} s(s+1) \cdots (s+q-1) \int_1^N B_q(x - [x]) x^{-s-q} dx \right\}. \end{aligned}$$

Koska ζ -funktion sarjakehitelmä on itseisesti suppeneva, kun $\sigma > 1$, voimme ottaa raja-arvon termeittäin, jolloin

$$(4.3) \quad \begin{aligned} \zeta(s) &= \frac{1}{s-1} + \frac{1}{2} + \sum_{r=2}^q \frac{B_r}{r!} s(s+1) \cdots (s+r-2) \\ &\quad - \frac{1}{q!} s(s+1) \cdots (s+q-1) \int_1^\infty B_q(x - [x]) x^{-s-q} dx. \end{aligned}$$

Kaavan (4.3) integraali suppenee puolitasoissa $\sigma > 1$ sekä $\sigma > 1 - q$, ja se suppenee tasaisesti jälkimmäisen puolitason jokaisessa kompaktissa osajoukossa. Näin ollen kaava (4.3) muodostaa $\zeta(s)$:n analyyttisen jatkon puolitasoon $\sigma > 1 - q$ ja ainoa erikoispiste on napa pisteessä $s = 1$. Koska q on mielivaltainen luonnollinen luku, muodostaa kaava (4.3) $\zeta(s)$:n analyyttisen jatkon koko kompleksitasoon. Asettamalla nyt $s = 0$ kaavaan (4.3), saamme $\zeta(0) = -\frac{1}{2}$.

Lause 4.2.

$$\zeta(2n) = \frac{(-1)^{n-1}}{2} \frac{B_{2n}}{(2n)!} (2\pi)^{2n}.$$

Todistus. Ks. [3, s. 221].

Lause 4.3. $\zeta(1 - 2n) = -\frac{B_{2n}}{2n}$.

Todistus (vrt. [10, ss. 80-81]). Voimme esittää funktionaaliyhtälön muodossa

$$\xi(s) = \xi(1 - s)$$

(ks. [12, s. 16]), kun $\xi(s) = \frac{s}{2}(s-1)\pi^{-\frac{s}{2}}\Gamma(\frac{s}{2})\zeta(s)$. Näin ollen

$$\zeta(1 - s) = \frac{\pi^{-\frac{s}{2}}\Gamma(\frac{s}{2})\zeta(s)}{\pi^{\frac{s-1}{2}}\Gamma(\frac{1-s}{2})}.$$

Kun asetamme $s = 2n$, niin soveltamalla lauseita 2.2 ja 4.2 saamme

$$\begin{aligned} \zeta(1 - 2n) &= \frac{\pi^{-n}\Gamma(n)\zeta(2n)}{\pi^{n-\frac{1}{2}}\Gamma(-n + \frac{1}{2})} \\ &= \pi^{-2n+\frac{1}{2}}(n-1)! \\ &\quad \times \frac{(-n + \frac{1}{2})(-n + \frac{3}{2}) \cdots (-n + n - \frac{1}{2})}{\Gamma(-n + n + \frac{1}{2})} \\ &\quad \times \frac{(-1)^{n-1}B_{2n}2^{2n-1}\pi^{2n}}{(2n)!} = -\frac{B_{2n}}{2n}. \end{aligned}$$

Seuraavat ζ -funktion esitykset tietyillä kokonaislukuarvoilla ja niihin liittyvät algoritmit ovat peräisin Borweinien, Bradleyyn ja Crandallin artikkelista [2]. Esitämme $\zeta(2)$:n, $\zeta(3)$:n, $\zeta(4)$:n, $\zeta(5)$:n ja $\zeta(7)$:n sarjakehitelmien avulla ja osalle näistä esitämme myös algoritmin. Jokaisessa esityksessä ilmenee termi $\binom{2k}{k}$, jolla tarkoitetaan perinteistä binomikerrointa eli

$$\binom{2k}{k} = \frac{(2k)!}{k!(2k-k)!} = \frac{(2k)!}{k!k!}.$$

Ensimmäiseksi tarkastelemme $\zeta(2)$:ta ja $\zeta(4)$:a. Yleisesti ottaen parillisilla positiivisilla kokonaisluvuilla voidaan ζ -funktion arvot laskea residylausetta soveltamalla. Esimerkiksi tulos

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

voidaan todistaa integroimalla funktiota $f(z) = \pi z^{-2} \cot \pi z$ pitkin kaarimonikulmiota C , joka muodostuu janoista $(\pm 1 \pm i)(N + \frac{1}{2})$ [9, s. 247]. Integroinnin yhteydessä sovelletaan residylausetta ja lopputulos muodostuu integroitavan funktion residyyistä integrointipolun sisällä. Siis esitykselle

$$(4.4) \quad \zeta(2) = 3 \sum_{k=1}^{\infty} \frac{1}{k^2 \binom{2k}{k}}$$

saadaan tulos $\zeta(2) = \frac{\pi^2}{6}$. Vastaavalla menettelyllä kuin edellisessä tapauksessa pystytään todistamaan, että

$$\sum_{n=1}^{\infty} \frac{1}{n^4} = \frac{\pi^4}{90}.$$

Näin ollen esityksellä

$$(4.5) \quad \zeta(4) = \frac{36}{17} \sum_{k=1}^{\infty} \frac{1}{k^4 \binom{2k}{k}}$$

on voimassa $\zeta(4) = \frac{\pi^4}{90}$. Samoihin lopputuloksiin päästäisiin myös käyttämällä lausetta 4.2.

Olemme käsitelleet tähän mennessä tapaukset $s = 0$, $s = -2n$, $s = 1 - 2n$ ja $s = 2n$, kun $n \in \mathbb{N}$. Jokaiselle näistä on löytynyt eksplisiittinen kaava, jonka perusteella arvot on pystytty laskemaan. Parittomien positiivisten kokonaislukujen tapausta emme ole vielä tarkastelleet. Niiden kohdalla asiat mutkistuvat, sillä niiden 200 vuoden aikana, mitä ζ -funktiota on tutkittu, ei vastaavanlaisia esitysmuotoja ole löydetty. Tarkastelemme loppuosiossa arvoja $\zeta(3)$, $\zeta(5)$ ja $\zeta(7)$ ja esitämme niille myös sarjakehitelmät ja niitä vastaavat algoritmit.

Ensimmäisen sarjakehitelmän

$$(4.6) \quad \zeta(3) = \frac{5}{2} \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^3 \binom{2k}{k}}$$

esitti Hjortnaes.

Algoritmi 2. *Lasketaan $\zeta(3)$ kaavan (4.6) perusteella, kun haluttu tarkkuus D on annettu.*

```

N = 1 + ⌊ $\frac{5D}{3}$ ⌋
c = 2
s = 0
for n = 1 to N do
  s = s +  $\frac{(-1)^{n+1}}{n^3 c}$ 
  c =  $\frac{c(4n+2)}{n+1}$ 
end for
return  $\frac{5s}{2}$ 

```

Hjortnaesin esityksestä Koecher johti puolestaan $\zeta(5)$:lle kaavan

$$(4.7) \quad \zeta(5) = 2 \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^5 \binom{2k}{k}} - \frac{5}{2} \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^3 \binom{2k}{k}} \sum_{j=1}^{k-1} \frac{1}{j^2}.$$

Algoritmi 3. Lasketaan $\zeta(5)$ kaavan (4.7) perusteella, kun haluttu tarkkuus D on annettu.

$$N = 1 + \lfloor \frac{5D}{3} \rfloor$$

$$a = 0$$

$$c = 2$$

$$s = 0$$

for $n = 1$ to N **do**

$$g = \frac{1}{n^2}$$

$$s = s + \frac{(-1)^{n+1}(4n-5a)}{n^3 c}$$

$$c = \frac{c(4n+2)}{n+1}$$

$$a = a + g$$

end for

return $\frac{s}{2}$

Edelleen Borwein ja Bradley kehittivät $\zeta(7)$:lle kaavan

$$(4.8) \quad \zeta(7) = \frac{5}{2} \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^7 \binom{2k}{k}} + \frac{25}{2} \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^3 \binom{2k}{k}} \sum_{j=1}^{k-1} \frac{1}{j^4}.$$

Algoritmi 4. Lasketaan $\zeta(7)$ kaavan (4.8) perusteella, kun haluttu tarkkuus D on annettu.

```

 $N = 1 + \lfloor \frac{5D}{3} \rfloor$ 
 $a = 0$ 
 $c = 2$ 
 $s = 0$ 
for  $n = 1$  to  $N$  do
   $g = \frac{1}{n^2}$ 
   $s = s + \frac{(-1)^{n+1}(5a+g)}{n^3c}$ 
   $c = \frac{c(4n+2)}{n+1}$ 
   $a = a + g$ 
end for
return  $\frac{5s}{2}$ 

```

Kaikkien kolmen algoritmin kompleksisuus on selvästi $O(D)$ [2]. Lisäksi jokainen näistä algoritmeista on helppo toteuttaa käytännössä. Kuitenkin on olemassa laskennallisesti tehokkaampia menetelmiä laskea $\zeta(n)$:n arvoja. Näistä tarkempaa tietoa löytyy esimerkiksi Borweinin, Bradleyyn ja Crandallin artikkelista [2].

5 Yhteenveto

Riemannin ζ -funktio on monisäikeinen ongelmakenttä, jonka parissa matemaatikot ovat taistelleet jo yli 200 vuotta. Ensimmäiset kaksi suurta laskennallista tulosta olivat Riemannin yleistämä Eulerin tulokaava kompleksiluvuille ja analyyttisen jatkamisen jälkeen kehitetty Riemannin funktionaaliyhtälö, jonka avulla pystyttiin laskemaan ζ -funktion arvoja koko kompleksitasossa. Toinen merkittävä askel ζ -funktion teoriassa oli alkulukulauseen todistus, jonka avulla pystyttiin arvioimaan $\pi(x)$:ää laskennallisesti kevyemmin suurillakin x :n arvoilla.

Viime vuosikymmeninä myös tietojenkäsittelynäkökulma ja sen myötä laskennallinen tutkimus on tullut entistä tärkeämmäksi. Tärkein laskennallinen tutkimuskohde on tietenkin ollut kriittisellä suoralla olevat ζ -funktion nollakohdat, joka on keskeisin osa-alue Riemannin hypoteesin todistamisessa tai kumoamisessa osoittamisessa. Onneksi mielenkiintoa on löytynyt myös $\zeta(n)$:n tutkimiseen, kun $n \in \mathbb{Z}$.

Kuten olemme jo huomanneet, tapaukset $\zeta(2n)$, $\zeta(0)$, $\zeta(1-2n)$ ja $\zeta(-2n)$ ovat laskennallisesti helpompia kuin $\zeta(2n+1)$, sillä muissa tapauksissa on löydetty tarkka kaava tai arvo, jonka perusteella laskenta voidaan suorittaa. Sen sijaan parittomien ja ykköstä suurempien positiivisten kokonaislukujen tapauksessa samankaltaista yleispätevää kaavaa ei ole löydetty. Näin ollen tarvitaan vielä paljon tutkimusta näiden kohdalla, ja tutkimuksessa painotuu yhä enemmän tehokkaampien algoritmien kehittäminen, koska mitä suurempi kokonaisluku $2n+1$ on, sitä enemmän $\zeta(2n+1)$:n laskeminen vaatii tehoa ja aikaa tietokoneelta.

Viitteet

- [1] L. V. Ahlfors, *Complex Analysis*, 3rd ed, McGraw-Hill, 1979.
- [2] J. M. Borwein, D. M. Bradley & R.E. Crandall, Computational strategies for the Riemann zeta function, *J. Comput. Appl. Math.* **121** (2000), 247-296.
- [3] O. Hijab, *Introduction to Calculus and Classical Analysis*, 2nd ed, Springer, 2007.
- [4] H. Joutsijoki, Residylause ja sen sovelluksia, Pro gradu -tutkielma, Matematiikan, tilastotieteen ja filosofian laittoa, Tampereen yliopisto, 2007.
- [5] H. Joutsijoki, Riemannin ζ -funktio ja sen sovelluksia, Tarkastuksessa oleva lisensiaatintutkimus, Matematiikan ja tilastotieteen laitos, Tampereen yliopisto, 2009.
- [6] R. Nevanlinna & V. Paatero, *Funktio teoria*, Otava, 1963.
- [7] Prime Glossary,
<http://primes.utm.edu/glossary/xpage/Eratosthenes.html>.
 Checked 9.12.2009.
- [8] B. Palka, *An Introduction to Complex Function Theory*, Springer-Verlag, 1991.
- [9] H.A. Priestley, *Introduction to Complex Analysis*, 2nd ed, Oxford, 2003.

- [10] H. Rademacher, *Topics in Analytic Number Theory*, Springer-Verlag, 1973.
- [11] K. H. Rosen, *Elementary Number Theory and Its Applications*, 4th ed, Addison Wesley Longman, 2000.
- [12] E. C. Titchmarsh, *The Theory of the Riemann Zeta-function*, 2nd ed, Oxford, 1986.

Kasvojentunnistukseen perustuvan mobiilikäyttöliittymän arviointi

Kari Jussila

Tiivistelmä.

Tässä tutkimusraportissa esittelemme alkuun perusteluja kasvojentunnistuksen tutkimuksen jatkamiseksi, kuinka multimodaalisuus luo uusia mahdollisuuksia teknologian ja ihmisen vuorovaikutuksen parantamiseksi. Esittelemme oman mobiililaitteella tekemämme tutkimuksemme ja kerromme tutkimusmenetelmistämme, vaadituista olosuhteista ja testien suorittamisesta. Lopuksi käymme läpi testien antamia tuloksia, sekä testiin osallistujilta kerättyä palautetta. Tunnuslukujen ja palautteen pohjalta analysoimme kasvojentunnistukseen perustuvan käyttöliittymän toimivuutta nyt ja mahdollisia tulevaisuuden näkymiä.

Avainsanat ja -sanonnat: Kasvojentunnistus, mobiilikäyttöliittymä

CR-luokat: D.2.5, H.1.2

1. Johdanto

Kasvojentunnistusta konenäön avulla on tutkittu menneiden vuosien aikana [Riisgaard Hansen et al., 2006; Hannuksela et al., 2008]. Kasvojentunnistuksen toteuttaminen ja sen vaatimat algoritmit ovat olleet aikaisemmin niin raskaita, että ne ovat vaatineet pc-tasoista laskentatehoa toimiakseen. Viimeisten vuosien aikana prosessorien laskentateho on kasvanut merkittävästi ja näin on tarjoutunut mahdollisuus kasvattaa pienten mobiililaitteiden laskentatehoa. Tämän kasvaneen laskentatehon ja kameroiden integroimisen mobiililaitteisiin mahdollistamana on alettu tutkia kasvojentunnistuksen toteuttamista mobiililaitteissa, sekä niihin laadittuihin sovellusten käyttämisessä. [Riisgaard Hansen et al., 2006]

Uusissa kasvojentunnistusalgoritmeihin keskittyneissä tutkimuksissa [Hannuksela et al., 2008] on löydetty jo niin tehokkaasti toimivia algoritmeja, että reaaliaikainen kasvojentunnistus on mahdollista mobiililaitteilla. Reaaliaikainen tunnistus on välttämättömyys, jotta kasvojentunnistusta voidaan käyttää laitteen ohjaamiseen. Reaaliaikaisuuden puute kasvojentunnistuksessa tekisi laitteen käytöstä hidasta, käyttäjän ohjausliikkeet muuntuisivat viiveellisiksi ohjauskomennoiksi, ja näin ollen käyttömukavuus kärsisi. Myös virheiden

määrä kasvaisi suureksi, sillä malttamaton käyttäjä ehtisi antaa jo uusia ohjauskomentoja ennen vanhojen kommentojen suoritusta.

Nykyään monissa mobiililaitteissa on niin kutsuttuja kiihtyvyyssantureita. Nämä anturit tunnistavat laitteen liikettä suhteessa maahan, eli ne toimivat suhteessa painovoimaan. Anturit mahdollistavat monenlaisia ohjausmenetelmiä vain laitetta kallistelemalla [Partridge et al., 2002]. Tällaisissa tilanteissa laitteen käyttäjältä vaaditaan kuitenkin laitteen asennon tarkastamista. Laitteen oikea toiminta vaatii ennalta määrätyn asennon aloitusasennokseen. Kasvojentunnistus ei ole riippuvainen laitteenkäyttäjän asennosta. On perusteltua siis uskoa, että laite, joka tunnistaa käyttäjänsä kasvot, vastaa erilaisiin tarpeisiin. Ohjausta voi näin ollen käyttää myös makuuasennossa samaan tapaan kuin seistessä. Algoritmin tunnistessa kasvojen etäisyyden ja kallistelun pelkän sijainnin lisäksi, mahdollistuu vielä tarkempi ohjaus pelkästään kasvojen avulla, esimerkiksi kuvien loitonnuksen ja lähennys suhteessa kasvojen etäisyyteen.

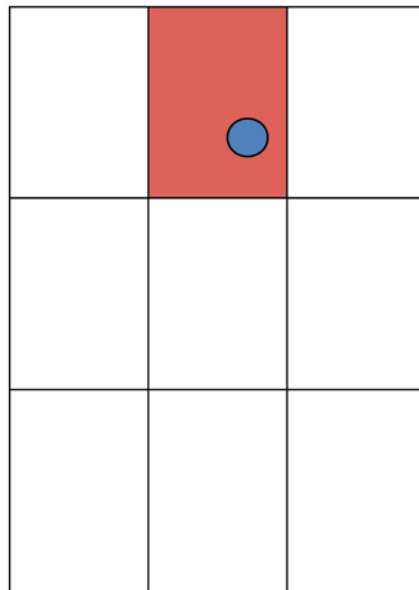
Multimodaalisuus käyttöliittymissä tarkoittaa käyttöliittymän kykyä ottaa vastaan ja antaa palautteena useampia syötemodaliteetteja eli erilaisia ohjaustapoja [Multimodal Interaction, 2009]. Useampien kuin yhden syötemodaliteetin tunnistaminen ei ole välttämätöntä useimmille käyttäjille. Se kuitenkin lisää niin sanottua vuorovaikutuskaistan leveyttä laitteen ja käyttäjän välillä. Tämä laajennettu vuorovaikutusmahdollisuus tuo mahdollisuuden erilaisten käyttöliittymien käyttämiseen myös niille, joille normaalien syötemodaliteettien käyttö on hankalaa tai mahdotonta.

Edellisiä tutkimustuloksia sekä multimodaalisuuden tarjoamia mahdollisuuksia ajatellen, päätimme yhdessä ohjelmiston rakentajan ja ohjaajani Toni Vanhalan kanssa lähteä tutkimaan, mitä kasvojentunnistuksella tällä hetkellä saadaan aikaiseksi. Ongelmaksi valitsimme tilanteen, jossa käyttäjä tarvitsisi suurta työpöytää, mutta käytössä on vain mobiililaitteen pieni näyttö. Kokeelliseksi esimerkiksemme valitsimme kartan navigoinnin mobiililaitteella, jossa suurta karttaa pitäisi pystyä selaamaan pienellä mobiililaitteen näytöllä. Tätä kartan selaamista varten ohjelmaan liitettiin kasvojentunnistusalgoritmi, jonka avulla kartan navigointi onnistuu. Vertailukelpoisia tuloksia saadaksemme rakensimme samaan navigointiohjelmaan myös kosketuskäyttöliittymän, näin saimme suhteutettua kasvojentunnistuksesta saatuja tuloksia jo käytössä olevaan ohjausmenetelmään.

2. Tutkimusmenetelmät

2.1. Testiohjelmiston kuvaus

Testiohjelmistoksi päätettiin tehdä kartan navigointiin käytettävä ohjelma. Kartaksi valittiin Prahan alueen kartta, jossa ei näkyisi tuttuja paikkoja, jolloin testihenkilön mielenkiinto voisi herpaantua. Näytöllä näkyy kerralla pieni osa suuresta kartasta ja näyttö on jaettu yhdeksään yhtä suureen suorakaiteen muotoiseen alueeseen. (Kuva 1). Alueiden väliset rajat eivät kuitenkaan ole näkyvissä, jotta kartta olisi helposti luettavissa. Näytöllä näkyy koko ajan kursori, joka liikkuu suhteessa käyttäjän kasvoihin. Kartan navigointi tapahtuu siirtämällä kursori jonkin muun kuin keskimmäisen alueen päälle, joihin näyttö on jaettu. Kursorin oltua 300 millisekuntia alueen päällä alue aktivoituu ja muuttuu punaiseksi. Aktivoitumisesta 200 millisekunnin jälkeen kartta alkaa liikkua valitun alueen suuntaisesti. Tämä yhteensä 500 millisekunnin viive on välttämätön, jotta kasvojentunnistuksessa mahdollisesti tapahtuvien virheiden aiheuttama kursorin heittelehtiminen ei tekisi navigoinnista mahdotonta. Kartan selaamisen pysäyttäminen tapahtuu keskittämällä kursori ruudun keskellä sijaitsevan alueen päälle. Keskimmäinen alue ei aktivoitu kursorista ja kartta pysyy paikoillaan.



Kuva 1. Näytön jako navigointialueisiin.

Valittu kasvojentunnustus algoritmi perustuu kasvojen värialueen tunnistamiseen HSV-väriavaruudessa, jossa värit tunnistetaan niiden sävyn (hue), kylläisyyden (saturation) ja kirkkouden (value) perusteella. Ihonväriksi katsomme värialue on $0 < H < 20$, $0.34 < S < 0.41$, kun V voi saada mitä tahansa

arvoja. H:n arvot ovat välillä 0-360 ja S:n arvot välillä 0-1. Käytetyt raja-arvot on haettu manuaalisesti, keräämällä erilaisista kuvista kasvojen ja taustan pikseleitä. Pikseleitä kerätessä valittiin sekä huonosti, että hyvin valaistuja kasvojen ja taustan pikseleitä. Lopulta kokeilemalla saavutettiin värialue, jota voidaan pitää kasvojen värisenä.

Vertailukelpoisten tulosten saavuttamiseksi koskettamalla tapahtuva ohjaus luotiin samalla tavalla toimivaksi, kuin kasvojentunnistuksella tapahtuva ohjaus. Aktivoinnissa ja kartan liikkeelle lähdössä olleita viiveitä ei olisi tarvittu kosketusohjauksessa, mutta niiden puute olisi tehnyt tuloksista vertailukelvottomia ajan suhteen. Erona tapojen välille jäi vain kursorin puute kosketusohjauksessa; siinä näytön pinnan koskettaminen halutulta alueelta toimi halutun alueen aktivoivana toimenpiteenä. Kartan pysäyttäminen tapahtuu täten nostamalla ohjauskynä näytön pinnalta pois.

2.2. Testien sisältö

Sovelluksella suoritettut testit olivat jokaiselle testiin osallistujalle samanlaiset. Ainoa eroavaisuus tuli siitä, että joka toinen osallistuja aloitti kasvojentunnistuksella ja joka toinen kosketukseen perustuvalla ohjauksella. Tämä valinta haluttiin tehdä siksi, että koehenkilöiden mahdollinen alkuinnostus, tai loppuun sijoittuva kyllästyneisyys ei vaikuttaisi kummastakaan ohjaustavasta kerättäviin tuloksiin. Jokaiselle osallistujalle oli varattuna samat neljä reittiä kartalta, joissa jokaisessa oli 15 kohdetta haettavana. Haettavat kohteet merkittiin punaisella pallolla karttaan. Kohteen valinta tapahtui ohjauskynällä koskettamalla, minkä jälkeen opastus siirtyi osoittamaan seuraavaan kohteeseen. Reittejä voidaan pitää vaikeusasteeltaan tasavertaisina: jokaisessa oli siirtymiä kolmea eri pituutta, ja jokaisen pituisia siirtymiä oli viisi kappaletta.

Testisuorituksista keräsimme talteen aika- ja virhetilastoja. Mittasimme jokaisesta reitistä keskimääräisen ajan kahden kohteen väliselle siirtymälle. Valitulla mittaustavalla saimme parhaiten selville kahden erilaisen ohjaustavan nopeuseron. Lisäksi tämä mitattu suure kertoo hyvin mahdollisen parannuksen samalla ohjaustavalla tehtyjen suorituskertojen välillä. Toinen mittauskohde oli keskimääräisten virhevalintojen määrä yhtä kohdetta kohti. Virhevalinnaksi laskettiin sellaiset kohteen valintayritykset, joissa käyttäjä pyrki valitsemaan kohteen kartalta, mutta ei osunut kohteeseen.

Testisuoritusten jälkeen käyttäjät täyttivät palautelomakkeen (Liite 1). Lomakkeessa käyttäjiä pyydettiin arvioimaan käyttöliittymän helppokäyttöisyyttä sekä heidän omia tuntemuksiaan tehdyistä testeistä. Lomakkeen kysymyksistä osa oli asteikollisia valintakysymyksiä ja osa avoimia kysymyksiä. Palautteen kerääminen käyttäjiltä oli tarpeellista, koska uuden teknologian käyttömahdol-

lisuuksien arvioinnissa on tärkeää tietää käyttäjien mielipide sen käytettävyydestä. Teknologian käyttöönottoa ei voi perustella pelkästään sen uutuudella vaan on tärkeää, että se tuo uusia käyttömahdollisuuksia vanhojen rinnalle.

2.3. Testiolosuhteet

Testauksessa käytetyn tekniikan ja sovelluksen olosuhdevaatimukset tuli ottaa huomioon. Kasvojentunnistuksen perustuessa ihonväriin oli selvää, että tarvittaisiin ympäristö, jossa konenäkö voisi tunnistaa kasvot värin perusteella. Omien testiemme tuloksena huomasimme, että valaistus oli todella tärkeä tekijä kasvojentunnistuksen optimaalisen toiminnan kannalta. Lisäksi häirtatekijöinä olivat liian suuret kaula-aukot ja ihonväriset objektit, jotka häiritsivät kasvojentunnistusalgoritmin toimintaa. Koska jokaiselle testiin osallistujalle tuli taata yhdenmukaiset olosuhteet, joissa testien suoritus sujuisi ilman ylimääräisiä ongelmia, oli olosuhteet hoidettava kuntoon ennen jokaista testikertaa. Parhaaksi ympäristöksi osoittautui sellainen, jossa oli valkoinen seinä taustalla, sekä kohdevalo osoittamassa testihenkilön kasvoihin. Näin kasvojentunnistus saatiin toimimaan tarpeeksi hyvin, että sovelluksen käytettävyys oli hyväksyttävällä tasolla. Tällaisten olosuhteiden luominen onnistui myös useampaan eri paikkaan, jolloin testaaminen helpottui, kun jokaisen osallistujan ei tarvinnut saapua tiettyyn paikkaan.

2.4. Osallistujat

Alkuun mietimme tutkimukseen osallistuvien testihenkilöiden valintaa, mistä kohderyhmästä ja erityisesti minkä ikäisiä henkilöitä valittaisiin. Valintaa tehdessä mietimme, keiden olisi helpointa käyttää uudenlaista käyttöliittymää mobiililaitteessa. Halusimme testihenkilöiden suoriutuvan sovelluksen käytöstä tarpeeksi pienin ponnistuksin, jotta testituloksiin ei vaikuttaisi liiaksi testihenkilöiden kokemattomuus teknologian kanssa. Koehenkilöillä ei ollut mahdollisuutta tutustua perusteellisesti testauksessa käytettyyn sovellukseen ja kasvojentunnistuksella ohjaamiseen, vaikka pieni tutustumisaika ennen varsinaisten testien suorittamista oli tarjolla. Testiin valittavien henkilöiden toivottiin siis olevan teknologiaan ennakkoluulottomasti suhtautuvia ja kykeneviä suoriutumaan annetuista tehtävistä. Edellä mainituista syistä oli luontevaa valita testiin osallistujat nuorista aikuisista. Koehenkilöiden keski-ikä oli lopulta hiukan yli 20 vuotta. Testauksen arvioitiin olevan tarpeeksi suuntaa-antava kymmenellä osallistujalla, näistä kymmenestä osallistujasta neljä oli naisia. Naisten edustuksen tarpeeksi suuri osuus osallistujista mahdollisti sen, että sukupuolten välisiä eroavaisuuksia voitiin saada näkyviin.

3. Tutkimuksen tulokset

Tutkimusta käynnistäessämme emme varsinaisesti odottaneet mitään tuloksia. Mitään hypoteeseja ei asetettu, joihin olisimme halunneet etsiä vahvistusta tai kumoavia tuloksia. Seuraavassa esitellään tutkimuksessa kerättyjä tuloksia ja niistä laskettuja erilaisia tunnuslukuja. Myöhemmässä alakappaleessa käymme läpi käyttäjien antamaa palautetta avointen kysymysten vastauksista sekä muuten useasti esille nousseita asioita.

3.1. Tutkimuksen tunnuslukuja ja tilastoja

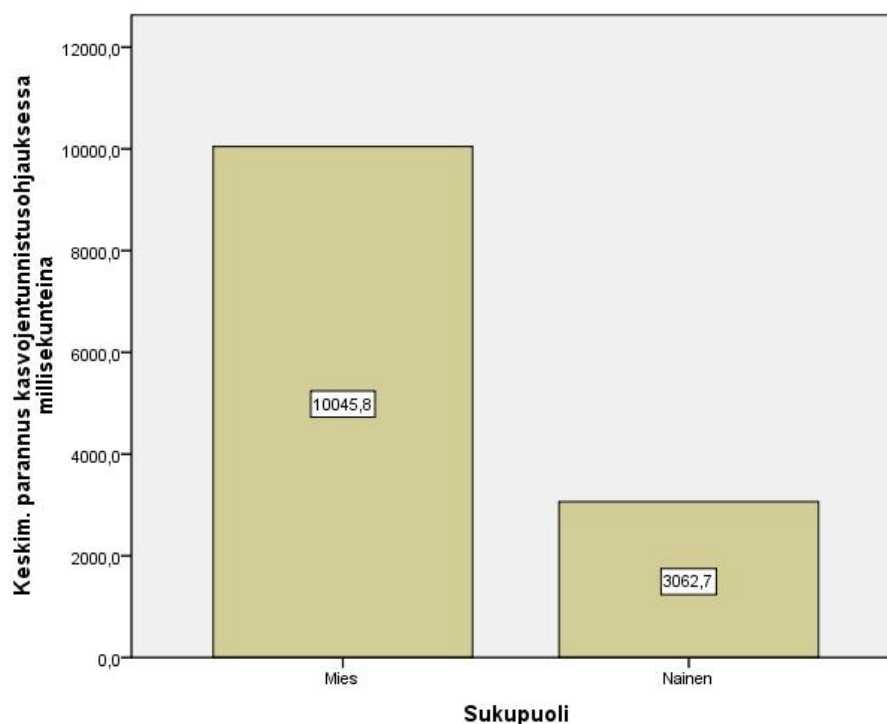
Koko tutkimuksen odotetuimmat ja oleellisimmat tunnusluvut olivat mahdolliset nopeuserot kahden ohjaustavan välillä. Mittaamamme keskimääräinen yhteen siirtymään kohteiden välillä kulunut aika osoittautui poikkeavaksi ohjaustapojen välillä. Jokaiselle testihenkilölle laskettiin keskimääräinen aika, jonka henkilö kulutti yhteen siirtymään kummallakin ohjaustavalla. Näistä henkilöiden keskimääräisistä ajoista laskimme kaikkien yhteiset keskimääräiset siirtymäajat. Kosketusohjauksen osalta keskimääräiseksi siirtymäajaksi kohteiden välillä muodostui 6419 millisekuntia, kasvojentunnustuksen keskimääräisen siirtymäajan ollessa 18328 millisekuntia. Nopeuseroa ohjaustapojen välillä voidaan pitää merkittävänä kasvojentunnustukseen perustuvan ohjauksen viedessä lähes kolminkertaisen ajan verrattuna kosketusohjaukseen.

Yllätyksenä edellä mainittua nopeuseroa ei voida pitää, sillä kasvojentunnustukseen perustuva ohjaustapa on vielä tutkimusasteella, eikä se ollut entuudestaan tuttu koehenkilöille. Ohjaustapojen eroja ja mahdollisuuksia tarkemmin kuvaavia tunnuslukuja voidaankin löytää verrattaessa testiin osallistuneiden henkilöiden kehittymistä kahden suorituskerran välillä eri ohjaustavoilla. Laskettuamme jokaiselle osallistujalle aikaeron kahden suoritettun reitin välillä molemmille ohjaustavoille erikseen, saimme mielenkiintoisia tuloksia. Kosketusohjaukseen testihenkilöt olivat keskimäärin hyvin tottuneita, eikä merkittävää kehitystä kahden suorituskerran välillä ilmennyt. Kosketusohjaukseen perustuvien suoritusten keskimääräinen parannus siirtymää kohden oli 1341 millisekuntia, kun kasvojentunnustusta käytettäessä parannusta kahden kerran välillä syntyi 7253 millisekuntia. Uuden ohjaustavan opettelun alkuvaiheessa nopea kehitys onkin odotettavaa, mutta näin suuri kehitysaskel yllätti meidät.

Koehenkilöiden tekemät virheet, joita mittasimme jokaista suorituskertaa kohti, eivät lopulta tuoneet paljoa lisäinformaatiota. Virheiden määrässä ei ollut eroja ohjaustapojen välillä, eikä myöskään suorituskertojen välillä. Eroavaisuuksien puute johtuu osallistujien hyvästä suoriutumisesta; virheitä tehtiin viidentoista siirtymän reiteillä keskimäärin yksi. Kohteiden karkailua ei siis

ilmennyt edes ensimmäisellä kasvojentunnistukseen perustuvalla ohjauksella, ainakaan se ei aiheuttanut virhepainalluksia.

Halusimme testejä järjestäessämme nähdä, olisiko sukupuolella vaikutusta testien tuloksiin. Lähestulkoon kaikki keskimääräiset aika-arvot, joita laskimme, olivat yhteneviä miesten ja naisten välillä. Virheissäkään ei eroja sukupuolten välille syntynyt, koska virheitä tehtiin niin vähän. Yksi selkeä eroavaisuus sukupuolten välille testeissä kuitenkin muodostui. Parannus kahden kasvojentunnistuksella ohjatun reitin välillä oli miehillä selkeästi naisia suurempi. Naiset paransivat tulostaan keskimäärin 3063 millisekuntia siirtymää kohden, kun miehet paransivat tulostaan toisella yrityskerralla jopa 10046 millisekuntia (kuva 2). Syitä tähän eroon emme voi tutkimuksen puitteissa päätellä, voimme vain arvella sen johtuvan miesten luontaisesti suuremmasta mielenkiinnosta ja halusta oppia uutta teknologiaa.



Kuva 2. Miesten ja naisten keskimääräinen parannus suoritusajassa kasvojentunnistuksella ohjattaessa.

3.2. Käyttäjiltä kerättyä palautetta

Käyttäjien antamasta palautteesta selkeimmin esille nousi kosketuskäyttöliittymän helppous verrattuna kasvojentunnistuksella ohjaamiseen. Keräämämme tiedot suoritusajoista tukevat palautteen ymmärtämistä. Ei ole ihme, jos kolminkertaisen ajan vaatinut suoritus on käyttäjien mielestä ollut vaikeampi. Palautteen yksipuolisuuteen vaikuttaa selkeästi myös käyttäjien aikaisempi ko-

kemus kosketusohjauksesta, siihen verrattuna uusi ohjaustapa tuntui vaikealta ja hankalalta jopa niiden mielestä, joille kasvojentunnistuksella ohjaaminen sujui ongelmitta.

Miellyttävyyttä kysyttäessä osallistujat olivat hyvin yksimielisiä siitä, että kosketusohjaaminen tuntui miellyttävältä ja kasvojentunnistus säännönmukaisesti hieman epämiellyttävältä. Epämiellyttävyyden perusteiksi palautteesta nousi kasvojentunnistusohjauksen epätarkempi ohjautuvuus ja ohjauksen karkailu kasvojen kadotessa kameran alueelta.

Jokainen kokeeseen osallistunut henkilö arvioi myös omaa kiihtyneisyytään ohjaustilanteiden aikana. Asteikko oli yhdeksänportainen ja vain muutamalla osallistujalla ero oli kahden portaan sisällä, jokaisen silti arvioidessa kiihtyneisyytensä korkeammaksi kasvojentunnistukseen perustuvan ohjauksen aikana. Kiihtyneisyys korreloi käyttäjillä selkeästi keskittymisen kanssa, mutta myös ohjauksen karkailu ja epäonnistumiset nostivat kiihtyneisyyttä. Kosketusohjauksen aikana käyttäjät saattoivat kysellä testin etenemisestä, tai puhua muita asioita, koska ohjaaminen oli rutiininomaista. Käyttäjien keskittyneisyys näkyi selkeästi hiljaisuutena ja laitteen tiukkana tuijotuksena kasvojentunnistusta käytettäessä. Kasvojentunnistuksen vaatima tiukka keskittyminen voisi osaltaan selittää aikojen selvästi suurempaa parannusta tällä ohjaustavalla, mutta asia jää tutkimuksemme puitteissa vain arvioksi. Yksi käyttäjä arvioi kosketuskäytön jopa tylsäksi tavaksi, joskin silti käytännössä toimivammaksi.

Kasvojentunnistusta käytettäessä käyttäjän oli mahdollista ohjailla kasvonsa asemaa suhteessa laitteen kameraa joko kallistelemalla laitetta kädessään, tai siirtelemällä sitä kasvojen edessä suurin kädenliikkein. Jokainen osallistujista käytti tekniikkana laitteen kallistelua, jolloin kasvojen suhde kameraan muuttui helpommin. Varsinaista vertailua emme yksipuolisen ohjaustapavalinnan vuoksi saaneet tapojen välillä aikaan, mutta jo toisen tavan nopea hylkääminen käyttäjien toimesta kertoo kallistelun luonnollisesta sopivuudesta ohjaustavaksi.

Käyttäjät antoivat myös palautetta siitä, miten kasvojentunnistusta käytettäessä navigoinnin vaikeusaste vaihteli eri suuntien välillä. Erityisesti käyttäjät valittelivat kartan alaspäin liikuttelun vaikeutta. Ongelma oli todellinen monille osallistujille ja sen havaittiin johtuvan mobiililaitteen kameran sijoittelusta. Kameran sijainti näytön oikeassa yläkulmassa aiheuttaa sen, että kun laitetta kallistaa alaspäin selataksaan karttaa, tulee katselukulma epämiellyttävän pieneksi. Ylös sijoitettu kamera näkee laitetta alas kallistettaessa kasvot pitkään keskemällä, kuin mitä kasvot suhteessa näyttöön ovat. Tästä johtuen taas ylöspäin navigointi oli todella helppoa ja joillakin testiin osallistujilla kartta pyrki karkailemaan tahattomasti ylöspäin. Ongelmaa on vaikea korjata kame-

ran sijoittelulla, sillä keskelle näyttöä kameraa ei ole kovin helppoa tai edes mielekästä asentaa. Kuitenkin konenäön voisi suhteuttaa siten, että kasvojen ollessa keskellä suhteessa näyttöön, tulkitsisi myös kasvojentunnistus kasvot keskelle näkökenttäänsä. Tällä suhteuttamisella saataisiin keskitys kohdalleen, mutta kasvojentunnistamisen alue pienenesi huomattavasti. Silloin ohjaaminen tapahtuisi pienemmillä liikkeillä ja virhealttius kasvaisi nykyisestä. Oikea kompromissi pitäisi etsiä kokeilemalla erilaisia toteutuksia.

4. Tutkimuksen analysointia ja kasvojentunnistuksen tulevaisuuden näkymiä

Tutkimuksen antamat tulokset kertovat selkeästi, että vielä kasvojentunnistukseen perustuvien käyttöliittymien kaupallistuminen on vielä edessäpäin. Kosketukseen perustuva ohjaus oli sekä mitattavien suureiden, että käyttäjien antaman palautteen mukaan selkeästi toimivampaa ja nopeampaa. Arvokkaana tuloksena voidaan kuitenkin huomata tutkimukseen osallistuneiden henkilöiden kehittyminen kasvojentunnistusohjauksessa jo testiin osallistumisen aikana. Vaikka kasvojentunnistuksesta ei koskaan saataisi absoluuttisilla mittareilla kosketuskäytön voittajaa, voidaan sitä hyödyntää lisättäessä multimodaalisuutta tilanteisiin, joihin kosketuskäyttö ei sovellu.

Suurimpana ongelmana kasvojen väriin perustuvassa tunnistuksessa voidaan pitää olosuhteiden tarkkoja vaatimuksia. Taustan ja vaatteiden tulee poiketa ihonväristä, tai algoritmi ei pysty tunnistamaan kasvoja oikein. Tämä rajoittaa nykyisellään algoritmin käytön vain määrättyihin olosuhteisiin. Kyseistä algoritmia ei kuitenkaan kannata hylätä kokonaan; sitä kehittelemällä voitaisiin saada aikaan algoritmi, joka tunnistaa kasvot yhä väristä, mutta voisi käyttää apunaan muitakin menetelmiä. Kasvojen muodon ja värin tunnistamisen avulla voitaisiin saada aikaan jo hieman luonnollisemmassakin ympäristössä toimiva algoritmi. Tällaisen algoritmin luomisessa tarvitsisi löytää sopiva ratkaisu tarkkuuden ja laskentanopeuden välillä, jotta tunnistus olisi yhä mahdollista tehdä reaaliajassa mobiililaitteella. Hattujen ja hiustyylien vaikutus algoritmin toimintaan olisi myös tutkittava, joten alueen tutkimista on syytä jatkaa tulevaisuudessakin.

Testauksessa käyttämäämme ongelmaan suuren näyttöalueen selaamisesta kasvojentunnistus ei välttämättä levottomuutensa vuoksi tuonut ratkaisua. Kukaan tuskin haluaa selata esimerkiksi työpöytänsä siten, että kun keskittyminen kasvojen sijaintiin hetkeksikin herpaantuu muuta tehdessä, suuntaa näyttö eri suuntaan juuri etsitystä sijainnista. Ainakin kasvojentunnistus pitäisi olla kytkettävissä nopeasti pois päältä ja takaisin, tarpeen mukaisesti. Tärkeämpää kuitenkin oli havaita tekniikan potentiaali: tavalliseen mobiililaitteeseen

saatiin yksinkertaisella algoritmilla kasvot tunnistava konenäkö, jota voitiin käyttää laitteen ohjauksessa.

Tekemämme tutkimuksen laajuudesta johtuen siitä ei voida tehdä luotettavia yleistyksiä kaikkien teknologian käyttäjien keskuudessa. Tulevaisuuden teknologian käyttäjinä testiin osallistuneet henkilöt kuitenkin antoivat arvokasta tietoa meille kasvojentunnistukseen suhtautumisestaan ja kyvystä oppia sen käyttöä. Teknologiaa käyttämään tottunut sukupolvi omaksuu tämänkin tekniikan, kunhan se saadaan toiminnaltaan sille tasolle, että sitä voidaan soveltaa osana laitteiden ohjausta.

Viiteluettelo

- [Bradski, 1998] Gary R. Bradski, Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal Q2 '98*.
- [Hannuksela et al., 2008] Jaru Hannuksela, Pekka Sangi, Markus Turtinen and Janne Heikkilä, Face tracking for spatially aware mobile user interfaces. In: *3rd International Conference, ICISP 2008, Lecture Notes in Computer Science 5099*, 405-412.
- [Multimodal interaction, 2009]
http://en.wikipedia.org/wiki/Multimodal_interaction, Checked 14.12.2009.
- [Riisgaard Hansen et al., 2006] Thomas Riisgaard Hansen, Eva Eriksson and Andreas Lykke-Olesen, Use your head: exploring face tracking for mobile interaction. In: *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, Montréal, 2006, 845 - 850.
- [Partridge et al., 2002] Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello and Roy Want, TiltType: accelerometer-supported text entry for very small devices. In: *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology*, Paris, 2002, 201-204.

(Liite 1)

Kasvojentunnistukseen perustuvan mobiiliäyttöliittymän arviointilomake

Ikä : _____ vuotta

Sukupuoli: Mies Nainen

Kätisyys: Oikeakätinen Vasenkätinen

1. Kuinka miellyttävältä kasvojentunnistukseen perustuva ohjaaminen tuntui?

				Ei epämiellyttävältä					
				eikä miellyttävältä				Miellyttävältä	
Epämiellyttävältä									
-4	-3	-2	-1	0	+1	+2	+3	+4	

2. Kuinka miellyttävältä kosketusnäyttöön perustuva ohjaaminen tuntui?

				Ei epämiellyttävältä					
				eikä miellyttävältä				Miellyttävältä	
Epämiellyttävältä									
-4	-3	-2	-1	0	+1	+2	+3	+4	

3. Kuinka kiihtyneeksi tunsit itsesi kasvojentunnistuksella ohjaamisen aikana?

				En rauhalliseksi					
				enkä kiihtyneeksi				Kiihtyneeksi	
Rauhalliseksi									
-4	-3	-2	-1	0	+1	+2	+3	+4	

4. Kuinka kiihtyneeksi tunsit itsesi kosketusnäytöllä ohjaamisen aikana?

				En rauhalliseksi					
				enkä kiihtyneeksi				Kiihtyneeksi	
Rauhalliseksi									
-4	-3	-2	-1	0	+1	+2	+3	+4	

5. Kuinka toimivaa kasvojentunnistuksella ohjaaminen oli, saiko tehtävät tehtyä?

				Ei kumpaakaan					
								Toimivaa	
Toimimatonta									
-4	-3	-2	-1	0	+1	+2	+3	+4	

6. Kuinka toimivaa kosketusnäytöllä ohjaaminen oli, saiko tehtävät tehtyä?

				Ei kumpaakaan					
								Toimivaa	
Toimimatonta									
-4	-3	-2	-1	0	+1	+2	+3	+4	

7. Kuinka tehokasta kasvojentunnistuksella ohjaaminen oli, suoriuditko tehtävistä nopeasti?

Tehotonta	Ei kumpaakaan						Tehokasta	
-4	-3	-2	-1	0	+1	+2	+3	+4

8. Kuinka tehokasta kosketusnäytöllä ohjaaminen oli, suoriuduitko tehtävistä nopeasti?

Tehotonta	Ei kumpaakaan						Tehokasta	
-4	-3	-2	-1	0	+1	+2	+3	+4

9. Kuinka tyytyväinen yleisesti ottaen olit kasvojentunnistuksella ohjaamiseen?

Tyytymätön	En kumpaakaan						Tyytyväinen	
-4	-3	-2	-1	0	+1	+2	+3	+4

10. Kuinka tyytyväinen yleisesti ottaen olit kosketusnäytöllä ohjaamiseen?

Tyytymätön	En kumpaakaan						Tyytyväinen	
-4	-3	-2	-1	0	+1	+2	+3	+4

11. Kumpi ohjaustavoista tuntui sinusta luontevammalta? Miksi?

12. Kumpaa ohjaustapaa käyttäisit mieluummin kartan selaamisessa mobiililaitteella? Miksi?

13. Mikä kokeessa oli vaikeinta ja/tai epämiellyttävintä? Miksi?

14. Mikä kokeessa oli helpointa ja/tai miellyttävintä? Miksi?

15. Kumpaa ohjaustapaa käytit kasvojentunnistuksen yhteydessä enemmän, laitteen kallistelua, vai puhelimen liikuttelua isoilla liikkeillä kasvojesi edessä? Miksi?

Ohjaaja: Toni Vanhala

Sosiaalinen media opetuskäytössä

Maiju Karhunen

Tiivistelmä.

Tässä tutkielmassa käsitellään sosiaalisen median roolia opetuskäytössä. Tutkielmassa tuodaan esille, mitä sosiaalisella mediallyä tarkoitetaan, miten sosiaalista mediaa voidaan käyttää opetuksessa, mitä opettajalta vaaditaan sen käytön suhteen sekä mitä hyötyä ja haittaa sosiaalisen median käytöstä on opetuksen kannalta. Tutkielmassa esitellään myös tutkivan oppimisen teoriaa, jota peilataan tutkielman aikana sekä opettajan rooliin että sosiaalisen median sovelusten käyttöön.

Avainsanat ja -sanonnat: blogit, keskustelupalstat, kuvajakelupalvelut, oppiminen, opetus, opettaja, sosiaalinen media, tutkiva oppiminen, verkkoviestintätyökalut, verkkoyhteisöpalvelut, videoverkkopalvelut, vuorovaikutteisuus, wikit.

CR-luokat: C.2.6, K.3.1, K.3.2.

1. Johdanto

Sosiaalinen media voidaan määritellä lukemattomilla eri tavoilla. Eräs sosiaalisen median määritelmä voisi kuulua näin: sosiaalinen media on Internetin avustuksella toimiva vuorovaikutteinen sovellus, jonka sisältöä käyttäjät voivat muokata ja jakaa toisten käyttäjien kesken.

Sosiaalinen media pitää karkeasti jaotellen sisällään blogit, keskustelupalstat, kuvajakelupalvelut, verkko-oppimisympäristöt, verkkoviestintätyökalut, verkkoyhteisöpalvelut, videoverkkopalvelut sekä wikit.

Tutkielmani pääpaino on sosiaalisessa mediassa ja opettajissa, joten pyrin tutkielmassani keskittymään erityisesti opettajiin ja heidän suhtautumiseen sosiaaliseen mediaan. Tarkastelen myös sosiaalisen median käyttöä opetustilanteissa sekä oppilaan roolia sosiaalisen median opetuskäytössä. Apuna tarkastelussa käytän olemassa olevia tutkimuksia ja muita tutkielmani aihetta sivuavia kirjallisia lähteitä.

Tutkielman luvussa 2 perehdytään siihen, mitä sosiaalisella mediallyä itse asiassa tarkoitetaan ja mitä kaikkea siihen voidaan laskea kuuluvaksi. Kun sosiaalisen median sisältö on selvillä, siirrytään lukuun 3, jossa tarkastellaan tutkivan oppimisen teoriaa. Luvussa 4 mietitään sosiaalisen median mahdollisuuksia opetuskäytössä ja pohditaan tutkivan oppimisen yhdistämistä sosiaali-

sen median sovelluksiin. Luvussa 5 puolestaan pohditaan, mitä opettajalta vaaditaan sosiaalisen median käytön suhteen opetustilanteessa sekä pohditaan opettajan roolia tutkivan oppimisen kannalta. Luku 6 summaa sosiaalisen median opetuskäytön myönteisiä ja kielteisiä puolia ja toimii samalla yhteenvetona koko tutkielmalle.

2. Sosiaalinen media

Tämän luvun tehtävänä on luoda lukijalle selvä käsitys siitä, mitä sosiaalinen media itse asiassa tarkoittaa sekä minkälaisia sovelluksia sosiaaliseen mediaan kuuluu.

Tässä tutkielmassa sosiaalinen media pitää sisällään Internetissä toimivat vuorovaikutteiset sovellukset. Vuorovaikutteisuuksella puolestaan tarkoitetaan tässä yhteydessä sitä, että Internetin käyttäjät voivat osallistua sosiaalisen median sovelluksen käyttöön yhdessä toisten käyttäjien kanssa esimerkiksi tekstiä muokkaamalla tai kommentoimalla jo julkaistua tekstiä.

Koska sosiaalinen media on suhteellisen laajasti ymmärrettävissä oleva käsite, on sitä hyvä jakaa pienempiin osiin. Tämä onnistuu käymällä läpi sovelluksia, joiden katsotaan mahtuvan sosiaalisen median piiriin.

Tällaisia sovelluksia ovat muun muassa keskustelupalstat, blogit, wikit, verkko-oppimisympäristöt, kuvajakelupalvelut, verkkoyhteisöpalvelut, videoverkkopalvelut sekä verkkoviestintätyökalut.

Internetin keskustelupalstoilla ihmiset ovat vuorovaikutuksessa toistensa kanssa useimmiten kirjoittamalla kysymyksiä ja vastauksia tiettyihin aihepiireihin, kuten terveyteen tai musiikkiin, liittyen. Kanssakäymiseen voi liittyä myös keskustelupalstalle lähetettyjä kuvia tai jopa videopätkiä.

Blogit ovat Internetistä löytyviä päiväkirjaa muistuttavia sivustoja. Blogin kautta voi jakaa esimerkiksi ruokaohjeita, mutta blogi soveltuu opiskeluunkin.

Wikit ovat Internetistä löytyviä muokattavissa olevia tietosanakirjoja. Tärkeimpänä esimerkkinä wikeistä on Wikipedia, joka on kaikkien käyttäjien muokattavissa oleva tietosanakirja. Muokkaus-oikeus wikiin voi olla myös ainoastaan esimerkiksi tietyllä kouluryhmällä.

Verkko-oppimisympäristössä opiskelu pyrkii jäljittelemään luokkahuoneessa tapahtuvaa opiskelua mahdollisimman hyvin. Tarjolla on riittävästi materiaalia, toisten kanssa voi keskustella ja opettajalta voi kysyä neuvoa. Verkko-oppimisympäristöstä esimerkkinä mainittakoon Moodle.

Kuvajakelupalveluiden avulla Internetin käyttäjät voivat lähettää ottamiaan kuvia toisten käyttäjien katseltaviksi. Kuvajakelupalveluihin yhdistyy monesti muunlaistakin vuorovaikutteellisuutta. Käyttäjien voi olla mahdollista esimerkiksi kommentoida toisten ottamia kuvia.

Tällä hetkellä ehkä merkittävimpinä sosiaalisen median sovelluksina voidaan pitää verkkoyhteisöpalveluita, joiden avulla ihmiset ovat yhteydessä toisiinsa päivittäin. Tunnetuin esimerkki verkkoyhteisöpalvelusta lienee suuren suosion saavuttanut Facebook, jossa jokaisella käyttäjällä on oma profiili, jota käyttämällä käyttäjä on yhteydessä muihin käyttäjiin.

Videoverkkopalvelujen avulla käyttäjä pääsee katsomaan ja jakamaan videoita toisten käyttäjien kesken. Tärkeimpänä esimerkkinä videoverkkopalveluita on YouTube, jossa käyttäjän on mahdollista katsoa jopa kokonaisia elokuvia pätkissä.

Verkkoviestintätyökaluista esimerkkinä ovat erilaiset Internet-yhteyden vaativat keskustelusovellukset kuten Messenger. Messengerin avulla käyttäjä voi keskustella reaaliaikaisesti toisella puolella maapalloa asuvan ihmisen kanssa joko kirjoittamalla, puhumalla tai videokeskustelun välityksellä.

Kuten esitellyistä sosiaalisen median sovellusesimerkeistä käy ilmi, sosiaalinen media pitää sisällään paljon erilaisia sovelluksia. Kaikkia sosiaalisen median sovelluksia yhdistää kuitenkin yksi tekijä: kaikki sovellukset korostavat käyttäjiensä välistä vuorovaikutusta.

Tässä luvussa pohdittiin sosiaalista mediaa hieman yksityiskohtaisemmin ennen siirtymistä seuraavien lukujen laajempiin kokonaisuuksiin. Seuraavassa luvussa käsitellään yhtä oppimisen muotoa, tutkivaa oppimista.

3. Tutkiva oppiminen

Tässä luvussa käsitellään, mitä tutkivalla oppimisella tarkoitetaan. Tämän lisäksi pohditaan tutkivan oppimisen myönteisiä ja kielteisiä puolia. Luvun aikana esitellään myös eräs tutkivaan oppimiseen liittyvä tutkimus ja sen tulokset. Tutkiva oppiminen perustuu John Deweyn 1900-luvun alussa kehittämiin ajatuksiin, joiden mukaan opettamisen pitäisi perustua oppijan tiedonnälkään eli siihen, että oppija haluaa hakea itse uutta tietoa vanhentuneen tilalle. Oppiminen koetaankin tiedon tulkitsemiseksi ja omien tulkintojen kokeilemiseksi käytännössä. (Rauste-von Wright & von Wright, 1998)

Tutkivan oppimisen positiivisia puolia ovat muun muassa uuden tiedon omaksuminen aiemmin opittua tietoa pohjana käyttäen, sosiaalisten vuorovaikutustilanteiden lisääntyminen sekä se, että sama asia on mahdollista tulkita usealla eri tavalla. Tutkiva oppiminen mahdollistaa sen, että oppimista ei tarvitse arvioida ainoastaan yhdellä tavalla, vaan sen arviointiin on mahdollista käyttää useita kriteereitä. Tutkiva oppiminen on myös aina yhteydessä tiettyyn asiayhteyteen, jolloin asian käsittely ja ymmärtäminen helpottuvat. Tutkivassa oppimisessa oppiminen on nimenomaan oppijan oman toiminnan tulosta, jol-

loin oppimisen perimmäinen tarkoitus toteutuu täysin. (Rauste-von Wright & von Wright, 1998)

Suomalaisen version tutkivasta oppimisesta on kehittänyt Kai Hakkarainen tutkimusryhmineen. Tässä versiossa tutkivan oppimisen keskiössä on tutkittavan asian tai ilmiön todellisen ymmärryksen lisääntyminen sekä muun muassa oman työskentelyn kriittinen arviointi. Perusperiaatteina suomalaisessa tutkivan oppimisen versiossa pidetään esimerkiksi oppimista, jonka päämääränä on ymmärrys, keskeisten käsitteiden riittävää huomiointia sekä oppimisen näkemisestä osana ongelmanratkaisua. Oppilaat saavat itse kehittää tutkimusongelmansa ja ratkaista ne opettajan seurattessa oppilaidensa edistymistä. Perinteisen luokkahuonetyöskentelyn lisäksi tutkivaa oppimista pystytään soveltamaan hyvin myös tietotekniikan yhteydessä. (Hakkarainen et al., 1999)

Tutkivasta oppimisesta tietotekniikkaan yhdistettynä esittelen Gobertin ja muiden (2002) tutkimuksen, jossa he käyttivät hyödyksi WISE-oppimisympäristöä. Oppimisen aiheena oli tulivuorten, vuorijonojen ja maanjäristysten syntyminen ja tutkimukseen osallistuneet oppilaat olivat Yhdysvaltojen itä- ja länsirannikolta. WISE-oppimisympäristön piirtotyökalua käytettiin havainnollistamaan ajattelua. Oppimisympäristön avulla oppilaat pyrittiin saamaan arvioimaan kriittisesti syntyneitä piirroksia.

Oppilaat etsivät ensin itsenäisesti tietoa Internetistä, jonka jälkeen he piirsivät yhteisesti ehdotuksensa esimerkiksi maanjäristyksen syntymisestä. Tämän jälkeen länsirannikon oppilaat lähettivät syntyneen piirroksensa sähköisesti itärannikon oppilaiden arvioitavaksi. Kun itärannikon oppilaat olivat arvioineet piirroksen, saivat länsirannikon oppilaat parannella piirrostaan arvioinnin perusteella. Tutkimuksen lopussa oppilaat pääsivät arvioimaan omaa oppimistaan.

Tutkimustulokset osoittivat, että kaikkien ryhmien osaaminen tulivuorten, maanjäristysten ja vuorijonojen syntyneen suhteen oli parempaa WISE-oppimisympäristön avulla toteutetun kokeilun jälkeen. Eryityisesti ne, jotka tiesivät vähän esimerkiksi maanjäristyksistä paljoa ennen kokeilua oppivat eniten. Kaiken kaikkiaan oppilaat kokivat oppineensa itse piirrettyjen mallien avulla enemmän ja ymmärtäneensä aihetta paremmin kuin mitä he olisivat oppineet perinteisten opetusmetodien avulla.

Hakkaraisen ja muiden (1999) mukaan tutkiva oppiminen vie kuitenkin perinteiseen oppimiseen nähden enemmän aikaa. Tästä johtuen olisi tärkeää, että oppiminen keskittyisi juuri keskeisiin oleviin ajatuksiin ja sille varattaisiin riittävästi aikaa. Koska tutkiva oppiminen vie paljon aikaa, eivät Hakkarainen ja muut (1999) suosittele sitä käytettäväksi lukuvuoden aikana enemmän kuin kahdesta neljään kertaan, 5-7 viikon mittaisissa oppimisjaksoissa kerrallaan.

Tutkivaan oppimiseen liittyy myös muita negatiivisia puolia. Yksi negatiivisista puolista on se, että oppimiskokemuksista ei välttämättä opita mitään tai niistä opitaan asioita, jotka eivät ole olennaisia. Myös tutkivassa oppimisessa korostuva tiedon tulkinta saattaa vahvistaa liiaksi oppijan itseluottamusta, mutta ei itse tietovarastoa. (Rauste-von Wright & von Wright, 1998)

Tässä luvussa esiteltiin tutkivaa oppimista ja siihen liittyviä positiivisia ja negatiivisia asioita. Seuraavassa luvussa käsitellään sosiaalisen median käyttöä opetuksen yhteydessä.

4. Sosiaalinen media opetuskäytössä

Tässä luvussa käydään läpi sosiaalisen median mahdollisuuksia opetuskäytössä ottaen huomioon opettajien lisäksi myös oppilaat. Luvussa pohditaan myös, voisiko tutkivaa oppimista yhdistää oppimiseen, jossa sosiaalinen media on mukana.

4.1. Sosiaalinen media osana opetusta

Sosiaalinen media soveltuu käytännössä hyvin opetuskäyttöön, sillä kaikkia luvussa 2 esiteltyjä sosiaalisen median sovelluksia on mahdollista käyttää osana oppimista. Muutamat sosiaalisen median sovelluksista soveltuvat opetuskäyttöön kuitenkin toisia sovelluksia paremmin.

Esimakua sosiaalisen median opetuskäytön tutkimusten suhteen tarjoaa vuodelta 2006 peräisin oleva Andersonin tutkimus, jossa etäopetuskurssin yhteydessä käytettiin hyväksi sosiaalista mediaa. Osallistujat kokivat sosiaalisen median käytön mielenkiintoisena, mutta kaikki sovellukset eivät sopineet yhtä hyvin oppimiseen kuin toiset. Anderson koki kuitenkin jo tuolloin, että sosiaalinen media tulee löytämään paikkansa osana opetusta.

Pääsin itse todistamaan sosiaalisen median tulemistä osaksi opetusta erään mediakasvatuksen kurssin kautta muutamia vuosia sitten. Kurssin vaihtoehtoisena suoritustapana tarjottiin blogin pitämistä luennoilla käsiteltäviin asioihin nojautuen. Blogi toimi tavallaan luentopäiväkirjan lailla, mutta sillä erotuksena, että kaikkien oli mahdollista lukea ja kommentoida blogia. Blogia piti myös päivittää säännöllisesti. Tämä lisäsi opiskelumotivaatiota ja auttoi hahmottamaan asioita paremmin.

Miinuspuolena blogin käytön suhteen koin sen, että kommentointi eli vuorovaikutuksellisuus muiden kurssilaisten ja opettajan kanssa oli vähäistä. Muilta osin blogi soveltui hyvin kurssin aihepiiriin ja tuki sekä opetusta että oppimista. Lankshear (2006) pitääkin blogien tuloa hyvänä asiana, sillä blogien kautta ihmisten on mahdollista päästä uudelle vuorovaikutuksen tasolle toistensa kanssa.

Blogi soveltuisi melko hyvin edellisessä luvussa käsitellyn tutkivan oppimisen yhteyteen, sillä sen avulla olisi mahdollista julkaista tietoa sekä osaltaan varmistaa myös se, että opettaja voisi seurata oppijan edistymistä blogin välityksellä. Blogin haittapuolena tutkivan oppimisen kannalta olisi kuitenkin se, ettei blogin avulla pysty kokeilemaan, mikä kuuluu olennaisesti tutkivaan oppimiseen, samalla lailla kuin esimerkiksi WISE-oppimisympäristön piirtotyökalua käyttämällä.

Rongas (2008) tuo esille blogien sekä wikien käytön Sosiaalisen median oppimisen oppaassaan. Hänen huomionsa kohteena on eräs suomalainen lukio, jossa käytettiin blogeja ja wikejä osana äidinkielen opetusta. Aluksi muodostettiin perinteiseen tapaan ryhmät ja tämän jälkeen ryhmäläiset tekivät ryhmätöitä blogien muodossa. Lopulta blogit julkaistiin wikeissä, jolloin kaikkien ryhmien esitelmät olivat helposti saatavilla. Kokeilu onnistui hyvin ja sai positiivista palautetta sosiaalisen median ja opetuskäytön yhdistämisestä.

Vaikka blogien ja wikien opetuskäyttö on periaatteessa helppo toteuttaa, tuo se samalla lisähaasteita opettajalle. Lisähaasteista suurin lienee valvonta: opettajan olisi kyettävä valvomaan esimerkiksi blogien sisältöä riittävästi. Valvonnan lisäksi opettajan tulisi osoittaa jollakin tavalla oppilaalle, että opettaja todella lukee hänen blogiaan. Blogeissakin on siis ongelmansa.

Sosiaalisen median piiriin kuuluvien sovellusten käyttöä ovat tarkastelleet myös Stepnyan ja muut (2007). He pohtivat, pitäisikö sosiaalinen media liittää osaksi koulujärjestelmää. Sosiaalisen median soveltuvuutta opetuskäyttöön tutkittiin tutkimuksen aikana virtuaalisen oppimisjärjestelmän kautta.

Stepnyan ja muut (2007) havaitsivat, että oppilaat omaksuivat helposti uuden oppimisjärjestelmän ja sitoutuivat kurssiin. Oppimistehokkuuden suhteen jäi kuitenkin parannettavaa, sillä esimerkiksi oppimisympäristössä käydyt keskustelut eivät suoranaisesti koskettaneet käsiteltäviä aiheita.

Virtuaaliset oppimisympäristöt liittyvät olennaisesti myös tutkivaan oppimiseen. Stepnyan ja muiden (2007) tutkimuksessa esille noussut ongelma virtuaalisten oppimisjärjestelmien keskusteluiden tasosta olisi mahdollisesti korjattavissa tutkivan oppimisen myötä tulevalla ”ehdolla” opettajan valvonnasta.

Myös videoverkkopalvelut voisivat toimia osana opetusta. Tällöin opettaja voisi antaa oppilaille tehtäväksi katsoa tietyn videon esimerkiksi YouTubesta ja tämän jälkeen keskustella videosta muiden oppilaiden kanssa. Videosta heränneet ajatukset voisi kirjata vaihtoehtoisesti myös omaan blogiin.

Videoverkkopalveluiden lisäksi myös kuvajakelupalvelut voisivat soveltua hyvin osaksi opetusta. Oppilaalle voitaisiin esimerkiksi tehdä oma kuvakansio Internetiin, jota oppilas voisi täydentää opettajan ohjeiden mukaisesti.

Tutkiva oppimisen yhdistäminen osaksi sosiaalisen median opetuskäyttöä olisi varmasti kaikin puolin hyvä ratkaisu, sillä kuten edellisestä luvusta kävi ilmi, oppilaat ovat keskeisessä roolissa tutkivassa oppimisessa ja myös opettajan rooli on tärkeä. Sosiaalisen median sovellusten kautta tähän oppimisprosessiin saataisiin lisää mielekkyyttä ja todennäköisesti myös tutkivan oppimisen teorian toivomaa kokeilunhalua. Tutkiva oppiminen ei kuitenkaan soveltuisi jatkuvaan käyttöön, sillä se vie normaalia oppimista enemmän aikaa (Hakkarainen ja muut, 1999).

4.2. Sopivat opetustilanteet sosiaalisen median käytölle

Sosiaalinen media tulisi valita juuri tietyn opetustilanteen mukaan. Pääpaino on sillä, mikä oppiaine on kyseessä ja miten opettaja haluaa oppilaidensa käyttävän sosiaalista mediaa kyseisen oppiaineen parissa.

Omien kokemusteni mukaan sosiaalinen media sopii niin ryhmätyöskentelyyn kuin itsenäisesti tapahtuvaan opiskeluun. Parhaiten se tuntuu soveltuvan pari- tai pienryhmätyöskentelyyn. Tällöin kaikki ryhmän jäsenet pääsevät varmasti osallistumaan työskentelyyn. Mikäli ryhmäkoko on puolestaan liian suuri, vaarana on se, että sosiaalisen median käyttö ei edistä ryhmähenkeä, vaan aiheuttaa ryhmän pirstaloitumista.

Sosiaalisen median ottamista osaksi opetuskäyttöä tukee Laitisen ja Rissanen (2007) Osaatko Wikitellä? -tutkimus korkeakoulusektorilla. Tutkimuksen pääpainona oli tutkia sosiaalisen median käyttöä Kuopion ja Joensuun yliopistoissa sekä Savonia- ja Pohjois-Karjalan ammattikorkeakouluissa. Tutkimustuloksista käy ilmi muun muassa se, että suurin osa vastaajista oli osallistunut verkkokurssille ja erilaiset sosiaalisen median sovellukset tiedettiin hyvin.

Mielenkiintoisinta ja samalla sosiaalisen median opetuskäyttöä tukevaa tutkimustuloksissa oli se, että kyselyyn vastanneista opettajista 79 prosenttia oli käyttänyt sosiaalista mediaa työntekoon. On hyvin mahdollista, että kynnys ottaa sosiaalinen media mukaan ainakin osaksi opetusta on myös madaltunut.

Vaikka tutkimustulokset ovat positiivisia, kannattaa muistaa se seikka, että eniten sosiaalisen median palveluita käyttivät tutkimuksen mukaan nuoret aikuiset. Samankaltainen johtopäätelmä tulee esille myös Whiten (2007) tutkimustuloksista koskien verkkokurssille osallistuneiden sosiaalisen median käyttöä. Yhtenä suurimmista haasteista sosiaalisen median ja opetustyön yhdistämisen kannalta tuleekin olemaan vanhempien opettajien tutustuttaminen sosiaaliseen mediaan.

Tässä luvussa keskityttiin sosiaalisen median mahdollisuuksiin osana opetusta ja pohdittiin myös sosiaalisen median soveltuvuutta tutkivaan oppimiseen. Seuraavassa luvussa syvennyttään tarkemmin opettajan rooliin sosiaalisen median opetuskäytössä.

5. Opettaja ja sosiaalinen media

Tässä luvussa tarkastellaan opettajan roolia sosiaalisen median opetuskäytössä. Samalla tarkastellaan myös sitä, mitä kaikkea opettajalta vaaditaan, mikäli hän haluaa tuoda sosiaalisen median osaksi opetustilannetta. Luvussa pohditaan myös miten opettajan rooli muuttuisi, jos oppimisteorianaa käytettäisiin tutkivaa oppimista.

5.1. Opettajan rooli sosiaalisen median opetuskäytössä

Opettajan roolilla on iso merkitys, kun hän päättää yhdistää opetukseensa sosiaalisen median. Opettaja ei missään nimessä saa jäädä taustalle, vaan hänen tulee olla koko ajan samalla lailla, tai jopa enemmän, näkyvillä kuin hän on perinteisessä luokkahuoneessa tapahtuvassa opetustilanteessa.

Kuten jo edellisessä luvussa tuli esille, olisi tärkeää, että esimerkiksi opettajan käynnit oppilaan blogissa näkyisivät jollakin lailla myös oppilaalle. Tällöin oppilaan motivaatio nähdä vaivaa blogipäivitystensä suhteen olisi aivan eri tasolla, mitä se olisi ilman tietoa opettajan valvonnasta. Samalla myös oppilaan kynnys kysyä kysymyksiä opettajalta blogin välityksellä madaltuisi ja tämän myötä vuorovaikutussuhde kasvaisi.

Opettajan roolia hankaloittaa kuitenkin se, että opettaja saattaa helposti ajautua tarkkailijan osaan oppilaiden blogeja lukiessaan ja unohtaa opetuksellisen vastuunsa. Opettajan tulisi kuitenkin muistaa, että mikäli sosiaalista mediaa käytetään osana oppimista, tulisi sosiaalisen median olla nimenomaan oppimista lisäävä väline, ei opettajan keino päästä helpommalla. Ajatustani tukee myös tutkiva oppiminen, jossa opettajan rooli ja vastuu koetaan tärkeiksi (Rauste-von Wright & von Wright, 1998).

Ferdig (2007) tuo esille opettajan roolista sosiaalisen median parissa myös erään mielenkiintoisen asian – opettaja saattaa vakoilla oppilaitaan verkkoyhteisöpalvelujen, kuten Facebookin kautta, jolloin opettajan näkemys oppilaasta saattaa muuttua täysin. Opettaja voi halutessaan käyttää rooliaan opettajana väärin sosiaalisen median avulla. Käytännössä opettajan tulisi kuitenkin pitää roolistaan opettajana, ei vakoilijana, kiinni.

Opettajan tulee huolehtia myös siitä, että hän kykenee neuvomaan riittävästi oppilaitaan esimerkiksi blogin tai wikin käytön suhteen. Tämän lisäksi opettajan on kyettävä selittämään ero koulussa tapahtuva sosiaalisen median käytön ja kotikäytön välillä.

Sosiaalisen median sovellukset eivät suoraan sovi tietyn luokka-asteen käyttöön, joten opettaja joutuu näkemään ehkä normaalia enemmän vaivaa sopivan sosiaalisen median sovelluksen löytämiseen ja oman roolinsa korostamiseen tietyn sosiaalisen median sovelluksen parissa. Tutkivan oppimisen kannalta tä-

tä ongelmaa voisi pienentää laittamalla oppilaita kokeilemaan itse sosiaalisen median sovelluksia ja valitsemaan kokeilujen kautta heille sopivat sovellukset.

Vaikka opettajan rooli osana sosiaalista mediaa kuulostaa haastavalta, on hänen mahdollista hakea sen käyttöön tukea. Erityisesti Internetistä löytyy paljon erilaisia sosiaalista mediaa käsitteleviä opuksia ja verkostoja, jotka on suunniteltu opettajaa ajatellen.

Oppaista mainittiin jo edellisen luvun yhteydessä Sosiaalisen median oppimisen opas, joka sisältää paljon tietoa sosiaalisesta mediasta ja sen käytöstä opetuksessa. Opas antaa myös paljon vinkkejä ja konkreettisia esimerkkejä, sekä tarjoaa hyödyllistä tietoa esimerkiksi tietoturvaan liittyen. (Rongas, 2008)

Toinen Internetistä löytyvä, opettajalle hyödyllinen apuväline sosiaaliseen mediaan tutustumiseen on Sosiaalinen media oppimisen tukena -verkosto, joka tarjoaa erilaisia sosiaalisen median palveluita kuten blogin ja keskustelupalstan osallistujilleen. Verkostoa käyttämällä opettajan on helppo kokeilla, mitä sosiaalisen median avulla voi tehdä ja saada vinkkejä omaan opetukseensa muilta käyttäjiltä. (Venäläinen, 2009)

5.2. Sosiaalisen median vaatimukset opettajalle

Siinä missä sosiaalinen media vaikuttaa opettajan rooliin opettajana, asettaa se myös tiettyjä vaatimuksia. Opettajan on hallittava tietokoneen ja Internetin käyttö sujuvasti, jotta sosiaalisen median käyttöönotto onnistuu. Opettajan on kyettävä myös neuvomaan oppilaitaan, mikäli ongelmatilanteita sosiaalisen median parissa syntyy. Jotta sosiaalisen median ottaminen mukaan opetukseen onnistuu, on opettajan osattava suhtautua avoimesti uusiin opetusmenetelmiin.

Robertsonin (2008) mukaan opettajat suhtautuivat positiivisesti sosiaalisen median, tässä tapauksessa wikien, ottamiseen osaksi opetustilannetta. Opettaja pitivät wikejä tutkimustulosten mukaan muun muassa helppokäyttöisinä ja kokivat opettajan roolin tärkeäksi. On kuitenkin muistettava, että kyseessä oli ainoastaan pientä ryhmää koskettanut tutkimus.

Sosiaalinen media asettaa selvästi lisävaatimuksia opettajalle ja pakottaa opettajan sisäistämään uusia asioita. Opettaja voi kuitenkin vaikuttaa itse siihen, miten nopeasti hän ottaa käyttöön uusia opetusmenetelmiä, kuten esimerkiksi luokan yhteisen wikin, vai ottaako ollenkaan. Ideaalitalanne olisi, jos opettaja kykenisi yhdistämään sosiaalista mediaa tavalliseen opetukseen ja saamaan oppimisesta tätä kautta uuden ulottuvuuden esille. Perinteisen oppimisen korvaaminen osaksi tutkivalla oppimisella tarjoaisi sekä oppijalle että opettajalle enemmän myös sosiaalisen median käytön suhteen.

Vaikka sosiaaliseen mediaan suhtaudutaan tässä tutkielmassa aiemmin esiteltyjen tutkimusten mukaan suhteellisen nopeasti, on myös täysin vastakkaisia mielipiteitä olemassa. Carroll ja Waggoner (2008) tutkivat lastentarhaikäisis-

tä lukioikäisiin opettavia opettajia ja heidän suhtautumistaan sosiaaliseen mediaan sekä sen yhdistämiseen opetukseen. Opettajat pitivät perinteistä opetustapaa parempana kuin sosiaalisen median mukaan tuomista opetukseen. Sosiaalinen media koettiin erityisen hankalaksi, mikäli tietotekniikan käyttö ei ollut hallussa. Tämänkin tutkimuksen kohdalla on kuitenkin hyvä muistaa, että se tehtiin pienelle ryhmälle. (Carroll ja Waggoner, 2008)

Kuten edellä olleista esimerkeistä nähdään, kokevat toiset opettajat sosiaalisen median hyvänä, toiset huonona asiana. Tämä ristiriita tulee varmasti säilymään myös tulevaisuudessa. Osaltaan juuri opettajien ennakoasenteet tulevat vaikuttamaan sosiaalisen median leviämiseen.

Tässä luvussa tutkittiin opettajan sosiaalista mediaa opettajan rooliin sidottuna sekä tarkasteltiin hieman tutkivan oppimisen soveltuvuutta sosiaalisen median opetuskäyttöön. Seuraavassa, tutkielman päättävässä luvussa, summataan sosiaalisen median opetuskäytön positiivisia ja negatiivisia puolia yhteen ja pohditaan, millainen tulevaisuus sosiaalista mediaa odottaa sen opetuskäytön suhteen.

6. Yhteenveto

Tässä tutkielmassa käsiteltiin sosiaalista mediaa opetuskäytössä erityisesti opettajan roolista tarkasteltuna. Lähdimme liikkeelle sosiaalisen median määrittelystä, jatkoimme tutkivaan oppimiseen, josta siirryimme sosiaalisen median tutkimiseen suhteessa opetuskäyttöön ja päädyimme lopulta tarkastelemaan, mitä vaatimuksia sosiaalinen media tuo opettajalle ja miten se muuttaa opettajan roolia.

Lopuksi on aika tarkastella, mitä hyviä ja huonoja puolia sosiaalisen median yhdistämisellä opetukseen on sekä millainen tulevaisuus sosiaalista mediaa ja sen käyttöä opetustilanteissa mahdollisesti odottaa.

Hyvinä puolina sosiaalisessa mediassa voidaan pitää ennen kaikkea sen vuorovaikutuksellisuutta sekä sitä, että sosiaalisen median avulla opettaja tai oppilas ei ole enää tiettyyn paikkaan tai aikaan sidottu. Hyvää on myös se, että sosiaalisen median avulla monesta kurssista saadaan tehtyä mielekäs ja oppilaita kiinnostava – itse oppiminen tapahtuu siinä sivussa, kun oppilas esimerkiksi keskustelelee kurssin keskustelupalstalla muiden oppilaiden kanssa.

Yhtenä huonoimmista puolista sosiaalisessa mediassa, niin opetuskäytön kuin muun käytön osalta, on se, että sosiaalisen median käyttöä on hankala valvoa riittävästi. Opettajat saattavat myös kokea sosiaalisen median tuomat uudistukset liian suuriksi, jolloin sosiaalinen media nähdään enemmän negatiivisena kuin positiivisena asiana. On myös mahdollista, että sosiaalisen median

sovellukset ovat vielä liian hankalakäyttöisiä, eivätkä ne palvele kaikkia ikäryhmiä tasapuolisesti.

Jotta sosiaalinen media pystyisi kehittymään ja saavuttamaan paremman aseman opetuskäytössä, olisi hyvä tehdä muutamia parannuksia tämänhetkiseen tilanteeseen. Jokaisen opettajan tulisi hallita tietokoneen käyttö sujuvasti, jotta sosiaalisen median tuominen osaksi opetusta olisi mahdollista. Jokaisessa koulussa tulisi myös olla riittävästi tietokoneita, jotta sosiaalisen median käyttö koulussa onnistuisi.

Todellisuudessa nämä asiat tuskin tulevat onnistumaan vielä vuosiin ja tällöin sosiaalisen median ottaminen osaksi opetusta viivästyy edelleen, vaikka sosiaalisen median käyttö muuten onkin vilkasta.

Viiteluettelo

- Anderson, T. (2006). Teaching a Distance education course using educational social software. Retrieved September 14, 2009, from <http://terrya.edublogs.org/2006/01/02/teaching-a-distance-education-course-using-educational-social-software/>.
- Carroll, J.B. & Waggoner, J.C. (2008). Introducing Social Software to K-12 Teachers in a Research Setting. *MERLOT Journal of Online Learning and Teaching*, 4 (1), 67-72.
- Facebook. Retrieved December 3, 2009, from <http://www.facebook.com>.
- Ferdig, R.E. (2007). Editorial: Examining Social Software in Teacher Education. *Journal of Technology and Teacher Education*. 15(1), 5-10.
- Gobert, J., Slotta, J., Pallant, A., Nagy, S. & Targum, E. (2002). A WISE Inquiry Project for Students' East-West Coast Collaboration. (Paper presented in the Annual Meeting of the American Educational Research Association, New Orleans, April 1-5, 2002.) Retrieved December 7, 2009, from http://mtv.concord.org/publications/online_learning.pdf.
- Hakkarainen, K., Lipponen, L., Ilomäki, L., Järvelä, S., Lakkala, M., Muukkonen, R., Rahikainen, M., & Lehtinen, E. (1999). Tieto- ja viestintätekniiikka tutkivan oppimisen välineenä. Tarkistettu 7. joulukuuta, 2009, http://www.helsinki.fi/science/networkedlearning/julkaisut/tvt_tutkivan_oppimisen_valineena.pdf.
- Lankshear, C. (2006). Blogging as Participation: The active sociality of a new, literacy (Paper presented in AERA: the American Educational Research Association Annual Meeting, San Francisco, April 11, 2006). Retrieved September 14, 2009, from <http://www.geocities.com/c.lankshear/bloggingparticipation.pdf>.

- Messenger. Retrieved December 3, 2009, from <http://windowslive.com/Desktop/messenger>.
- Moodle. Retrieved December 3, 2009, from <http://www.moodle.com>.
- Osaatko Wikitellä? Selvitys sosiaalisen median käytöstä korkeakoulusektorilla, vuonna 2007. Tarkistettu 14. syyskuuta, 2009, http://www.smoot.fi/dokumentteja/Osaatko_wikitella_svy_050907.pdf.
- Rauste-von Wright, M. & von Wright, J. (1998). Oppiminen ja koulutus.
- Robertson, I. (2008). Learners' attitudes to wiki technology in problem based, blended learning for vocational teacher education. *Australasian Journal of Educational Technology*, 24 (4), 425-441.
- Rongas, A. (2008). Sosiaalinen media oppimisen tukena. Tarkistettu 14. syyskuuta, 2009, <http://sosiaalinenmedia.googlepages.com/home>.
- SMOOT, Sosiaalinen media opetuksen ja oppimisen tukena -projekti. Tarkistettu 14. syyskuuta, 2009, <http://www.smoot.fi>.
- Stepanyan, K., Mather, R. & Payne, J. (2007). Integrating social software into Course Design and Tracking Student Engagement: Early Results and Research Perspectives. In G. Richards (Ed.), *Proceedings of E-Learn 2007: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, 7386-7395.
- Venäläinen, V. (2008). Sosiaalinen media oppimisen tukena -verkosto. Tarkistettu 1. joulukuuta, 2009, <http://sometu.ning.com>.
- White, D. (2007). Results of the 'Online Tool Use Survey' undertaken by the JISC funded SPIRE project 2007, Retrieved September 14, 2009, from <http://tallblog.conted.ox.ac.uk/wp-content/uploads/2007/03/survey-summary.pdf>.
- Wikipedia. Retrieved December 3, 2009, from <http://www.wikipedia.org>.
- WISE. Retrieved December 7, 2009, from <http://www.wise.berkeley.edu>.
- YouTube. Retrieved December 3, 2009, from <http://www.youtube.com>.

Ohjaaja: Raimo Hälinen

Mainossokeus – katsaus tutkimustuloksiin

Tony Keisala

Tiivistelmä.

Tässä tutkielmassa käsitellään Internet-sivustoilla olevien mainosten havaitsemista. Tutkielma pyrkii tarkastelemaan mainossokeudesta (banner blindness) tehtyjä tutkimuksia ja niiden tuloksia. Tutkimustulokset ovat osoittaneet, että mainosbanneri ei ole tehokas ärsyke vangitsemaan käyttäjän tarkkaavaisuutta. Tästä huolimatta mainosbannerit ovat suosittu mainoskeino Internet-sivuilla. Mainosbannereiden havaitsemisesta on saatu myös vastakkaisia tutkimustuloksia. Tutkimustulokset joiden mukaan mainosbannerit havaitaan Internet-sivuilla suhteellisen hyvin, liittyvät vapaaseen Internet-sivujen selaamiseen. Tällöin käyttäjä ei pyri selaamisellaan saavuttamaan ennalta asetettua tavoitetta.

Avainsanat ja -sanonnat: mainosbanneri, tarkkaavaisuus, klikkausprosentti, tunnistaminen, muistista palauttaminen

CR-luokat: H.1.2, H.5.2

1. Johdanto

Internet-sivut ovat pullollaan mainosbannereita. Mainosbannerit ovat usein värikkäitä, animoituja ja paljon huomiota herättäviä visuaalisia elementtejä. Mainosbannerit yleistyivät pian Internetin kotitalouksiin leviämisen jälkeen. Mainosbannereiden huomioimisen tehokkuutta on tutkittu erilaisilla koeasetelmilla, joihin liittyy mainosten tunnistus, muistista palauttaminen ja klikkausprosentti (click-through rate).

Mainosbannereiden havaitsemiseen ja muistamiseen liittyvät tutkimustulokset ovat osoittaneet mainossokeuden (banner blindness) olemassaolon. Yksittäisistä tutkimuksista on kuitenkin vaikea tehdä päteviä kokonaisvaltaisia johtopäätöksiä mainossokeuden kokonaiskuvasta. Aluksi tutkimukset keskittyivät bannereiden visuaalisiin ominaisuuksiin ja niiden havaitsemiseen yksinkertaisissa koeolosuhteissa. Myös mainosbannereiden erittäin alhaiset klikkausprosentit (0,5 – 3 % luokkaa) antoivat ymmärtää, että mainosbannerit ovat todella heikosti havaittavia visuaalisia elementtejä Internet-sivuilla. Pian kuitenkin huomattiin, että bannereiden havaitseminen ja mainossokeus pitävät sisällään paljon muutakin kuin bannereiden sisäisiä ominaisuuksia tai klikkausprosentteja. Edellämainituista syistä johtuen onkin tärkeää kartoittaa kokonaisvaltaisemmin mainossokeuteen aiheuttavia syitä ja olosuhteita.

Tässä tutkielmassa käydään läpi bannereiden visuaalisen ulkoasun aiheuttamia tekijöitä mainossokeuteen ja navigointiin sekä ihmisen havaitsemiseen vaikuttavia tekijöitä, jotka vaikuttavat siihen, kuinka ihminen kerää informaatiota. Nämä voivat edelleen selittää miksi tietynlaiset bannerit jäävät havaitsematta. Esimerkiksi Pagendam ja Schaumburg (2001) esittävät erilaisten selaustyöliien vaikuttavan suuresti bannereiden havaitsemiseen. Yesilada ja muut (2008) puolestaan esittävät hypoteesin, jonka mukaan Internet-sivut sisältävät erilaisia visuaalisia sektioita, joilla he olettavat olevan yhtä suuri vaikutus mainossokeuden synnylle kuin bannereiden visuaalisilla ominaisuuksilla. Tämä tutkielma pyrkii antamaan monipuolisen katsauksen mainossokeutta käsittelevään kirjallisuuteen ja kartoittamaan erilaisia tekijöitä, jotka johtavat bannereiden heikkoon muistettavuuteen ja tunnistamiseen.

2. Bannerit

Mainosbannerit ovat Internet-sivuilla esiintyviä mainoksia. Niiden tarkoitus on vangita käyttäjän tarkkaavaisuus värikkäällä sekä mahdollisesti animoidulla ulkoasulla. Internet-sivustot sisältävät erilaisia bannereita, jotka voidaan luokitella eri kategorioihin tarkoituksen tai sisällön perusteella. Kaikki bannerit eivät ole mainosbannereita. Tässä tutkielmassa kuitenkin keskityn pääasiallisesti mainosbannereihin. Erityyppisiä bannereita ovat muun muassa animoidut mainosbannerit, staattiset mainosbannerit sekä välkkyvät tekstibannerit. Yleisimmin bannereita on käytetty staattisina kuvina JPEG-, GIF- ja PNG-formaateissa. Myöhemmin myös animaatioita ja audiota sisältävät bannerit ovat yleistyneet. Animaatioiden on yleisesti uskottu lisäävän bannereiden näkyvyyttä sekä käyttäjän tarkkaavaisuuden suuntautumista bannereihin. Kuvassa 1 on esimerkkejä erilaisista mainosbannereista.

Mainosbannereiden näkyvyyttä on tutkittu mittaamalla, kuinka monta kertaa ne ovat keskimäärin tulleet valituiksi (klikkausprosentti) sekä erilaisilla muistiinpalautukseen ja tunnistukseen liittyvillä koeasetelmilla.



Kuva 1. Esimerkki erilaisista mainosbannereista (Bayles, 2002).

2.1. Miksi bannereita ei aina havaita web-sivuilla?

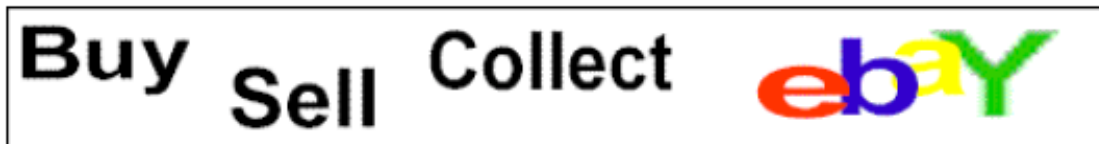
Kun mainosbannerit ilmestyivät Internetiin, alettiin eri tutkimuksissa mitata bannereiden klikkausprosenttia. Williamson (1996) esitti keskimääräiseksi klikkausprosentiksi 2,1 % vuonna 1996. Sutherland (1999) esitti, että klikkausprosentti on tippunut 0,5 %:iin vuonna 1999. Tästä voisi päätellä, että bannereita ei havaita Internet-sivuilla niin usein kuin ehkä voisi olettaa.

Bayles (2000) kuitenkin esittää, että bannereita havaitaan Internet-sivuilla, mutta niitä ei aina ymmärretä tai ne vain sivuutetaan tietoisesta ajattelusta. Syitä bannereiden heikkoon havaitsemiseen voi olla useita. Näitä syitä eritellään tässä kohdassa. Tutkimustulokset ovat osittain ristiriitaisia: tietynlaisia bannereita havaitaan selkeästi paremmin kuin bannereita keskimäärin. Mielenkiintoista onkin, että yksinkertaisemmat staattiset bannerit havaitaan monesti paremmin kuin värikkäät ja animoidut. Edellä mainittuun asiaan ei oikein tunnu löytyvän selvää vastausta, mutta on turvallista olettaa tämän johtuvan yksinkertaisten mainosbannereiden sekoittumisesta muihin Internet-sivun elementteihin. On siis mahdollista, että yksinkertaisimmat mainosbannerit nähdään esimerkiksi tekstielementteinä eikä bannereina.

2.2. Web bannereiden muistista palauttaminen ja tunnistaminen

Bayles (2000) tutki Ebayn ja Amazonin mainosbannereiden muistamista ja tunnistamista. Tutkimuksessa käytettiin staattisia sekä animoituja mainosbannereita. Kokeeseen osallistui 12 miestä ja 23 naista, ikävuosien 18 ja 44 väliltä. Mainokset asetettiin keskimäärin keskelle Internet-sivua. Koeolosuhteet satunnaisettiin siten, että jokainen koehenkilö katsoi Ebayn ja Amazonin bannereita satumanvaraisesti. Koehenkilöt suorittivat tutkimuksen aikana neljä etsintätehtävää kahdella eri Internet-sivulla.

Etsintätehtävien jälkeen muistista palauttamista mitattiin koehenkilöiden vapailla piirustuksilla. Koehenkilöitä pyydettiin piirtämään parhaansa mukaan tyhjälle paperille Internet-sivujen malli ja grafiikat. Bannereiden tunnistustehtävä suoritettiin näyttämällä koehenkilöille etsintätehtävien jälkeen 12 erilaista banneria, joista koehenkilöiden tuli valita Internet-sivuilla tehtävien aikana näkemänsä bannerit (Bayles, 2000). Kuvissa 2 ja 3 on esitetty tutkimuksessa käytettyjä bannereita.



Kuva 2. Kokeessa käytetty Ebay:n mainosbanneri (Bayles, 2000).



Kuva 3. Kokeessa käytetty Amazonin mainosbanneri (Bayles, 2000).

2.2.1. Muistista palauttaminen

Tutkimustulokset osoittavat, että lähes puolet (43 %) koehenkilöistä palauttivat jotakin alueesta, jossa Amazonin tai Ebayn mainosbannerit sijaitsivat. Ebayn mainosbannerin ollessa ruudulla koehenkilöistä 14 % muisti nähneensä jonkin bannerin. Koehenkilöistä 32 % muistivat bannerin sekä Ebayn yrityksen nimen bannerista. Koehenkilöistä 57 % ei muistanut mitään Internet-sivuston bannerialueesta, kun Ebayn mainosbanneri oli ruudulla. Amazonin mainosbannerin sekä yrityksen nimen muisti 17 % koehenkilöistä, 23 % koehenkilöistä muisti nähneensä jonkin bannerin. Amazonin mainosbanneri jäi kokonaan huomaamatta 60 %:lla koehenkilöistä. Ainoastaan seitsemän koehenkilöä muisti mainosbannerin staattisen tai animoidun tilan oikein. Seitsemästä oikein muistetusta tilasta kuusi kohdistui animoituihin mainosbannereihin (Bayles, 2000).

2.2.2. Tunnistaminen

Jokainen koehenkilö tunnisti vähintään yhden mainosbannerin oikein: 26 % koehenkilöistä tunnisti yhden mainosbannerin oikein, 74 % koehenkilöistä tunnisti kaksi mainosbanneria oikein. Edelliset lukemat eivät huomioi bannereiden tilan (staattinen vai animoitu) oikeata tunnistamista, mutta lukemat kertovat, tunnistivatko koehenkilöt mainosbannereista oikeat yritykset (Bayles, 2000). Koehenkilöistä 57 % tunnisti mainosbannerin tilan oikein, tulos oli sama animoitujen sekä staattisten bannereiden kohdalla.

Bayles (2000) osoittaa tutkimuksessaan, että mainosbannereiden tunnistaminen oli suhteellisen korkeaa luokkaa. Mielenkiintoista on myös huomata se, että Ebayn graafisesti yksinkertaisempi mainosbanneri muistettiin ja tunnistettiin paremmin kuin Amazonin mainosbanneri. Ebayn mainosbannereiden kohdalla muistamista tehosti suuresti, mikäli yritys oli entuudestaan tuttu koehenkilölle. Vastaavaa ilmiötä Amazonin mainosbannerin kohdalla ei ilmentynyt (Bayles, 2000).

Ainoastaan kolme (9 %) koehenkilöistä muisti ja tunnisti molemmat mainokset sekä yritykset oikein, sekä muistivat ja tunnistivat mainosbannerin oikean tilan. Toisaalta koehenkilöt, jotka eivät pystyneet palauttamaan muistista kummankaan yrityksen banneria tai bannerin oikeata tilaa (40 %), tunnistivat kuitenkin bannereita melko hyvin. Näistä koehenkilöistä 79 % tunnisti oikein kaksi mainosbanneria. Tulokset myös osoittavat, että koehenkilöt, jotka tunnistivat ainoastaan yhden mainosbannerin oikein (26 %), tunnistivat mainosbannerin animoidussa tilassa.

2.3. Animaation käytön vaikutus muistamiseen

Jotkin tutkimukset ovat tarkastelleet bannermainosten muistettavuutta. Bayles (2002) osoitti, että animoidut bannerit eivät korreloi muistista palauttamisen kanssa. Toisaalta Bayles käytti testissään ainoastaan kahta banneria, joten tuloksia on vaikea yleistää. Pagendarm ja Schaumburg (2001) puolestaan ehdottavat, että Internetin satunnaiselaajat muistavat bannerit paremmin kuin tavoitehakuiset Internetin selaajat. Ryhmä satunnaiselaajia selasi 55 Internet-sivulla omien mielenkiinnonkohteidensa mukaan, ryhmä tavoitehakuista käyttäjiä selasi samoja sivuja vastatakseen listaan kysymyksiä. Kummallekin ryhmälle esitettiin Internet-sivuilla 16 erilaista banneria. Satunnaiselaajat palauttivat selvästi enemmän aiheita ja tuotteita kuin tavoitehakuinen ryhmä (3 tai 17 kertaa enemmän), mutta keskimääräinen palauttaminen bannereiden yksityiskohdista oli satunnaiselaajillakin todella heikkoa, lukema oli keskimäärin 1 - 2 banneria (Pagendarm & Schaumburg, 2001).

2.4. Huomioitavaa mainosbannerien suunnittelussa

Monet mainosbannerien suunnittelijat suosittelevat animaatioiden käyttöä mainosbannereissa. Animaation on sanottu lisäävän mainosbannereiden vaikutusta. Psykologisesti on todistettu, että liikkeessä olevat kohteet vetävät ihmisen tarkkaavaisuutta tehokkaasti puoleensa. Mutta miten animaatiot todellisuudessa vaikuttavat mainosbannereiden havaitsemiseen? Bayles (2002) esittää, että Internet-käyttäjät kyllä huomaavat ja muistavat animaation, mutta he eivät välttämättä liitä sitä mainoksiin, joissa animaatio on ollut. Koehenkilöt usein

myös palauttivat animoidun mainosbannerin sijainnin väärin. Mainosbannereiden animaation voidaan väittää lisäävän bannereiden näkyvyyttä. Koehenkilöt raportoivat nähneensä animaatioita Internet-sivuilla, kun heille on näytetty animoituja mainosbannereita, mutta heidän tekemänsä havainnot eivät näyttäisi korreloivan mainosten tai mainoksissa olevien yritysten muistamiseen. Kyseessä on enemmänkin vain liikkeen havainnoinnista vailla liitosta mainoskontekstiin. Myös Benway ja Lane (1998) osoittavat tutkimuksessaan, että mainossokeus ilmenee myös animoiduissa bannereissa. Heidän tutkimuksessaan koehenkilöiden tuli hakea informaatiota Internet-sivulta, jossa informaatio oli saatavilla ainoastaan banneria klikkaamalla. Koehenkilöt raportoivat informaation haun olleen vaikeaa bannereiden avulla. Benway ja Lane (1998) ehdottavat, että mikäli bannereita käytetään linkkeinä, niin niiden tulisi muistuttaa mahdollisimman paljon linkitettyjä tekstejä. Toisaalta, tämä väite ei ole saanut vahvistusta. Tärkeää on myös huomata, että mainossokeus ei välttämättä rajoitu ainoastaan bannereihin, vaan myös taulukkomainen tekstielementit voivat aiheuttaa mainossokeutta. Internet-selaaja siis saattaa erehtyä luulemaan tekstielementtiä mainosbanneriksi tekstielementin bannerimaisesta ulkoasusta johtuen.

Colbert ja muut (2008) ehdottavat Follow-on -mainosten käyttöä perinteisten mainosbannereiden sijaan. Follow-on -mainokset ovat erillisiä mainosikkunoita, jotka avautuvat selatun Internet-sivun päälle. Follow-on -mainokset ovat yleensä Flash-animaatioita. Colbert ja muut (2008) esittävät lupaavia tutkimustuloksia follow-on -mainosten tehokkuudesta, tai ainakin niiden paremmuudesta perinteisten mainosbannereiden suhteen. Tutkimuksen päätulokset osoittavat, että klikkausprosentti on selkeästi suurempi follow-on -mainoksissa kuin perinteisissä bannermainoksissa. Koehenkilöt raportoivat myös Internet-sivuston yleisen käytettävyyden olleen parempi sivustoilla, joilla on follow-on -mainoksia bannermainosten sijaan. Follow-on -mainokset kuitenkin tarvitsevat ehdottomasti lisätutkimusta, eikä niiden voida vielä väittää olevan lääke mainossokeuteen. Myös väite paremman käytettävyyden puolesta kuulostaa toistaiseksi melko hataralta.

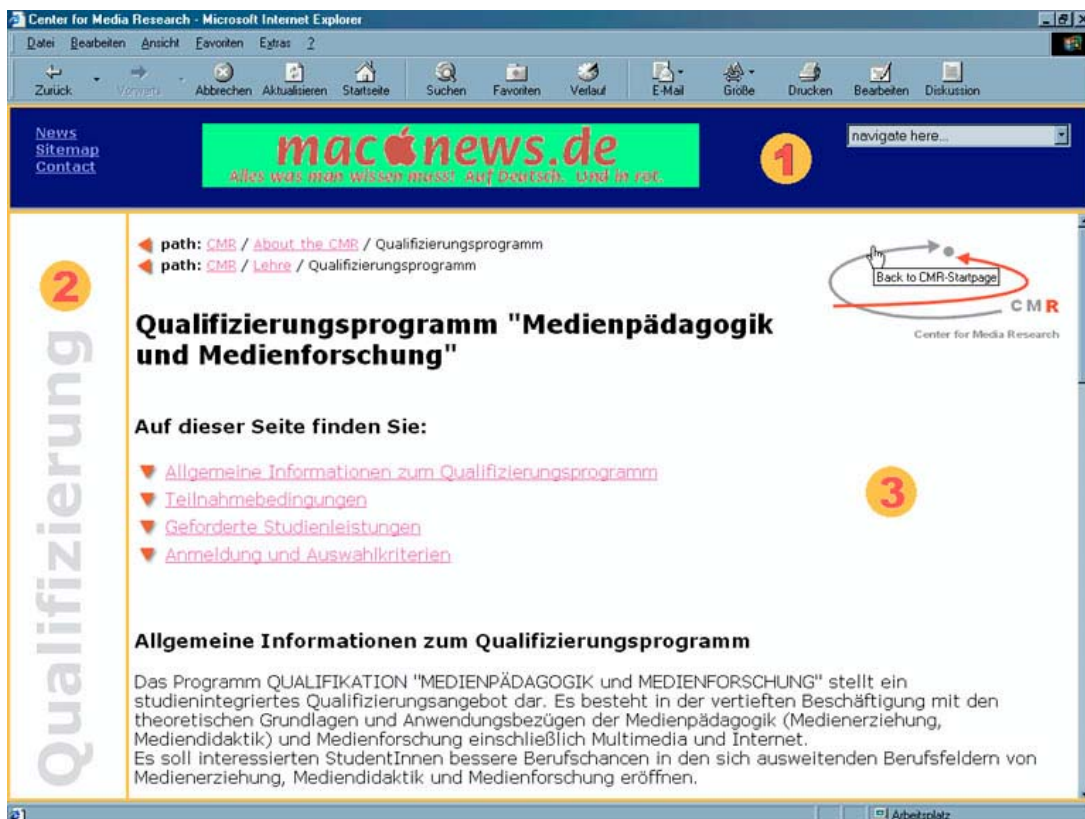
3. Navigointityylit web-sivuilla

3.1. Tavoitehakuinen informaation haku ja vapaa selaus

Tässä kohdassa tarkastellaan tavoitehakuisen Internetin selaamisen sekä vapaan Internetin-selaamisen yhteyttä mainossokeuteen. Tavoitehakuisesti Internetiä selaavalla henkilöllä on jokin tietty tavoite toiminnassa. Tämä tavoite on usein jonkin tietyn informaation löytäminen. Tällöin henkilö on usein keskittynyt tekemiseensä, jolloin myös toimintaa häiriöivät tekijät suodatetaan parem-

min informaation prosessoinnin ulkopuolelle. Pagendarm ja Schaumburg (2001) esittivät hypoteesin, että vapaasti Internetiä selaavat henkilöt huomaavat ja muistavat mainosbannereita tavoitehakuksia Internetin-selaajia enemmän. Tämä johtuu heidän mukaansa siitä, että mainosbannerit prosessoidaan syvällisemmin, kun Internetin-selaamisessa ei ole ennalta asetettua tavoitetta. Mainosbannereiden havaitsemisessa ei siis välttämättä ole suuria eroja, erot syntyvätkin informaation prosessoinnissa, jotka vaikuttavat muistamiseen ja myöhempään tunnistamiseen.

Pagendarm ja Schaumburg (2001) tekivät kokeen, jolla pyrittiin tutkimaan mainosbannereiden muistista palauttamisen ja tunnistamisen poikkeavuutta tavoitehakuksella Internetiä selaavan (GDSI-ryhmä) ja vapaasti Internetiä selaavan (ALB-ryhmä) ryhmien välillä. Kokeeseen osallistui 32 henkilöä, jotka olivat iältään 19–43 -vuotiaita. Heidät jaettiin itsearviointin perusteella tavoitehakuksiin ja vapaasti Internetiä selaavien ryhmiin. Molempiin ryhmiin allokoitui 16 koehenkilöä. Koe suoritettiin modifioidulla mediatutkimuskeskuksen Internet-sivulla. Sivustolla oli mahdollista tutkia 55 eri HTML-sivua, joilla oli tekstejä, projekteja, kurssimateriaaleja, tutkimusaiheita sekä taulukoita ja kuvia, jotka liittyivät mediatutkimuskeskuksen työntekijöihin. Kuvassa 4 on esimerkki testisivuista.



Kuva 4. Kuvankaappaus testisivusta. (Pagendarm & Schaumburg, 2001)

Koehenkilöille esitettiin 16 animoitua graafista bannerimainosta, kutakin 45 sekunnin ajan. Mainosbannerit esiintyivät navigointikehyksessä. Tavoitehakukselle ryhmälle annettiin lista kysymyksiä, joihin heidän tuli hakea vastaukset. Vapaasti Internetiä selaavien ryhmää ohjeistettiin selaamaan mediatutkimuskeskuksen Internet-sivustoa omien mieltymysten mukaan. Viidentoista minuutin selaamisen jälkeen koehenkilöitä pyydettiin suorittamaan vapaan muistista palautuksen ja tunnistamisen testi. Vapaa muistista palauttamisen testi jakaantui kolmeen alaosaan: mainostetut yhtiöt, tuotteet tai palvelut; mainoskopioidut ja sloganit; kuvakertomukselliset vaikuttimet ja perusvärit. Koehenkilöitä pyydettiin kirjoittamaan paperille muistamiensa yritysten tuotteiden, bannereiden sloganit tai teksti sekä bannereiden värit. Tunnistustehtävät suoritettiin HTML-sivun avulla, jossa mainosbannerit esitettiin samankaltaisen kontekstin mukana kuin aikaisemmin kokeessa. Koehenkilöiden täytyi katsoa mainosbannereita ja arvioida viiden pisteen mitta-asteikolla, kuinka varmasti he olivat nähneet mainosbannerin aikaisemmin testissä. Matala arvo (0) tarkoittaa epäröintiä ja suuri arvo (4) täyttä varmuutta, että mainosbanneri on esiintynyt testissä aikaisemmin.

3.2. Tuloksia

Tuloksista on nähtävissä, että ABL-ryhmän koehenkilöt muistivat ja tunnistivat selkeästi GDSI-ryhmää enemmän mainosbannereita ja niiden ominaisuuksia. Ero näiden ryhmien välillä oli hyvin selkeä yritysten tuotteiden ja palveluiden, kuvakertomuksellisten vaikuttimien ja perusvärien sekä tunnistettujen mainosbannereiden kohdalla. Ero ei ollut tilastollisesti merkittävä mainoskopioiden ja sloganeiden kohdalla. Tutkimustulokset on esitetty tarkemmin taulukossa 1.

	ABL-ryhmä	GDSI-ryhmä	
Muistetut mainostetut yritykset, tuotteet ja palvelut	1.00 (1.41)	0.06 (0.25)	p<.01
Muistetut mainoskopioidut ja sloganit	0.63 (1.31)	0.25 (0.45)	n.s.
Muistetut kuvakertomukselliset vaikuttimet ja perusvärit	1.87 (1.59)	0.63 (0.72)	p<.05
Tunnistetut mainosbannerit	1.52 (0.92)	0.75 (0.57)	p<.01

Taulukko 1. Keskiarvoja, muistista palauttaminen ja tunnistaminen (Pagendarin & Schaumburg, 2001).

Pagendarin ja Schaumburgin hypoteesi näyttäisi siis pitävän paikkansa. Kuten oli odotettavissa, ABL-ryhmän mainosbannereiden tunnistusarvot olivat selkeästi GDSI-ryhmää korkeammalla. Tämä luultavimmin johtuu navigointi-

tyylieroista. Päämäärättömästi Internetiä selaavat henkilöt altistuvat helpommin mainosbannereille, koska bannereiden väri- ja animaatioinformaatio laukaisee tarkkaavaisuuden suuntautumisen bannereita kohti – joka johtuu bottom-up -tyylisestä informaation prosessoinnin prosessista. Tavoitehakuisesti Internetiä selaavat henkilöt puolestaan tuntuvat soveltavan kognitiivista kaavamaisesta informaation prosessointia, joka vaimentaa bannereiden syvällisemmän prosessoinnin (Pagendam & Schaumburg, 2001). Toisin sanoen, bannerit luultavasti havaitaan, mutta niitä ei prosessoida, jolloin myöskään muistijälkeä ei tallennu. Tutkimustuloksista on mielenkiintoista huomata mainoskopioiden ja sloganeiden huono muistettavuus myös ABL-ryhmän osalla. Tämä löydös on mahdollisesti selitettävissä ”picture superiority effect” (Nelson et al., 1976) ja ”dual coding theory” (Paivio, 1986) -teorioilla – näiden teorioiden mukaan kuvat yleensä palautetaan muistista teksti-informaatiota paremmin. On myös mahdollista, että bannereiden havainnointiaika on liian lyhyt myös päämäärättömästi Internetiä selaavien henkilöiden kohdalla – tällöin ei tapahdu tarpeeksi syvää informaation prosessointia mainosbannereiden animaatioitujen tekstiosien kohdalla. Edellä mainittua hypoteesia täytyy kuitenkin vielä tutkia lisää (Pagendam & Schaumburg, 2001).

Pagendam ja Schaumburg (2001) osoittavat, että bannereiden optimaalisen suunnittelun sijaan tulisi enemmän keskittyä tutkimaan ihmisten aikomuksia ja käyttäytymistä heidän selatessaan Internet-sivuja.

3.3. Skeemat

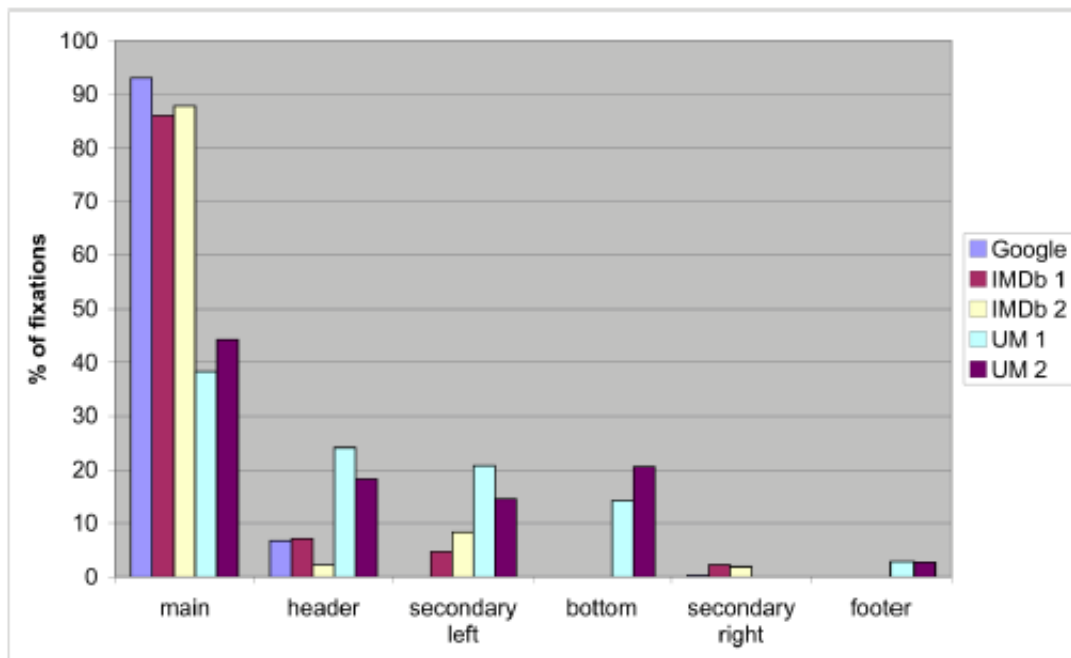
Navigointityylillä on oletettavasti suuri vaikutus siihen, mihin tarkkaavaisuus suuntautuu ja miten informaatiota prosessoidaan (Pagendam, 2001). Kokeneet Internetin käyttäjät (käyttävät Internetiä päivittäin tai useina päivinä viikossa) ovat aivan erilaisessa asemassa verrattuna henkilöihin, joiden Internetin käyttö on satunnaisempaa. Kokeneille Internet-käyttäjille muodostuu kognitiivisia skeemoja Internet-sivujen rakenteista ja elementeistä. On aiheellista olettaa, että nämä skeemat toimivat myös mainosbannereiden kohdalla. Opittu käsitys mainosbannereiden hyödyttömyydestä informaation etsinnän kannalta ehkäisee mainosbannereiden aiheuttavien ärsykkeiden pääsyn tietoiseen ajatteluun. Toisin sanoen mainosbannerin huomiota herättävä olemus ei aktivoi käyttäjässä skeemaa, joka johtaisi informaation etsimiseen mainosbannereiden välityksellä, vaan ärsykkeet vaimennetaan tietoisesta ajattelusta (Norman, 1999).

4. Visuaalisten elementtien rooli navigoinnissa

Internet-sivut koostuvat erilaisista visuaalisista elementeistä. Mainosbannerit ovat osaltaan visuaalisia elementtejä, mutta eivät suinkaan ainoita. Kirjallisuus-

nessa paljon esillä olleet syyt mainosokeus-ilmiön taustalla ovat liittyneet suurelta osin mainosbannereiden visuaaliseen ulkoasuun. Myös muita syitä mainosokeuteen on esitetty, kuten erilaisia Internet-sivujen selaustyylejä. Tässä luvussa pyrin tuomaan esille yllä mainituista syistä poikkeavia asioita, jotka voivat aiheuttaa mainosbannereiden sekä muiden bannereiden huomiotta jättämisen. Tämä luku käsittelee visualisia elementtejä ja niiden vaikutusta mainosokeuteen.

“Way edge” viittaa käyttöliittymän objektiin, visuaaliseen rakennelmaan, jota käytetään ryhmittelemään sisältöä eri sektioihin. Internet-sivut koostuvat perinteisesti otsikosta, alaviitteestä, pääsisällöstä sekä mahdollisesti vasemmas- ta/oikeasta palstasta. Yesilada ja muut (2008) olettavat mainosokeuden johtuvan yhtä paljon “way edgeistä” kuin väreistä tai muista yksittäisistä mainos- bannereiden ja Internet-sivujen visuaalisista elementeistä. Yesiladan oletusta tukevia tutkimustuloksia onkin löydetty Manchesterin yliopistossa tehdystä katseenseurantatutkimuksesta (eye-tracking study). Katseenseurantatutkimuk- sessa mitattiin fiksaatiopisteitä Googlen, IMDb:n ja Manchesterin yliopiston Internet-sivuilla. Fiksaatiopisteellä tarkoitetaan pientä aluetta (muutama neliö- senttimetri), jonka ihminen pystyy kerrallaan havaitsemaan tarkannäön alueel- laan. Tutkimustulokset osoittavat, että suuri osa fiksaatioista keskittyy pääsi- sällön selaamiseen, seuraavina sektioina otsikko ja vasen palsta (ks. kuva 5).



Kuva 5. Fiksaatioiden prosentuaalinen osuus eri sektioilla. (Yesilada et al. 2008)

Koehenkilöiden fiksaatiot siis keskittyvät Internet-sivujen keskiosaan, jossa pääsisältö sijaitsee. Eri Internet-sivujen välillä on eroja selaustyyliessä; esimerkiksi Manchesterin yliopiston Internet-sivujen kohdalla ei ole suurta eroa pääsisällön ja vasemman sektorin välillä.

Koska mainosbannerit eivät yleensä sijaitse näillä korkean fiksaatiofrekvenssin alueilla, voidaan yhteenvetona todeta, että Yesilada ja muut (2008) osoittivat, että "way edgeillä" on suuri merkitys mainossokeuteen. Mainosbannereita ei siis tämän tutkimuksen perusteella huomata bannereiden sijainnin takia.

5. Yhteenveto

Monet tutkimukset tutkivat mainosbannereiden tehokkuutta ekonomisesta näkökulmasta, tutkimuksissa analysoidaan klikkausprosentteja sekä bannereiden ominaisuuksia, jotka vaikuttavat positiivisesti klikkausprosentteihin. Tässä tutkielmassa on tuotu esille erinäisiä syitä ja selityksiä mainossokeudelle. Tutkimustuloksia on analysoitu bannereiden ominaisuuksien ja sijainnin perusteella sekä huomioitu ihmisen informaation prosessoinnin erilaiset tavat.

Tutkimustulokset osoittavat, että mainosbannereita havaitaan Internet-sivuilla yllättävän paljon – tämän väitteen puolesta puhuu esimerkiksi Baylesin (2000) Ebayn ja Amazonin mainosbannereilla tehty tutkimus. Tutkimustuloksista nähdään, että mainosbannereiden tunnistusarvot ovat erittäin korkealla. Mielenkiintoista Baylesin tutkimuksessa on myös se, että bannereiden tilalla (staattinen tai animoitu) ei ollut merkitystä tunnistustehtävän kohdalla. Tämä voi hyvinkin johtua siitä, että koehenkilöt eivät prosessoineet bannereiden animaatio-ominaisuutta riittävän syvästi. Animaatio saattaa aiheuttaa tarkkaavaisuuden siirtymisen herkemmin kohti mainosbanneria, mutta animaatio ei tunnu vaikuttavan siihen, miten ja millä tasolla bannerin informaatiota prosessoitetaan.

Navigointityylillä on myös suuri vaikutus mainosbannereiden havaitsemiseen ja eritoten bannerin informaation muistamiseen ja tunnistamiseen. Tavottehakuiset Internetin selaajat tyypillisesti vaimentavat mainosbannereiden luomia ärsykeitä tietoisesta ajattelusta. Pagendam ja Schaumburg (2001) esittävät tutkimustuloksissaan ABL- ja GDSI-ryhmien eroja mainosbannereiden informaation muistista palauttamisessa ja tunnistamisessa. ABL-ryhmä palautti muistista sekä tunnsti mainosbannereita huomattavasti GDSI-ryhmää enemmän. Ainoastaan mainoskopioiden ja sloganeiden muistista palauttamisen kohdalla eroa ei ollut näiden kahden ryhmän välillä. Tämä luultavasti johtuu pienestä bannereiden informaation prosessointiajasta, joka ei ole riittävä teks-

tuaalisen informaation muistiin koodaamiseen myöskään ABL-ryhmän kohdalla.

Säännöllisille Internetin käyttäjille muodostuu kognitiivisia skeemoja, jotka ovat opittuja malleja Internet-sivujen rakenteista. Nämä skeemat auttavat Internet-selaajaa löytämään haluamansa tiedon selaamaltaan sivulta. Skeemat ohjaavat Internet-selaajaa suuntaamaan katseen Internet-sivun alueille, jossa oleellinen informaatio sijaitsee. Tällöin mainosbannerit jäävät huomaamatta, koska Internet-selaajan fiksaatiopisteet eivät koskaan kohtaa bannereita. Ja vaikka katseen fiksaatio mainosbanneriin osuisikin, se yleensä sivuutetaan lähes automaattisesti, koska on opittu että mainosbannerit harvoin pitävät sisälleen oleellista informaatiota.

Viiteluettelo

- Bayles, M. (2000). Just how 'blind' are we to advertising banners on the web? *Usability News*, 2, 2, (2000).
- Bayles, M. (2002). Designing online banner advertisements: should we animate? *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing our world, changing ourselves*. ACM, 2002, 363-366.
- Benway, J. P. & Lane, D. M. (1998). Banner Blindness: Web Searchers Often Miss "Obvious" Links. Retrieved from http://www.internettg.org/newsletter/dec98/banner_blindness.html.
- Burke, M., Hornof, A., Nilsen, E., and Gorman, N. (2005). High-cost banner blindness: ads increase perceived workload, hinder visual search, and are forgotten. *ACM Transactions on Computer-Human Interaction*, 12, 4, 423-445.
- Colbert, M., Lewis, M., Chattractichart, J., & Edwards, N. (2008). Persuading users to perform follow-on tasks: an initial case study. *Proceedings of the 15th European Conference on Cognitive Ergonomics: The Ergonomics of Cool Interaction*, ACM, Article No. 14.
- Eysenck, M. W. & Keane, M. T. (2005). *Cognitive psychology: A student's handbook*. East Sussex: Psychology Press.
- Haber, R. N. & Hershenson, M. (1973). *The psychology of visual perception*. New York: Hold, Rinehart and Winston Inc.
- Nielsen, J. (1997). Why advertising doesn't work on the web. Retrieved from <http://www.useit.com/alertbox/9709a.html>
- Norman, D. A. (1999). Commentary: Banner Blindness, Human Cognition and Web Design, Internetworking. Retrieved from <http://www.internettg.org/newsletter/mar99/commentary.html>

Yesilada, Y., Jay, C., Stevens, R., & Harper, S. (2008). Validating the use and role of visual elements of web pages in navigation with an eye-tracking study. *Proceeding of the 17th International Conference on World Wide Web*, 11-20.

Ohjaaja: Päivi Majaranta

Assosiaatiosäännöt ja louhittujen sääntöjoukkojen karsiminen

Sami Kiviharju

Tiivistelmä

Assosiaatiosäännöt ovat datasta louhittavia, muotoa $X \rightarrow Y$ olevia hahmoja. Ne ovat samalla todennäköisyyttä ilmaisevia lauseita, jotka kuvaavat tiettyjen alkioiden yhtäaikaista ilmenemistä datassa. Niitä on käytetty tietämyksen muodostamiseen esimerkiksi ostoskorianalyysissä sekä multimediatatan louhinnassa. Tässä tutkielmassa esitellään assosiaatiosäännöt ja muutamia niihin sovellettavia jälkikäsittelemenetelmiä. Lääketieteellisestä datasta louhittuja sääntöjoukkoja karsitaan suuntaavien sääntöjen ja ahneiden sääntöjoukon peittojen avulla ja menetelmien tuloksia vertaillaan keskenään. Aineistona käytetään iän aiheuttamaa kuulonalemää käsittelevää ARHI-aineistoa.

Avainsanat: tiedonlouhinta, assosiaatiosäännöt, jälkikäsitteily, suuntaavat säännöt

CR-luokitus: H.2.8

1 Johdanto

Tiedonlouhinta on tietokantojen, tekoälyn ja koneoppimisen leikkauspisteessä oleva tutkimusalue. Se voidaan määritellä kiinnostavien mallien, säännönmukaisuuksien tai poikkeamien etsimiseksi suuresta joukosta dataa. [7] Tässä tutkielmassa esitellään lyhyesti tiedonlouhinnan alaan kuuluvat assosiaatiosäännöt sekä muutamia niihin liittyviä jälkikäsittelemenetelmiä. Assosiaatiosääntöjä voidaan käyttää tiedonlouhintaan useilla sovellusalueilla esimerkiksi bioinformatiikassa, lääketieteessä tai muussa tieteellisen aineiston analysoinnissa [9].

Assosiaatiosääntö on tiettyjen *tietoalkoiden* (item) samanaikaista esiintymistä ennustava, todennäköisyyttä ilmaiseva lause [5]. Assosiaatiosääntöjä louhitaan tyypillisesti tietokannasta tai joukosta transaktiodataa. Esimerkiksi asiakkaiden ostoskoreja kuvaavasta datasta (taulukko 1) voidaan louhia assosiaatiosääntö

$$\{\text{leipä \& juusto}\} \rightarrow \{\text{maito}\} .$$

Taulukon perusteella nähdään, että asiakkaan ostaessa leipää ja juustoa on 66 % todennäköisyys sille, että asiakas ostaa myös maitoa. Säännön tarkkuus on täten

66 %. Koska sääntö pätee kahdessa tapauksessa viidestä, on sen tuki 40 %. Säännön tuen voi tulkita niiden tapausten osuudeksi, joille sääntö on voimassa.

Taulukko 1: Esimerkki transaktiodatasta.

Transaktiotunnus	Tietoalkiot
1	leipä, juusto
2	juusto, maito
3	leipä, juusto, maito
4	juusto, maito
5	leipä, juusto, maito

Tässä tutkielmassa luodaan katsaus assosiaatiosääntöihin sekä esitellään ja vertaillaan muutamaa niihin liittyvää jälkikäsitteilymenetelmää. Erityisesti ollaan kiinnostuneita niiden toimivuudesta iän aiheuttamaa kuulonalenemaa (age-related hearing impairment) käsittelevän ARHI-aineiston kohdalla. Tapauksia aineistossa on 4342 ja muuttujia 915. Valtaosa muuttujista, 770 kappaletta, on laadullisia ja liittyy tutkittujen henkilöiden genotyyppeihin. Näiden lisäksi aineistossa on muun muassa painoa, pituutta ja tupakointia kuvaavia kvantitatiivisia muuttujia. Fransen ja kumppanit [2] ovat esitelleet aineistoa tarkemmin. Assosiaatiosääntöjen avulla aineistosta pyritään löytämään kuulonalenemaa selittäviä tekijöitä. ARHI-aineiston moniulotteisuudesta johtuen louhitut sääntöjoukot saattavat kasvaa suuriksi. Niinpä niihin sovelletaan jälkikäsitteilymenetelmiä, joiden tarkoituksena on karsia louhittujen sääntöjoukkojen kokoja, säilyttäen kuitenkin samalla alkuperäisten sääntöjoukkojen kuvailuvoima. Louhittavat säännöt ovat muodoltaan rajattuja, koska säännön oikea puoli on kiinnitetty ilmaisemaan kuulonaleneman ilmenemistä.

Tutkielma jakautuu seuraaviin osiin: Ensin luodaan yleisluontoinen katsaus tiedonlouhintaan ja assosiaatiosääntöihin, tämän jälkeen luvussa 3 esitellään assosiaatiosäännöt sekä tarkasteltavat jälkikäsitteilymenetelmät. Luvussa 4 vertaillaan suppeasti kahta tapaa muodostaa tiivistelmä louhitusta assosiaatiosääntöjoukosta. Tarkastelun tavoitteena on selvittää, onko tuotettujen tiivistelmien välillä samankaltaisuutta.

2 Tiedonlouhinta, tietämyksen muodostaminen ja assosiaatiosäännöt

Tiedonlouhinta voidaan nähdä osana laajempaa kokonaisuutta tietämyksen muodostamisen osana. Myös käsitettä *tietämyksen muodostaminen tietokannoista* (knowledge discovery in databases, KDD) käytetään synonyyminä tietämyksen muodostamiselle. Käsitteiden tiedonlouhinta sekä tietämyksen muodostaminen käytössä esiintyy kuitenkin jonkin verran ristiriitaisuutta ja tiedonlouhinnalla voidaan tarkoittaa tietämyksen muodostamista tai tiedonlouhinta voidaan nähdä sen osana. Seuraavassa listassa on esitetty Hanin ja kumppaneiden [3] esittämä jaottelu, joka noudattaa jälkimmäistä käsitteellistä jakoa:

1. Datan siivous (poistetaan virheellinen ja ristiriitainen data)
2. Datan integrointi (yhdistetään eri lähteistä kerätty data)
3. Halutun datan valinta (kiinnostavien muuttujien valinta ja haku tietokannasta)
4. Datan muuntaminen (diskretointi tai muut muunnokset louhintaa varten)
5. Tiedonlouhinta (hahmojen ja mallien etsiminen datasta)
6. Louhittujen hahmojen ja mallien arviointi ja karsinta
7. Tietämyksen esittäminen (relevanttien hahmojen ja mallien esittäminen käyttäjälle).

Yllä olevat vaiheet voidaan yleisemmällä tasolla jakaa esikäsittelyyn, tiedonlouhintaan ja jälkikäsittelyyn siten, että kohdat 1-4 ovat esikäsittelyä ja kohdat 6-7 jälkikäsittelyä. Tässä tutkielmassa keskitytään louhittujen hahmojen arvioinnin ja karsinnan menetelmiin.

Esikäsittelyn tarkoituksena on halutun datan tai sen osajoukon muuntaminen käytettävää louhintamenetelmää varten. Varsinaisia tiedonlouhinnan menetelmiä on useita: esimerkiksi assosiaatiosäännöt, päätöspuinduktio sekä klusterointi. Niiden tarkoituksena on hahmojen ja mallien muodostaminen annetun datan perusteella. Jälkikäsittelyn tarkoituksena on louhittujen hahmojen ryhmittely, karsiminen ja arviointi. Myös tulosten visualisointi ja esittäminen käyttäjälle sopivassa muodossa on osa jälkikäsittelyä.

Tietämyksen muodostamisen kaikki vaiheet ovat tärkeitä ja riippuvaisia toisistaan. Myöskään jälkikäsittelyn ja tulosten visualisoinnin merkitystä ei sovi väheksyä, sillä kaikki aiemmissa vaiheissa tehty työ on turhaa, mikäli käyttäjä ei ym-

määrä louhinnan tuloksia. Assosiaatiosääntöjen jälkikäsitteily voidaan jakaa neljään osaan: karsiminen, yhteenvetojen muodostaminen, ryhmittely ja visualisointi [1]. Karsimisen tavoitteena on poistaa epäkiinnostavia ja päällekkäisiä sääntöjä. Yhteenvetojen muodostamisen päämäärä on sääntöjen ryhmittely ja muotoilu yleisempien käsitteiden avulla. Kolmannessa vaiheessa toisiinsa liittyvät säännöt pyritään jaottelemaan ryhmiin ja visualisointivaiheessa aiempien vaiheiden tulokset pyritään esittämään informatiivisesti ja graafisesti.

Tässä tutkielmassa esitellään menetelmiä, joissa painopiste on tietyn selitettävän tekijän esiintymiseen vaikuttavien syiden etsimisessä. Toisin sanoen kiinnostuksen kohteena ovat säännöt, joissa tietyn muuttujan tietty arvo on kiinnitetty säännön oikeaksi puoleksi ja säännön vasen puoli (säännön ehdot) saa vaihdella vapaasti. Kyseinen tarkastelutapa soveltuu esimerkiksi silloin, kun ollaan kiinnostuneita tiettyä sairautta ennustavista tekijöistä tai tietyn ilmiön syistä. Erityistapauksena tietyn ilmiön esiintymisen tarkastelusta voidaan mainita kausaalisten suhteiden louhiminen assosiaatiosääntöjen avulla [8]. Keskittymällä tiettyyn kiinnostavaan arvoon saadaan kriteeri ja kiintopiste jälkikäsitteilyä varten, joten haluttuun vaikutukseen liittyvät tekijät voidaan valita ja ryhmitellä jatkotutkimusta varten. Ryhmittelyn avulla voidaan esimerkiksi selvittää geneettiset tekijät, joiden vaikutus korostuu tiettyjen ympäristötekijöiden vallitessa, toisin sanoen liittyy yhteen positiivisesti korreloivat ympäristö- ja geneettiset tekijät. Visualisoinnin avulla tuodaan esiin aiemmissa vaiheissa louhitut tekijät.

Jälkikäsitteilyn ja varsinaisen assosiaatiosääntöjen louhimisen välinen raja saattaa välillä olla varsin häilyvä. Osa jälkikäsitteilymenetelmistä voidaan toteuttaa osana louhintaa. Tällöin pyritään louhimaan pelkästään kiinnostavia sääntöjä niin, että erillistä jälkikäsitteilyvaihetta ei tarvita tai siinä suoritettava työ vähenee. Jälkikäsitteilymenetelmien esittäminen louhintamenetelmistä irrallaan on perusteltua, sillä ne eivät ole riippuvaisia tietystä louhinta-algoritmista ja näin niitä voidaan kuvata ja tarkastella helpommin erikseen.

3 Assosiaatiosäännöt ja niiden louhinta

Seuraava määritelmät pohjautuvat Liun ja kumppaneiden [6] sekä Toivosen ja kumppaneiden [10] esittämiin määritelmiin ja merkintöihin.

Assosiaatiosääntö kuvaa tiettyjen tietoalkioiden samanaikaista esiintymistä. Tietoalkiot ovat attribuutti-arvo -pareja. Otetaan esimerkkinä attribuutti *alentunut kuulo* ja sille arvot *kyllä* ja *ei*. Näistä voidaan muodostaa kaksi tietoalkiota: *alentunut kuulo=kyllä* ja *alentunut kuulo=ei*. Tapausten voidaan ajatella olevan tietoalkiojoukkoja.

Olkoon $I = \{i_1, \dots, i_j\}$ joukko tietoalkioita ja D joukko tapauksia. Kaikilla

tapauksilla c , pätee $c \subset I$. *Assosiaatiosääntö* on muotoa $X \rightarrow Y$ oleva implikaatio, jossa $X \subset I, Y \subset I$ ja $X \cap Y = \emptyset$. Implikaatio-merkin vasemmalla puolella oleva osa on säännön vasen puoli (left hand side, LHS) ja oikealla puolella oleva säännön oikea puoli (right hand side, RHS). Säännön vasenta puolta voidaan myös kutsua ehdoksi (antecedent) ja oikeaa puolta seuraukseksi (consequent). Säännön edeltäjäsäännöksi kutsutaan sääntöä, jonka ehtojen joukko on tarkasteltavana olevan säännön ehtojen osajoukko.

Assosiaatiosääntöjä käsittelevässä kirjallisuudessa käytetään usein lyhennysmerkintää $XZ \rightarrow Y$, merkinnän $X \cup Z \rightarrow Y$ sijaan. Lyhennysmerkintä voidaan laajentaa mielivaltaiseen määrään tietoalkioita, sekä ehdossa että seurauksessa.

Tarkastellaan tietoalkiojoukkoa X . Tietoalkiojoukon X kattamien tapausten joukko on $m(X) = \{c \in D \mid X \subseteq c\}$ ja sen kattamien tapausten lukumäärää merkitään $|m(X)|$.

Sääntö $X \rightarrow Y$ pätee joukossa D *minimituella* (minimum support) σ ja *minimitarkkuudella* (minimum confidence, minimum strength) γ , jos

$$|m(XY)| \geq \sigma n,$$

kun n on joukon D tapausten lukumäärä, ja

$$\frac{|m(XY)|}{|m(X)|} \geq \gamma.$$

Edellisen perusteella säännön $X \rightarrow Y$ tuki s voidaan laskea kaavalla:

$$s = \frac{\text{säännön } X \rightarrow Y \text{ kattamien tapausten lkm}}{\text{kaikkien tapausten lkm}} = \frac{|m(XY)|}{|m(\emptyset)|},$$

kun $m(\emptyset)$ kattaa kaikki tapaukset. Sääntö $X \rightarrow Y$ kattaa tapauksen c , jos $X \subseteq c$ ja $Y \subseteq c$. Tuki kertoo säännön kattamien tapausten osuuden kaikista tapauksista.

Säännön tarkkuus t voidaan laskea kaavalla

$$t = \frac{\text{säännön } X \rightarrow Y \text{ kattamien tapausten lkm}}{\text{säännön vasemman puolen kattamien tapausten lkm}} = \frac{|m(XY)|}{|m(X)|}.$$

Säännön tarkkuus kertoo todennäköisyyden sille, että säännön vasemman puolen ollessa voimassa myös oikea puoli pätee.

Tuen ja tarkkuuden lisäksi säännöille voidaan laskea muitakin mittalukuja (interestingness measures), kuten noste (lift), nostovoima (leverage) ja kattavuus (coverage) [11], joiden avulla sääntöjen kiinnostavuutta ja merkittävyyttä voidaan arvioida. Mittalukuja käytetään myös assosiaatiosääntöjen louhinnan ohjaamiseen sekä louhittujen sääntöjoukkojen karsimiseen.

3.1 Assosiaatiosääntöjen louhiminen

Assosiaatiosääntöjen louhinta-algoritmit ovat oma laaja tutkimusalueensa. Tässä tutkielmassa tyydytään esittämään vain yksinkertaisen ja tehottoman algoritmin pääpiirteet. Louhinnassa käytetään karsintaperusteena tyypillisesti minimimitukea σ ja minimitarkkuutta γ . Myös muita mittalukuja voidaan käyttää, eikä kaikissa menetelmissä käytetä tukea tai tarkkuutta. Esimerkiksi myöhemmin esiteltävä suuntaavien sääntöjen menetelmä ei käytä karsimiseen minimitarkkuutta.

Sääntöjä voidaan louhia seuraavasti: Muodostetaan kaikki mahdolliset ehto- ja seurausalkiojoukot. Käydään läpi kaikki ehto ja seuraus -yhdistelmät ja karsitaan pois säännöt, jotka eivät täytä annettuja ehtoja (tuki, tarkkuus ym.).

Usein säännön oikea puoli on tietty, kiinnitetty tietoalkio. Säännön seuraus kiinnitetään, koska ollaan kiinnostuneita sen ilmenemiseen vaikuttavista ehdoista. Näin on tehty myös tässä tutkielmassa, jossa kaikki louhitut säännöt ovat muotoa $X \rightarrow \text{kuulonolenema}=\text{kyllä}$.

Assosiaatiosääntö-menetelmän vahvuus on sen kyvyssä löytää kaikki annetut ehdot täyttävät säännöt. Tällä on kuitenkin käänköpuolensa, sillä sääntöjä löytyy usein huomattavia määriä. Päädytään helposti tilanteeseen, jossa tuotettuja sääntöjä on tutkittava tiedonlouhinnan menetelmin, jotta ne voidaan esittää käyttäjälle. Hahmojen osajoukon esittäminen ei ole tyydyttävä ratkaisu, koska pelkkä osajoukko hahmoista antaa vaillinaisen kuvan louhituista hahmoista ja kohdealueesta. [6, 1]

3.2 Sääntöjoukon peitto

Louhitut sääntöjoukot ovat usein suuria ja sisältävät paljon päällekkäisiä sääntöjä. Esimerkkinä päällekkäisistä säännöistä voidaan pitää sääntöjä $X \rightarrow Y$, $XZ \rightarrow Y$ ja $XZW \rightarrow Y$. Ensimmäinen sääntö on näistä selvästi kattavin, sillä muut säännöt sisältävät vähintään alkiojoukon X kaikki alkiot. Jälkimmäiset säännöt antavat lisäehtoja Y :n tapahtumiselle, kun tiedetään, että X tapahtuu, joten näiden sääntöjen tarkkuus on todennäköisesti suurempi kuin ensimmäisen säännön. Lisäksi ne saattavat kattaa vain osajoukon lyhimmän säännön kattamista tapauksista. Sovellusalueesta riippuen päällekkäiset säännöt voivat olla kiinnostavia tai ne voivat vaikeuttaa sääntöjoukon tulkintaa kasvattamalla sääntöjoukon kokoa.

Sääntöjoukkoja voidaan karsia sääntöjen ja sääntöjoukkojen peittoja laskemalla. Karsittu sääntöjoukko voidaan muodostaa esimerkiksi maksimoimalla katettujen tapausten määrä ja minimoimalla sääntöjoukon koko. Tällä tavoin muodostettu karsittu sääntöjoukko sisältää usein vain vähän sääntöjä verrattuna alkupeiräiseen [10]. Haittapuolena on, että muodostettu sääntöjoukko antaa vaillinaisen kuvan kohdealueesta, koska karsimisen myötä menetetään osa assosiaatiosääntö-

jen vahvuudesta löytää kaikki tuki- ja tarkkuusehdot täyttävät säännöt [6]. Tämä kuitenkin riippuu tavasta, jolla karsittu joukko on muodostettu.

Seuraavaksi esitellään kaksi tapaa muodostaa sääntöjoukkojen peittoja. Molemmat ovat Toivosen ja kumppaneiden [10] esittämiä. *Rakenteellinen sääntöjoukon peitto* (structural rule cover) lasketaan käyttämällä apuna pelkästään sääntöjen rakennetta eli niissä esiintyviä tietoalkioita [10]. *Ahne sääntöjoukon peitto* taas muodostetaan ottaen huomioon sääntöjen kattamat tapaukset.

3.2.1 Sääntöjoukon rakenteellinen peitto

Sääntöjoukkojen rakenteellinen peitto perustuu aiemmin kohdassa 3.2 esitettyyn havaintoon. Olkoot X , Y ja Z tietoalkiojoukkoja. Tällöin

$$m(XYZ) \subseteq m(XZ).$$

Siis sääntö $XY \rightarrow Z$ kattaa osan säännön $X \rightarrow Z$ kattamista tapauksista. Jos sääntöä $XY \rightarrow Z$ muistuttavat säännöt poistetaan sääntöjoukosta, kattaa jäljelle jäävä sääntöjoukko silti samat tapaukset kuin alkuperäinen sääntöjoukko.

Olkoot Δ ja Γ sääntöjoukkoja ja $\Delta \subseteq \Gamma$. Sääntöjoukko Δ on Γ :n rakenteellinen sääntöjoukon peitto, jos kaikille säännöille $(X \rightarrow Y) \in \Delta$ pätee, ettei ole olemassa sellaista sääntöä $(X' \rightarrow Y) \in \Gamma$ siten, että $X' \subset X$.

Näin muodostettu rakenteellinen sääntöjoukon peitto sisältää siis alkuperäisen sääntöjoukon lyhimmät säännöt.

3.2.2 Ahne sääntöjoukon peitto

Sääntöjoukon peiton laskennassa voidaan käyttää hyväksi tietoa säännön kattamista tapauksista. Näin voidaan muodostaa karsittuja sääntöjoukkoja, jotka paremmin säilyttävät alkuperäisen sääntöjoukon rakenteen ja esitysvoiman. Rakenteella tarkoitetaan tässä yhteydessä sääntöjoukon sääntöjen ehtojen lukumäärään jakautumaa. Rakenteelliseen sääntöjoukon peittoon valitaan lyhimmät säännöt, toisin sanoen, ne joissa on vähiten ehtoja. Ahneisiin peittoihin taas voi ainakin teoriassa päätyä myös pidempiä sääntöjä. Kuten edellinen sääntöjoukkojen peiton määrittely, myös seuraava on Toivosen ja kumppaneiden [10] esittämä.

Olkoon O alkuperäinen sääntöjoukko ja G sellainen ahne sääntöjoukon peitto, että $m(O) = m(G)$ eli sääntöjoukot kattavat täysin samat tapaukset. Ahne sääntöjoukon peitto on sääntöjoukko, joka kattaa täsmälleen samat tapaukset kuin alkuperäinen sääntöjoukko. Tyypillisesti ahne sääntöjoukko sisältää murto-osan alkuperäisen sääntöjoukon säännöistä. Taulukon 2 ensimmäisessä sarakkeessa on ARHI-datan pohjalta louhittujen sääntöjoukkojen kokoja. Toisessa sarakkeessa on kustakin joukosta muodostetun ahneen peittävän sääntöjoukon koko. Viimeisessä

sarakkeessa on sääntöjen kattama osuus kiinnostavista tapauksista. Ahne sääntöjoukko on muodostettu siten, että se kattaa täsmälleen samat tapaukset kuin alkuperäinen sääntöjoukko. Taulukon kaikki säännöt ovat muotoa $X \rightarrow a$, kun a on aina sama tietoalkio.

Taulukko 2: Eri ehdoilla louhittujen sääntöjoukkojen kokoja.

Alkuperäinen sääntöjoukko	Ahne sääntöjoukko	Katettuja tapauksia
1484	44	96%
694	6	40%
66	45	53%
762	128	96%
6	6	7%

Taulukosta 2 nähdään, että alkuperäisen sääntöjoukon kattamat tapaukset voidaan kattaa huomattavasti pienemmällä ahneesti muodostetulla sääntöjoukolla.

Algoritmi 1 esittää Toivosen ja kumppaneiden [10] esittämän ahneita kattavia sääntöjoukkoja laskevan algoritmin pääpiirteet.

Algoritmi 1 Ahne sääntöjoukon peitto.

Syöte: sääntöjoukko *alkuperäinenJoukko*

- 1: *ahneJoukko* $\leftarrow \emptyset$
 - 2: **while** $m(\textit{ahneJoukko}) \neq m(\textit{alkuperäinenJoukko})$ **do**
 - 3: **for all** *sääntö* \in *alkuperäinenJoukko* **do**
 - 4: Valitse alkuperäisen joukon säännöistä se joka lisää eniten tapauksia ahneen joukon kattamiin tapauksiin.
 - 5: Lisää valittu sääntö ahneeseen joukkoon.
 - 6: **end for**
 - 7: **end while**
 - 8: **return** *ahneJoukko*
-

Algoritmi 1 toimii ahneella tavalla ja valitsee sääntöjoukon peittoon aina sen säännön, joka kullakin kierroksella lisää eniten katettuja tapauksia ahneeseen sääntöjoukkoon. Tätä jatketaan, kunnes ahne joukko kattaa alkuperäisen sääntöjoukon tapaukset. Rivillä 4 oleva vertailu suoritetaan poistamalla tutkittavan säännön kattamista tapauksista ahneen joukon kattamat säännöt ja laskemalla erotusjoukon tapausten määrä. Sääntö, jonka erotusjoukko on suurin, valitaan lisäystä varten.

Ahneita sääntöjoukon peittoja muodostava algoritmi löytää yhden lähes optimaalisen ahneen sääntöjoukon [10]. Optimaalisin ahne sääntöjoukon peitto määritellään joukoksi, johon sisältyy vähiten sääntöjä. Löydetty sääntöjoukko ei välttämättä ole optimaalinen, sillä useiden yhtä hyvien sääntöjen joukosta valitaan mielivaltaisesti ja toteutusriippuvaisesti yksi. Tällaiset valintatilanteet voivat aiheuttaa ei-optimaalisen sääntöjoukon muodostamisen.

3.3 Suuntaavat säännöt

Yhteenvedoja voidaan peittävien sääntöjoukkojen lisäksi muodostaa suuntaavien sääntöjen avulla. Suuntaavat säännöt ovat tilastollisesti merkitseviä sääntöjä, jotka määrittävät muiden, ei-suuntaavien, sääntöjen suunnan. Ne antavat tiivistetyn näkökulman louhitun sääntöjoukon rakenteeseen ja poimivat datassa olevien assosiaatioiden yleiset suuntaviivat. Suuntaavien sääntöjen joukko on usein myös varsin pieni verrattuna kaikkien sääntöjen joukkoon. [6]

Liun ja kumppaneiden [6] esittämässä suuntaavien sääntöjen määrittelyssä käytetään sääntöjen karsimiseen minimitukea ja χ^2 -testiä. χ^2 -testiä käytetään karsinnan lisäksi sääntöjen suunnan laskemiseen. Tarkoituksena on karsia säännöt, jotka eivät kuvaa tilastollisesti merkitsevää riippuvuutta säännön ehdon ja seurauksen välillä koko datan kontekstissa. Myös säännöt, jotka eivät ole edeltäjäsääntöjensä määrittämien tapausten suhteen tilastollisesti merkitseviä, karsitaan pois. Esitetty menetelmä ei käytä tarkkuutta karsimiseen, vaikkakin se on helppo lisätä karsintaperusteeksi [6].

3.3.1 Suuntaavat säännöt

Säännön korrelaatio saadaan selville laskemalla, ovatko säännön ehto ja seuraus riippuvuussuhteessa keskenään. Kaikki säännöt ovat joko *korreloivia* tai *riippumattomia*. Nämä käsitteet määritellään assosiaatiosäännöille seuraavasti:

Määritelmä 1. Olkoon σ vaadittu minimituki ja m tilastollinen merkitsevyystaso. Sääntö $X \rightarrow Y$ on (σ, m) -korreloiva (lyhyemmin korreloiva), jos seuraavat ehdot täyttyvät:

1. Säännön tuki ylittää minimituen σ .
2. Säännön χ^2 -arvo suhteessa koko dataan ylittää χ^2 -arvon annetulla merkitsevyystasolla m .

Määritelmä 2. Olkoon σ vaadittu minimituki ja m tilastollinen merkitsevyystaso. Sääntö $X \rightarrow Y$ on (σ, m) ei-korreloiva (lyhyemmin ei-korreloiva tai itsenäinen), jos seuraavat ehdot täyttyvät:

1. Säännön tuki ylittää minimituen σ .
2. Säännön χ^2 arvo suhteessa koko dataan ei ylitä χ^2 -arvoa annetulla merkitsevyystasolla m .

Pelkkä tieto säännön korrelaatiosta ei vielä paljasta korrelaation suuntaa. Näin ollen riippuvuussuhteen olemassaolosta ei voida päätellä sen luonnetta eli sitä, onko korrelaatio positiivinen vai negatiivinen. Riippumatta siitä, onko sääntö korreloiva vai itsenäinen, voidaan säännölle määritellä korrelaatio-ominaisuuden avulla säännön suunta seuraavasti:

Määritelmä 3. Olkoot f_o χ^2 -testin laskemisessa käytettävä havaittu frekvenssi ja f odotettu frekvenssi.

Positiivinen korrelaatio: Jos säännön vasen ja oikea puoli korreloivat ja $f_o/f > 1$, sanotaan että sääntö korreloi positiivisesti. Käytetään arvoa 1 positiivisen korrelaation ilmaisemiseen. Säännön suunta on tällöin 1.

Negatiivinen korrelaatio: Jos säännön vasen ja oikea puoli korreloivat ja $f_o/f < 1$, sanotaan että sääntö korreloi negatiivisesti. Käytetään arvoa -1 negatiivisen korrelaation ilmaisemiseen. Säännön suunta on tällöin -1.

Ei korrelaatiota: Jos säännön vasen ja oikea puoli eivät korreloi keskenään, sanotaan että sääntö ei korreloi. Arvoa 0 käytetään riippumattomuuden ilmaisemiseen. Säännön suunta on tällöin 0.

Suuntaavan säännön suunnan täytyy erota sen *odotettujen suuntien* joukosta, joka määräytyy edeltäjäsääntöjen suuntien avulla. Edeltäjäsäännöistä otetaan kuitenkin huomioon vain osa.

Määritelmä 4. Säännön r odotettujen suuntien joukko määritellään seuraavasti:

1. Jos säännöllä on vain yksi ehto, säännön odotettujen suuntien joukko on $\{0\}$, eli säännön ehto ja seuraus oletetaan toisistaan riippumattomiksi aiemman tiedon puuttuessa.
2. Olkoon säännöllä r ehtoja k , $k > 1$, kappaletta. Sääntö on tällöin muotoa: $a_1, a_2, \dots, a_k \rightarrow y$. Säännön r voidaan myös nähdä koostuvan kahdesta ehtosaltaan pienemmästä säännöstä r_1 ja r_{rest} seuraavasti:

$$r_1: a_i \rightarrow y$$

$$r_{rest}: a_1, a_2, \dots, a_j \rightarrow y,$$

kun $\{a_1, a_2, \dots, a_j\} = \{a_1, a_2, \dots, a_k\} - \{a_i\}$. Käymällä läpi kaikki k kappaletta säännöstä r muodostettavissa olevaa r_1, r_{rest} -kombinaatiota saadaan

säännön r odotettujen suuntien joukko $\{E_i\}, i = 1, \dots, k$. Kunkin kombinaation r_1, r_{rest} odotettu suunta lasketaan seuraavalla tavalla, kun $x.dir$ on säännön x suunta:

jos $r_1.dir = r_{rest}.dir = 1$ **niin** $E_i = 1$.
jos ($r_1.dir = 1$ ja $r_{rest}.dir = 0$) **tai** ($r_1.dir = 0$ ja $r_{rest}.dir = 1$) **niin** $E_i = 1$.
jos $r_1.dir = r_{rest}.dir = 0$ **niin** $E_i = 0$.
muulloin $E_i =$ tuntematon

Suuntaava sääntö on assosiaatiosääntö, joka on χ^2 -testin avulla todettu tilastollisesti merkitseväksi suhteessa koko aineistoon sekä sen omiin edeltäjäsääntöihin nähden. Lisäksi sen suunta ei ole sama kuin minkään sen edeltäjäsäännön. Kaikkien suuntaavien sääntöjen suunta on yksi, joten ne korreloivat positiivisesti.

Määritelmä 5. Sääntö r on suuntaava, jos se täyttää seuraavat ehdot:

1. Säännön r suunta on 1 ($r.dir = 1$), joten säännön vasen ja oikea puoli korreloivat positiivisesti.
2. Säännön suunta ei kuulu *odotettujen suuntien* joukkoon.

Suuntaavilla säännöillä on tyypillisesti vain vähän ehtoja. Liu ja kumppanit [6] louhivat sääntöjä kolmestakymmenestä eri kohdealueisiin liittyvästä aineistosta ja laskivat suuntaavien sääntöjen ehtojen lukumäärän keskiarvoksi 1.91 ehtoa.

4 Vertailu

Assosiaatiosääntöjen jälkikäsitteilyä varten on olemassa useita erilaisia menetelmiä. Tämä johtaa luonnollisesti tarpeeseen vertailla, luokitella ja ryhmitellä eri menetelmiä. Mielekkäintä vertailu on silloin, kun se on sidottu tiettyyn kohdealueeseen ja sille ominaiseen dataan, sillä nämä vaikuttavat käytettävissä oleviin menetelmiin ja niiden tuloksiin. Esimerkkinä voidaan mainita kohdealueen tietämyksen ilmaiseminen assosiaatiosääntösapluunien (rule template) muodossa. Näin voidaan ilmaista kohdealueeseen liittyvää tietämystä ja ennako-oletuksia esimerkiksi muodossa "aina kun a, niin b". Ennako-odotukset rikkovia sääntöjä voidaan pitää mielenkiintoisina ja ne tuodaan käyttäjän nähtäville. [1] Aineistosta riippuen on myös mahdollista muodostaa taksonomioita eli luokitella datan alkioita yleiskäsitteiden avulla. Niiden avulla voidaan muodostaa sääntöjä, joissa käytetään yleiskäsitteitä, kuten maitotuotteet, tarkoittamaan tiettyjä datassa ilmeneviä alkioita. [1]

Tässä tutkielmassa rajoitutaan vertailemaan aineita sääntöjoukon peittoja sekä suuntaavia sääntöjä. Menetelmiä vertaillaan ARHI-aineistosta louhittujen sääntöjoukkojen suhteen. Vertailun tarkoituksena on selvittää, onko aineiden sääntöjoukkojen ja suuntaavien sääntöjen avulla tuotetuissa yhteenvedoissa samoja sääntöjä. Yhteenvedoja vertaillaan pelkästään yhteisten sääntöjen perusteella, joten yhteenvedojen mielekkyyttä tai kiinnostavuutta ei arvioida kohdealueen kannalta.

4.1 Esivalmistelut

Suuntaavia sääntöjä louhiva algoritmi olettaa sääntöjoukon olevan täydellinen. Täydellinen sääntöjoukko sisältää kaikkien sääntöjen kaikki edeltäjäsäännöt. Sääntö $a_1, a_2, \dots, a_k \rightarrow y$ muodostuu edeltäjäsäännöistä, jotka ovat muotoa:

$$\begin{aligned} &\{a_1, a_2, \dots, a_k\} - \{a_i\} \rightarrow y \text{ ja} \\ &\{a_i\} \rightarrow y, \end{aligned}$$

kun $1 \leq i \leq k$.

Louhitut sääntöjoukot on siis ennen menetelmien soveltamista täydennetty sisältämään kaikki edeltäjäsäännöt.

4.2 Vertailun toteutus

Vertailua varten toteutettiin Toivosen ja kumppaneiden [10] esittämä aineen sääntöjoukon peiton laskeva algoritmi sekä Liun ja kumppaneiden [6] esittämä suuntaavia sääntöjä etsivä algoritmi. Molemmat algoritmit ajettiin syötteenään samat täydennetyt sääntöjoukot. Kuten ennakoita voitiin päätellä, aineet sääntöjoukon peitot muodostuivat säännöistä, joilla on vain yksi ehto. Tämä oli odotettavissa, sillä kaikkien yhden ehdon sisältävien sääntöjen joukko kattaa kaikki tapaukset, jotka näiden sääntöjen yhdistelmätkin kattavat. Ahneisiin peittoihin olisi voinut päätyä myös sääntöjä, joissa on enemmän kuin yksi ehto, mutta näin ei kuitenkaan vertailutilanteessa tapahtunut. Pällekkäisiä sääntöjä aineilla joukoilla ja suuntaavilla säännöillä oli kahdessa tapauksessa neljästä.

Ahne algoritmi käsittelee tasapuolisesti kaikkia sääntöjoukon sääntöjä, kun taas suuntaavia sääntöjä louhittaessa osa säännöistä karsitaan pois tarkastelusta, koska ne eivät täytä tilastolliselle merkitsevyydelle asetettua ehtoa. Algoritmit saadaan toimimaan vertailukelpoisessa tilanteessa, jos sääntöjä etsitään ahneesti vain karsinnan läpäisseiden sääntöjen joukosta. Näin onkin tehty tässä tarkastelussa, joten suuntaavia sääntöjä ja aineen joukon sääntöjä etsitään samojen tilastollisesti merkitsevien sääntöjen joukosta.

4.3 Tulokset

Taulukossa 3 on listattu sääntöjoukkojen alkuperäiset ja täydennetyt koot. Näiden lisäksi on esitetty täydennetyssä sääntöjoukossa olevien tilastollisesti merkitsevien eli karsimattomien sääntöjen määrä. Karsimattomien sääntöjen joukosta on louhittu suunnatut säännöt sekä muodostettu ahne sääntöjoukon peitto.

Taulukko 3: Eri ehdoilla louhittujen sääntöjoukkojen kokoja

Alkuperäinen koko	Täydennetty koko	Karsimattomia sääntöjä
762	1852	972
66	154	78
23	23	17
6	6	4

Taulukossa 4 on esitetty louhittujen sääntöjoukkojen koot ja molemmissa joukoissa olevien sääntöjen lukumäärä. Kuten taulukon kahta viimeistä saraketta tarkastelemalla voidaan todeta, ovat ahneeseen sääntöjoukkoon valikoituneet säännöt samalla myös suuntaavia sääntöjä.

Taulukko 4: Eri ehdoilla louhittujen sääntöjoukkojen yhteisten sääntöjen lukumäärä

Suuntaavia sääntöjä	Ahneen joukon koko	Yhteisiä sääntöjä
193	17	17
37	27	27
17	16	16
4	4	4

Tehdyn suppean tarkastelun perusteella havaitaan, että osa ARHI-aineistosta muodostetuista suuntaavista säännöistä on päällekkäisiä niiden kattamien tapaus-ten suhteen. Tämän vuoksi kaikki suuntaavat säännöt eivät tule mukaan ahneisiin sääntöjoukon peittoihin.

5 Yhteenveto

Tässä tutkielmassa on vertailtu suuntaavia sääntöjä ja ahneita sääntöjoukon peittoja niiden tuottamien tiivistettyjen sääntöjoukkojen kautta. Molemmat menetelmät löysivät ja nostivat esille samoja hahmoja datasta. Näiden lisäksi on esitelty assosiaatiosäännöt yleensä, sääntöjoukon rakenteellisen peiton käsite ja muodostaminen, sekä lyhyt katsaus sääntöjen louhimiseen. Näiden lisäksi tutkielman ulkopuolelle on jätetty suuri määrä muita jälkikäsittelemenetelmiä. Edistyneempää materiaalia liittyen sääntöjen louhintaan ovat kehittäneet Webb ja Zhang [11], kun taas Silverstein ja kumppanit [8] ovat soveltaneet sääntöjä kausaalisuhteiden tarkasteluun. Han ja kumppanit [4] esittävät vuonna 2007 julkaistussa artikkelissaan katsauksen alan silloisiin saavutuksiin ja suuntauksiin.

Assosiaatiosäännöille löytyy useita sovellusalueita ja tietomäärien kasvaessa jälkikäsitteilyn merkitys vain korostuu. Jälkikäsitteily puoltaa paikkaansa kiinnostavana tutkimusalueena jo pelkästään siksi, että louhimisprosessin tulokset pysyisivät käytännöllisen kokoisina, niin ettei niitä uudestaan tarvitsisi käyttää datana tiedonlouhintaprosessissa.

Viitteet

- [1] Bart Baesens, Stijn Viaene, and Jan Vanthienen. Post-processing of association rules. In *Proceedings of the Workshop on Post-Processing in Machine Learning and Data Mining: Interpretation, Visualization, Integration, and Related Topics within The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2–8. ACM, 2000.
- [2] Erik Fransen, Vedat Topsakal, Jan Jaap Hendrickx, Lut Van Laer, Jeroen Huyghe, Els Van Eycken, Nele Lemkens, Samuli Hannula, Elina Maki-Torkko, Mona Jensen, Kelly Demeester, Anke Tropitzsch, Amanda Bonaconsa, Manuela Mazzoli, Angeles Espeso, Katia Verbruggen, Joke Huyghe, Patrick Huyghen, Sylvia Kunst, Minna Manninen, Amalia Diaz-Lacava, Michael Steffens, Thomas Wienker, Ilmari Pyykkö, Cor Cremers, Hannie Kremer, Ingeborg Dhooge, Dafydd Stephens, Eva Orzan, Markus Pfister, Michael Bille, Agnete Parving, Martti Sorri, Paul Van de Heyning, and Guy Van Camp. Occupational noise, smoking, and a high body mass index are risk factors for age-related hearing impairment and moderate alcohol consumption is protective: a european population-based multicenter study. *Journal of the Association for Research in Otolaryngology*, 9(3):264–276, 2008.
- [3] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2006.

- [4] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.
- [5] D. J. Hand, Heikki Mannila, and Padhraic Smyth. *Principles of Data Mining*. The MIT Press, 2001.
- [6] Bing Liu, Wynne Hsu, and Yiming Ma. Pruning and summarizing the discovered associations. In *KDD '99: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 125–134. ACM, 1999. ISBN 1-58113-143-7.
- [7] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Efficient algorithms for discovering association rules. In *KDD Workshop*, pages 181–192. AAAI Press, 1994.
- [8] Craig Silverstein, Sergey Brin, Rajeev Motwani, and Jeff Ullman. Scalable techniques for mining causal structures. *Data Mining and Knowledge Discovery*, 4(2):163–192, 2000.
- [9] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2006.
- [10] Hannu Toivonen, M. Klemettinen, P. Ronkainen, K. Hätönen, and Heikki Mannila. Pruning and grouping discovered association rules. In *ECML-95 Workshop on Statistics, Machine Learning, and Discovery in Databases*, pages 47–52, 1995.
- [11] Geoffrey I. Webb and Songmao Zhang. K-optimal rule discovery. *Data Mining and Knowledge Discovery*, 10(1):39–79, 2005.

Ohjaaja: Kati Iltanen

GATE Toolkit esimerkkinä nykyaikaisesta luonnollisen kielen käsittelyohjelmasta

Salla Kuisma

Tiivistelmä.

Tutkielmassa tutustutaan luonnollisen kielen käsittelyn nykytilaan ja GATE-ohjelmistoalustaan. Tutkielma paneutuu kieliteknologian ja erityisesti luonnollisen kielen käsittelyn tämänhetkisiin kehityssuuntiin käyttäen esimerkkinä GATE:a nykyaikaisesta luonnollisen kielen käsittelyohjelmistosta. GATE on monipuolinen, paljon dokumentoitu, avoimen lähdekoodin kielen käsittelyohjelmisto, jonka taustalla on pitkä kehityshistoria, ja se toimii siis hyvin esimerkkinä. Kerron, miten GATE vertautuu muihin samankaltaisiin ohjelmistoihin ja alan haasteisiin myös yleisemmällä tasolla. Luon tarkemman katsauksen GATE:n toteutukseen, toiminnallisuuteen ja käyttömahdollisuuksiin esitelläkseni sen avulla luonnollisen kielen käsittelytyökaluja ja niiden käyttömahdollisuuksia.

Avainsanat ja -sanonnat: Kieliteknologia, luonnollisen kielen käsittely, GATE Toolkit.

CR-luokat: A.2

1. Johdanto

Kieliteknologia on poikkitieteellinen ala, jolla on kosketuspintaa esimerkiksi informaatiotieteiden, kielitieteen, matematiikan ja tekniikan kanssa. Kieliteknologia voidaan karkealla tasolla jakaa luonnollisen kielen käsittelyyn (LKK), tietokone-lingvistiikkaan ja puheteknologiaan. LKK pyrkii kehittämään ja mahdollistamaan erilaisia luonnollista kieltä käyttäviä sovelluksia. Tekstistä voidaan vaikkapa tietokoneella ajettujen tunnistusalgoritmien ja tietorakenteiden avulla tunnistaa ja merkitä – eli annotoida – sanaluokkia. Tällä tavoin käsiteltyä tietoa on helpompi käyttää käytännön sovelluksissa. Annotaatioilla tarkoitetaan tekstiin liitettyjä lisätietomerkintöjä (tag), jotka sisältävät pääasiassa kielellistä tietoa. Tietokone-lingvistiikka kehittää erilaisia tapoja tunnistaa kielen rakenteita. Puheteknologia taas kehittää puheentunnistus- ja puheentuottamistekniikoita.

Cole [1997] lisää näihin osa-alueisiin vielä mm. dokumenttien käsittelemisen, tekstin tuottamisen, koneellisen kääntämisen, kielellisen tiedon varastoimisen ja kieliresurssit (korpuukset eli tekstikokoelmat, ontologiat eli käsitteistöt ja sanastot). Dokumenttien käsittelyllä tarkoitetaan tekstin käsittelyä ja tiedonpoimintaa (information extraction eli IE). Tiedonpoiminta tarkoittaa, että tietokone tunnistaa haluttuja ominaisuuksia tekstistä ja usein kerää näiden tietojen avulla

erillisen semanttisen kerroksen tekstikokoelmaan. Esimerkiksi kirjaston asiasanatietokanta ja semanttinen verkko voidaan toteuttaa tällaisilla metodeilla.

Kieliteknologisia sovelluksia on nykyään yhä enemmän, sillä elektronisen kielimateriaalin määrä on kasvanut viime aikoina räjähdysmäisesti. Käytännön sovelluksia voidaan käyttää kaikkialla, missä ollaan tekemisissä luonnollisen kielen kanssa: tekstinkäsittelyohjelmien oikeinkirjoitus, kännyköiden ennakoiva tekstinsyöttö, hakukoneiden hakusanatuki ja automaattinen kielenkääntäminen. Tulevan kehityksen kannalta on ratkaisevaa, miten tällaisia tekstimääriä pystytään hallitsemaan. Kieliteknologia on tässä merkittävässä asemassa, sillä sen avulla voidaan käsitellä ja luokitella suuriakin tekstimääriä. Myös erilaisten kielellisten käyttöliittymien kehittäminen on avainasemassa.

Tämä tutkielma keskittyy erityisesti luonnollisen kielen käsittelyyn ja käyttää esimerkkinä GATE Toolkit -ohjelmistoalustaa (General Architecture for Text Engineering). GATE toimii hyvänä esimerkkinä nykyaikaisesta LKK-ohjelmistosta, sillä se on riittävän laaja, se on saavuttanut toimivuutensa vuoksi vankan aseman tiedeyhteisössä ja se täyttää useita modernille LKK-ohjelmistolle asetettuja kriteereitä (ks. kohta 2). GATE-projekti lähti käyntiin vuonna 1995, ja GATE:n ensimmäinen version julkaistiin 1996 [Cunningham et al., 1997b]. GATE toimii kielitieteen, kieliteknologian ja informaationtekniikan aloilla toimivien tutkijoiden ja ohjelmistokehittäjien työkaluna. Se koostuu useista erillisistä ohjelmistokomponenteista, jotka käsittelevät, varastoivat ja visualisoivat tekstejä. GATE tarjoaa kielen käsittelyssä tarvittavat perustyökäluut, joiden avulla teksteistä voidaan tunnistaa ja merkitä esimerkiksi sanoja, lauseita, sanaluokkia ja erisnimiä. [Cunningham et al., 1997b.] Tällainen käsittely mahdollistaa tekstien jatkokäytön monenlaisissa sovelluksissa, vaikkapa lingvistisissä korpuksissa tai tiedonhakuohjelmissa.

Tutkielman tarkoituksena on luoda katsaus GATE-projektin nykytilaan ja sen asemaan suhteessa alan muuhun kehitykseen. Ensin tarkastellaan LKK-alan nykytilaa ja nykyisiä avoimia LKK-ohjelmistoja. Sitten paneudutaan tarkemmin GATE:n suunnitteluperiaatteisiin ja toteutusratkaisuihin. Myös GATE:n eri käyttömahdollisuuksia ohjelmistokehitysalustana, kieliteknologiatyökalun ja opetusvälineenä esitellään. Tämän jälkeen arvioidaan GATE:n heikkouksia ja vahvuuksia sekä verrataan kokonaisuuden toimivuutta samankaltaisiin ohjelmistoihin ja luodaan katsaus ohjelmiston mahdollisiin tuleviin kehityssuuntiin.

2. Kielen käsittelyn nykytila

Viime vuosikymmenien aikana tekninen kehitys on mahdollistanut luonnollisen kielen käsittelyn aivan uudella tasolla. Kuten Curran [2003] toteaa, luonnollisen kielen käsittelyyn liittyviä työkaluja, ohjelmistoja ja tekniikoita hyödynne-

tään yhä useammassa käytännön sovelluksessa. Curran selittää kasvua myös alan laajenemisella yhä laajemmille kielialueille ja yhä uusiin sovellusalueisiin.

Toisaalta myös elektronisen tekstin ja kielellisen online-materiaalin (sanastot, ontologiat ja asiasanastot) määrä on kasvanut huimasti [Cunningham et al., 2000a]. On siis vaan luonnollista, että niiden käsittelyyn tarvitaan entistä tehokkaampia ja laajempia käsittelytyökaluja. Cunningham ja muut toteavat myös, että tarvittavat laitteistot ovat kehittyneet siinä määrin, että myös käsittelyohjelmistoista on voitu kehittää entistä tarkempia ja tehokkaampia.

Tässä luvussa esittelen ensin hieman luonnollisen kielen käsittelyohjelmistojen tämän hetkistä tilaa. Kerron hieman ohjelmistojen jaottelusta, tämänhetkistä ongelmista ja mahdollisista ratkaisuvaihtoehdoista. Lopuksi esittelen muutaman avoimen LKK-ohjelmiston.

2.1. Kielen käsittelyjärjestelmien jaottelu

Cunningham ja muut [2000a] jaottelevat LKK-järjestelmät kolmeen osaan: täydentävät järjestelmät, joissa tuotettu lisäinformaatio lisätään alkuperäisen tekstin sisälle; viitteelliset järjestelmät, joissa lisäinformaation on tallennettu erillisiin, alkuperäisen tekstin sisältöön viittaaviin tiedostoihin; käsitteelliset järjestelmät, joissa alkuperäinen teksti säilytetään vain osana tietorakennetta, sillä uutta informaatiota ei ole tarpeen sitoa alkuperäiseen tekstiin. SGML ja TEI (www.tei-c.org) ovat esimerkkejä ensimmäisen ryhmän menetelmistä. TIPSTER-formaatti [Cunningham et al., 1997a] noudattaa toisen ryhmän määrittelyä, ja ALEP [Schmid et al., 1996] on esimerkki kolmannesta toteutustavasta [Cunningham et al., 2000a].

Cunningham ja muut [2000a] pitävät viitteellisiä järjestelmiä toimivimpina monestakin syystä. Ensinnäkin täydentävien järjestelmien tekstejä voi käyttää vain lukemiseen, kun taas viitteellisten järjestelmien lisätietotiedostot ovat luku- ja kirjoituskelpoisia. Tämä kuormittaa järjestelmää huomattavan paljon, sillä täydennetyt tekstit pitää tallentaa ja lukea kokonaan uudelleen joka kerran, kun tietoja halutaan muuttaa. Lisäksi tekstit pitää myös tallentaa alkuperäisessä muodossa normalisointivaiheessa. Cunningham ja muut lisäävät vielä, että viitteellisissä järjestelmissä on myös helpompi kontrolloida monitasoisia käyttöoikeuksia. Viitteellinen tietorakenne sopii hyvin kaksiulotteisen tiedon mallintamiseen, sillä se on jo lähtökohtaisesti kaksikulotteinen, ja lisäksi se mahdollistaa laajat haut [Cunningham et al., 2000a; Ide, 2000].

2.2. Alan ongelmakohtia ja ratkaisuehdotuksia

Ala kärsii kuitenkin vielä monista ongelmista. Curran [2003] myöntää, että osasyy luonnollisen kielen käsittelyn ongelmille ovat sille asetetut korkeat vaatimukset. Hänen mukaansa ihmiset asettavat turhankin suuria odotuksia sovel-

lusten tehokkuudelle, nopeudelle ja erityisesti äärimmäiselle tarkkuudelle. Myös Leidner [2003] toteaa, että ihmiset ovat äärimmäisen vaativia käyttäjiä ja täydellisen tarkkuuden toteuttamiseksi on vaikea luoda riittävää suorituskykyä.

Cunningham ja muut [2000a] huomauttavat, että kieliteknologia on niin monitieteellinen ala, että se aiheuttaa väistämättömästi ongelmia. On hankalaa kerätä sellaista tutkijaryhmää, joka koostuisi kaikkien alojen asiantuntijoista. He muistuttavat myös, ettei alalla ole vielä olemassa yhtä yhteisesti hyväksyttyä kielellistä mallia tai teoriaa, jota kaikki voisivat käyttää. Toisaalta nämä molemmat ongelmat yhdessä johtavat siihen, että pienet ryhmät kehittelevät vain omia sovelluksiaan itsenäisesti ilman laajempaa yhteistä määränpäättä. Leidner [2003] ja Curran [2003] lisäävät tähän, että tutkijaryhmät keskittyvät usein vain kielen käsittelyn teoreettisempiin puoliin jättäen tehokkuuden lisäämisen kaupallisille sovelluksille.

Leidner [2003], Curran [2003] ja Cunningham ja muut [2000a] toteavat kaikki ohjelmistojen ja niiden osien uudelleenkäyttöprosentin olevan liian pieni, joten on ajauduttu tekemään paljon turhaa päällekkäistä työtä. Leidnerin [2003] mukaan tulisi ennemmin keskittyä toimivien ohjelmointirajapintojen (API) luomiseen kuin viimeistelemättömien ohjelmien kehittämiseen. Curran [2003] taas keskittyisi generatiiviseen ohjelmointiin ja harkitusti jaoteltujen, helposti muunneltavien, matalan tason komponenttien kehittämiseen. Komponenttirakenne ratkaisisi sekä monialaisuuden että uudelleen käytettävyyden ongelman: matalan tason komponentteja on helppo käyttää uudelleen ja jaottelu takaa sen, että eri alojen ammattilaiset voivat keskittyä van oman alansa tehtäväkomponentteihin. Curran jakaisi komponenttien osa-alueet seuraavasti:

- tiedostojen käsitleminen
- tekstin käsitleminen
- sanaston käsitleminen
- ominaisuuksien eritleminen
- tietorakenteet ja algoritmit
- koneoppimismetodit
- kielelliset resurssit (sanastot, sanakirjat, ontologiat jne.).

On myös olennaista, että annotaatiot ovat kaikkien komponenttien välillä yhdenmukaiset, ettei niitä tarvitse ladata erikseen jokaisessa komponentissa. Yhdenmukaisuuden lisäksi myös eri tallennusmuotojen tukeminen on tärkeää [Curran, 2003; Leidner, 2003].

Leidner [2003] nostaa lisäksi esille viansietokyvyn, multimodaalisuuden ja monikielisyyden mukanaan tuomat ongelmat. Hän esittää, että kielen analysointiohjelmien tarvitsema harjoitusmateriaalin saatavuus on liian vähäistä,

vaikkakin Curran [2003] ja Cunningham ja muut [2000a] toteavat sen määrän olevan nopeassa kasvussa.

2.3. Kielen käsittelyohjelmistoja

Tässä luvussa tutustutaan tarkemmin muutamaan avoimen lähdekoodin LKK-ohjelmistoon: Outilexiin, NLTK:hon, XAIRA:an ja XDOC:in. Alalta löytyy toki useita kaupallisiakin sovelluksia, kuten WordSmith, MonoConc, XeLDA ja Intex, Nvivo [Durian, 2002]. Avoin lähdekoodi takaa kuitenkin ohjelmistojen vapaamman hyödyntämisen erilaisissa tutkimus- ja kehitysprojekteissa.

Vuonna 2006 ilmestynyt ranskalainen Outilex-ohjelmisto [Blanc and Constant, 2006] on uudehko tulokas alalla, eikä sen käytönkokemuksiin ja -tarpeisiin pohjautuva kehitys siis ole vielä kovin vankalla pohjalla. Outilex on avoin lingvistinen ohjelmistoalusta, joka sisältää LKK-työkaluja, sanastoja ja kielioppi-pohjia tekstin käsittelyä varten. Outilexin erikoisuuksia ovat kielivarantojen hallintatyökalu ja asiasanahakemisto [Blanc and Constant, 2006]. Toisaalta Outilex tukee vain muutamia tallennusmuotoja, eikä ollenkaan tiedonpoimintaa.

Englannissa kehitetyn NLTK:n toiminnallisuuteen ja kehitykseen ovat huomattavasti vaikuttaneet sen opetukselliset tavoitteet. NLTK:n lähtöajatuksena on ollut tuottaa opetusohjelmisto vieraan kielen ja kieliteknologian opettamiseen. NLTK on tähännynt seuraaviin kolmeen käyttötarkoitukseen: eritasoisten harjoitustehtävien toteuttaminen, havainnollistaminen opetuksen tukena sekä tutkimus- ja kehitysprojektit [Bird and Loper, 2004]. Se sisältääkin valmiiden ohjelmaliikoiden lisäksi tutoriaaleja, harjoitustehtäviä ja valmista opetusaineistoa [Loper and Bird, 2002].

NLTK:n suunnitteluperiaatteisiin on kuulunut alusta pitäen mm. helppokäyttöisyys, johdonmukaisuus, hyvä dokumentaatio ja yksinkertaisuus. Koko ohjelma on lisäksi toteutettu helposti opittavalla ja loogisella Python-kielillä. Näiden suunnitteluperiaatteiden ja ratkaisujen takia varjoon jäävät ohjelmiston tehokkuus, laajuus ja rakenteellisen tason selkeys. Myös tietorakenteiden laajeneminen ja monimutkaistuminen vaatimusten kasvaessa on aiheuttanut ohjelmistolle suuria ongelmia [Bird, 2006].

British National Corpus (BNC), joka on yksi laajimmista englanninkielisistä korpuksista, on kehittänyt tarpeitaan varten XAIRA-ohjelmiston (XML Aware Indexing and Retrieval Architecture). XAIRA on sitä edeltäneen SARA:n XML-versio, eikä se ole edeltäjänsä tavoin enää sidoksissa vain BNC:hen [Xaira]. XAIRA koostuu kolmesta osasta: Indeksointityökalusta, palvelinohjelmistosta ja käyttäjäohjelmistosta [BNC]. Tällaisen asiakas-palvelin -mallin etuna on, ettei loppukäyttäjältä vaadita tehokasta tietokonetta, mutta toisaalta tarvittavan ohjelmiston käyttäminen vaatii aina lisäpanosta.

Saksankieliseltä alueelta löytyy XDOC-sovellus, joka hyödyntää XML:ää ja sen rinnalla toimivia XSLT-formalismia ja työkaluja. XDOC:n päämääriin kuuluu ensinnäkin, että ohjelmiston tulee olla niin joustava ja vikasetoinen, että sen avulla voidaan käsitellä todellisia elektronisia tekstejä kaikilta aloilta (esim. biokemia ja matematiikka). Toiseksi kaikkia XDOC:n työkaluja tulee voida käyttää itsenäisesti, ja kolmanneksi ohjelmiston tulee olla niin helppokäyttöinen, että kaikki – ei vain asiantuntijat – osaavat käyttää sitä. XDOC hyväksyy sekä syötteenä että tulosteena vain XML-tiedostoja eikä tue muita, mikä hankaloittaa huomattavasti tekstikokoelman laajentamista, sillä kaikki tekstit tulee ensin muuntaa itse XML-muotoon. [Rösner and Kunze, 2002.]

3. GATE:n esittely

Tässä luvussa tutustutaan tarkemmin GATE:n toiminnallisuuteen ja sen kehityshistoriaan. GATE:n kehityshistoria toimii esimerkkinä LKK-ohjelmiston mahdollisesta kehityskaaresta. Vertaan GATE:a muihin samankaltaisiin ohjelmistoihin, ja kerron, miten se on vastannut alan yleisiin haasteisiin.

GATE haluaa tukea erityisesti kielten ja kieliteknologian alan tutkijoita ja kehittäjiä, informaatioteknologian kehitystä, IE-alan kehitystä sekä ohjelmiston käytettävyyttä. GATE on laajentanut toiminnallisuuttaan mahdollistaen myös tiedonpoiminnan, tekstin tuottamisen ja vieraan kielen opiskelun tukemisen. [Cunningham et al., 2000a.]

Kuten johdantoluvussa jo todettiin, GATE:n ensimmäinen versio julkaistiin 1996 [Cunningham et al., 1997b]. Viimeisin GATE:sta julkaistu versio on 5.1, mutta ehkä suurimmat parannukset tulivat jo GATE:n toisessa versiossa [Bontcheva et al., 2004; Cunningham et al., 2009]. Tämän jälkeen uudet julkaisut ovat sisältäneet enemmän laajennuksia ja korjauksia [Cunningham et al., 2009]. Esimerkiksi versiossa 4.0 esiteltiin merkittävä ANNIC-kokoelma, jonka avulla GATE:a voidaan käyttää myös tiedonpoiminnassa [Cunningham et al., 2009].

3.1. GATE:n suunnitteluperiaatteet

GATE:n kehitystä on ohjannut alusta asti tavoite siitä, ettei GATE:n käyttäjien, pääasiassa tutkijoiden, tarvitsisi kuormittaa itseään tehtävillä, jotka kuuluvat ohjelmistolle. Tällaisia ovat mm. tiedon varastointi, tiedon visualisointi, datan ja komponenttien lataaminen, prosessien hallinta, prosessitehtävien jakaminen ja tiedonvälittäminen ohjelmiston sisällä. GATE tarkoituksena on merkittävästi tehostaa NLP:n tarvitsemien toimintojen suoristusta ja välttää päällekkäisiä toimintoja (programming overheads) sekä välttää eri käyttöjärjestelmien ja formaattien tuomia ongelmia. [Cunninghamin et al., 2000b.]

GATE:n kehittäjät ovat pyrkineet vastaamaan näihin tavoitteisiin jakamalla GATE:n ominaisuudet kolmeen osaan: GDM (GATE Document Management), CREOLE (a Collection of REusable Objects for Language Engineering) ja GGI (GATE Graphical User Interface). GDM kattaa kaikki ohjelmiston tiedon varastointiin ja hallintaan liittyvät toiminnot. CREOLE on kielen käsittelykomponenttien kokoelma. GGI eli graafinen käyttöliittymä taas vastaa kaikesta tiedon visualisoinnista. Kaikki osiot on toteutettu oliopohjaisella ohjelmoinnilla, jotta komponenttien yhteensopivuus- ja uudelleenkäyttöprosentti voisi olla mahdollisimman korkea. Lisäksi koodiin on mahdollista päästä käsiksi kaikilla tasoilla, mikä helpottaa järjestelmän muunneltavuutta. [Cunningham et al., 2000b.]

Komponentteja on pyritty luomaan ja kehittämään neljän peruseriaatteen pohjalta. Ensinnäkin komponentit on koetettu jaotella mahdollisimman matalan tason tehtäviin, jotta eri osioita voitaisiin tarvittaessa käyttää mahdollisimman monessa eri tehtävässä. Toiseksi komponenttien suoritusta mitataan automaattisesti, jotta niiden toiminnasta voitaisiin saada mahdollisimman tarkka kuva sekä ohjelmistokehityksen että itse käytön tukemiseksi. Kolmanneksi komponenttien helppo integrointi, yhteensopivuus ja soveltuvuus pyritään takaamaan käyttämällä paljon käytettyjä ja laajalti tuettuja standardeja. Neljäntenä GATE:n tavoitteena on pyrkiä tuottamaan mahdollisimman kattava kokoelma kielen käsittelyn perustyökaluja, joita on tarvittaessa helppo laajentaa, muuntaa ja myös korvata. [Cunningham et al., 2002.]

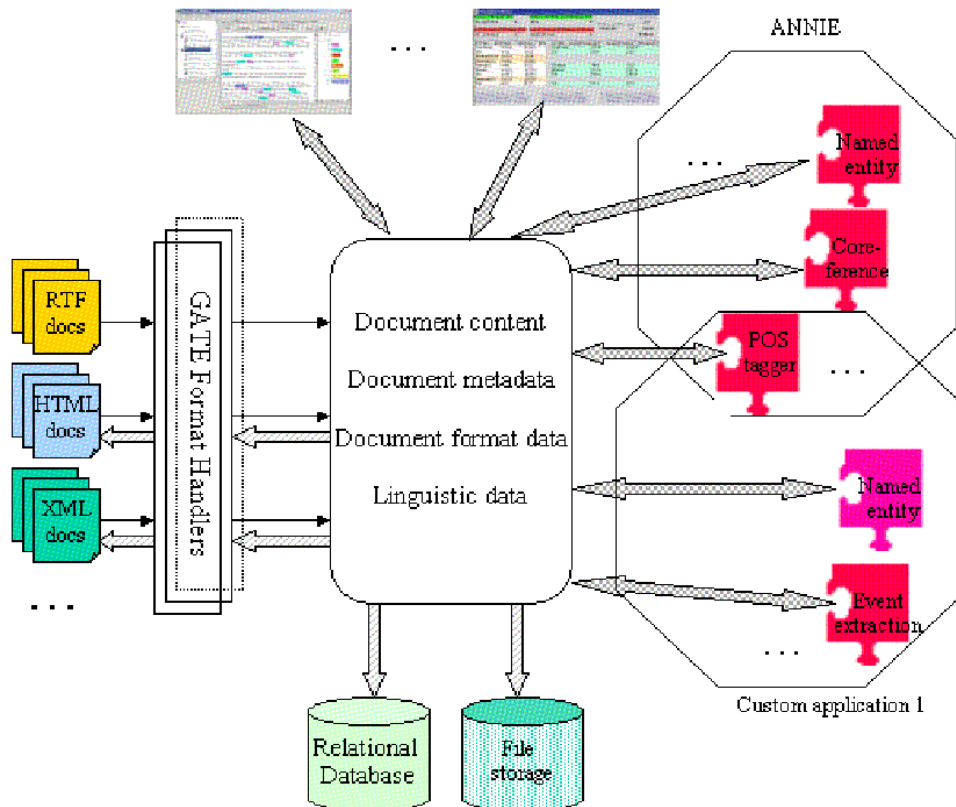
GATE toimii myös erinomaisesti kehitysalustana, sillä se on alunperinkin suunniteltu avustamaan ohjelmistokehitystä. Tässä suhteessa on merkittävää, että GATE:n ohjelmistorunkoon on helppo rakentaa lisäosia. GATE:ssa on myös sisäänrakennettu debuggaustoiminto, ja tiedon visualisoiminen on helppoa joka vaiheessa. GATE:n jo tarjoamat valmiit komponentit auttavat muiden osien kehityksessä, sillä ne toimivat itsenäisesti ja niitä voi hyödyntää kehityksen alla olevissa komponenteissa. [Cunningham et al., 2002.]

3.2. GATE:n ohjelmistoarkkitehtuuri ja tekninen toteutus

Tässä kohdassa paneudutaan tarkemmin GATE:n tekniseen toteutukseen – ohjelmistoarkkitehtuuriin ja toimintojen pohjalla oleviin tekniikoihin. GATE-arkkitehtuuri tarjoaa ohjelmistorungon ja siinä itsenäisesti toimivia komponentteja, joita kuka tahansa voi halutessaan muokata ja laajentaa tarpeisiinsa sopiviksi [Cunningham et al., 2002]. Kaikki komponentit ovat Java-luokkia ja näin toteuttavat olio-ohjelmoinnin periaatteita. Kaikkien ominaisuuksien tiedot on tallennettu XML-tiedostoihin. [Cunningham et al., 2009.]

Kuva 1 esittelee GATE:n ohjelmistorakennetta. Se on jaettu kuvan 1 keskellä näkyviin kieliresursseihin (mm. sanastot, korpuukset, kielioppipohja, ontologi-

at), oikealla oleviin prosessiresursseihin (mm. algoritmit, jäsentäjät, generaattorit, ngram-mallintajat) ja ylhäällä näkyviin visualisointiresursseihin (käyttöliittymä). Lisäksi kuvassa 1 näkyvät syöteenkäsittely vasemmalla ja tietovarannot alhaalla. Tämänkaltainen jaottelu mahdollistaa sen, että jokaista osa-aluetta voivat olla kehittämässä alan asiantuntijat (esim. lingvisti, ohjelmistokehittäjä, käyttäjä), eikä heiltä vaadita tarpeetonta tietoa muista aloista [Cunningham et al., 2002].



Kuva 1. GATE:n ohjelmistoarssitehtuuri [Bontcheva et al., 2002b].

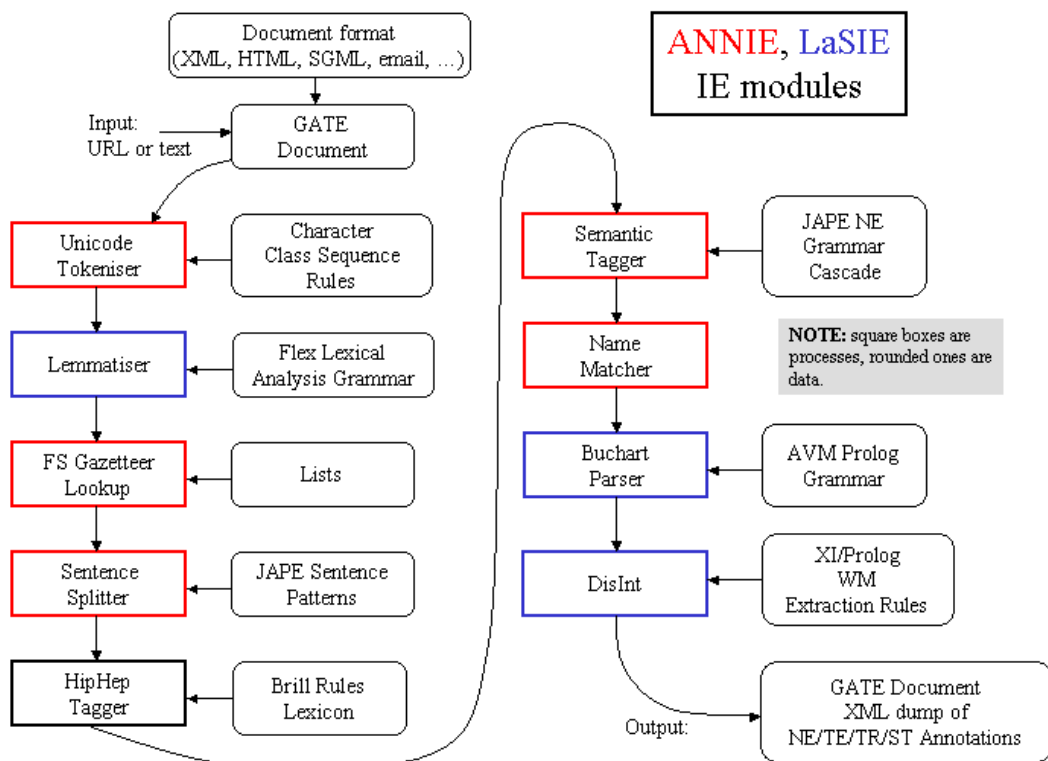
Curran [2003] huomauttaa, että GATE eroaa positiivisesti muista ohjelmistoista komponenttipohjansa ansiosta. GATE noudattaakin Curranin hyvälle LKK-ohjelmistolle asettamia vaatimuksia (ks. kohta 2.2).

Edellä mainittujen peruskomponenttien lisäksi GATE sisältää erilaisia arviointityökaluja tulosten vertailuun ja prosessien seurantaan. Cunninghamin ja muiden [2000a] mukaan moduulien välisen kommunikoinnin, yhdenmukaistamisen, arvioinnin ja kehittämisen tukeminen on merkittävä osa uudelleenkäytettävien komponenttien kehityksen mahdollistamista (ks. kohta 2.2).

GATE:n erikoisuuksiin kuuluu, että se tukee hyvin monia eri dataformaatteja (esim. xml, rtf, html, sgml, email, plain text) verrattuna esimerkiksi XDOC-tai Outilex-ohjelmistoihin (ks. kohta 2.3). GATE muuntaa kaikki syötteet TIPSTER-annotaatiomuotoon ja käyttää viitteellistä annotaatioiden tallennusjär-

jestelmää (ks. kohta 2.1). Tämä yhdenmukainen tietorakenne takaa komponenttien yhteensopivuuden. GATE tukee myös Unicodea, mikä mahdollistaa tietojen monikielisyyden.

CREOLE ja ANNIE sisältävät mm. seuraavia kielen käsittelykomponentteja, joista osa sisältyy molempiin ja jotka luetellaan tässä viitteellisessä suoritusjärjestyksessä [Cunningham et al., 2002; Cunningham et al., 2009]. Selaaja (tokeniser) jakaa tekstin merkityksellisiin merkkeihin ja merkkijonoihin (esim. numerot, välimerkit, erityyppiset sanat, symbolit). Sen tehtävä on helpottaa muitten työkalujen taakkaa. Lauseentunnistaja (sentence splitter) jakaa tekstin lauseisiin. Sanaluokantunnistaja (POS tagger) merkitsee sanojen sanaluokat. Gazetteer sisältää listoja, jotka koostuvat mm. maantieteellisistä erisnimistä, organisaatioista, ajanilmauksista tai lyhenteistä ja joiden mukaan sanoja luokitellaan. Semanttinen tunnistus (semantic tagger) etsii tekstistä semanttisesti merkityksellisiä, käsitteitä sääntöjä tai kaavoja. Orthomatcher tunnistaa tekstien tai tekstin osien suhteita toisiinsa. Coreferencer taas etsii itsenäisten kokonaisuksien (entiteetti) suhteita toisiin. Kuvassa 2 näkyy ANNIE:n koko rakenne mukaan lukien edellä esitellyt komponentit. Kuvassa mainittu LaSIE (Large Scale Information Extraction) on ANNIE:n edeltäjä.



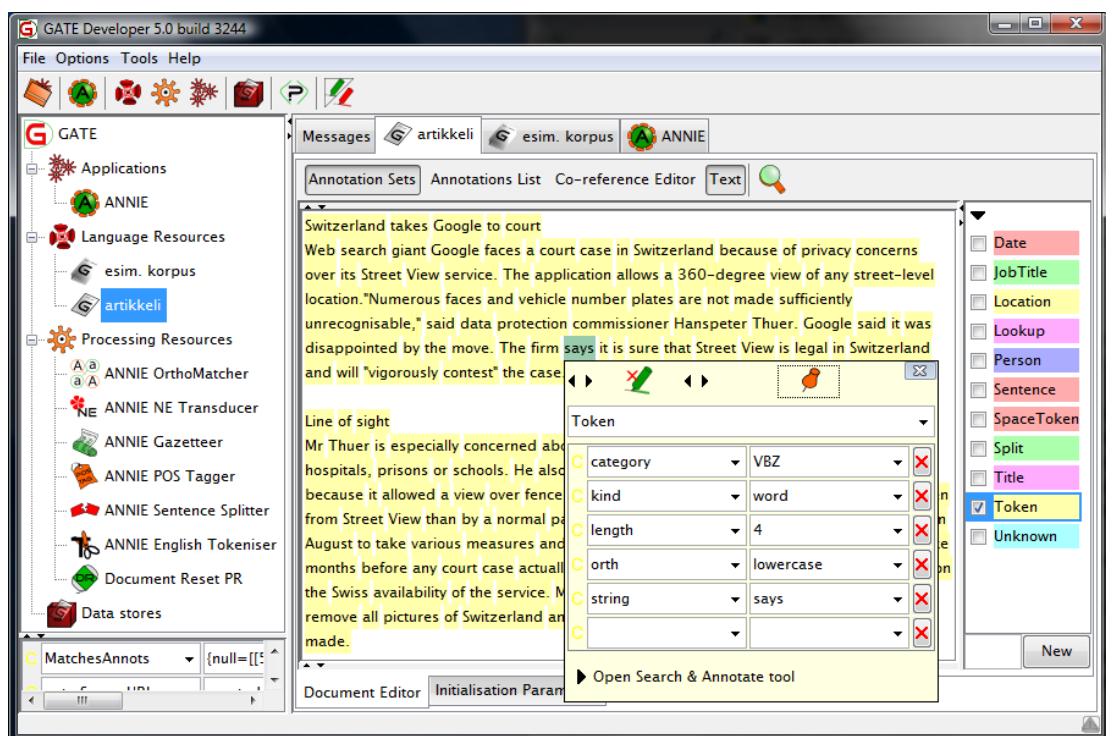
Kuva 2. ANNIE:n komponentit [Cunningham et al., 2009].

Suurin osa em. komponenteista on toteutettu JAPE:n (Java Annotations Pattern Engine) avulla. JAPE-kielioppipohja koostuu kaavakokoelmista. Jokainen

kaava sisältää vasemmanpuoleisen (LHS) ja oikeanpuoleisen (RHS) osan. LHS on säännöllinen lauseke ja RHS on kohdetta kuvaava annotaatio. Näitä kaavoja voidaan tunnistaa kolmella eri tavalla: määrittämällä tietty merkkijono, määrittämällä merkkijonon ominaisuus (esim. numero) tai määrittämällä kohteen tyyppi gazetteer-listasta. [Cunningham et al., 2000c; Cunningham et al., 2009.] Aswani ja muut [2005] tuovat esiin sen huomionarvoisen seikan, että GATE:en kuuluva korpusten indeksointiin ja hakuihin suunniteltu ANNIC (ANNotations-In Context) toimii myös JAPE:n kehitysalustana.

3.3. GATE:n käyttöesimerkki

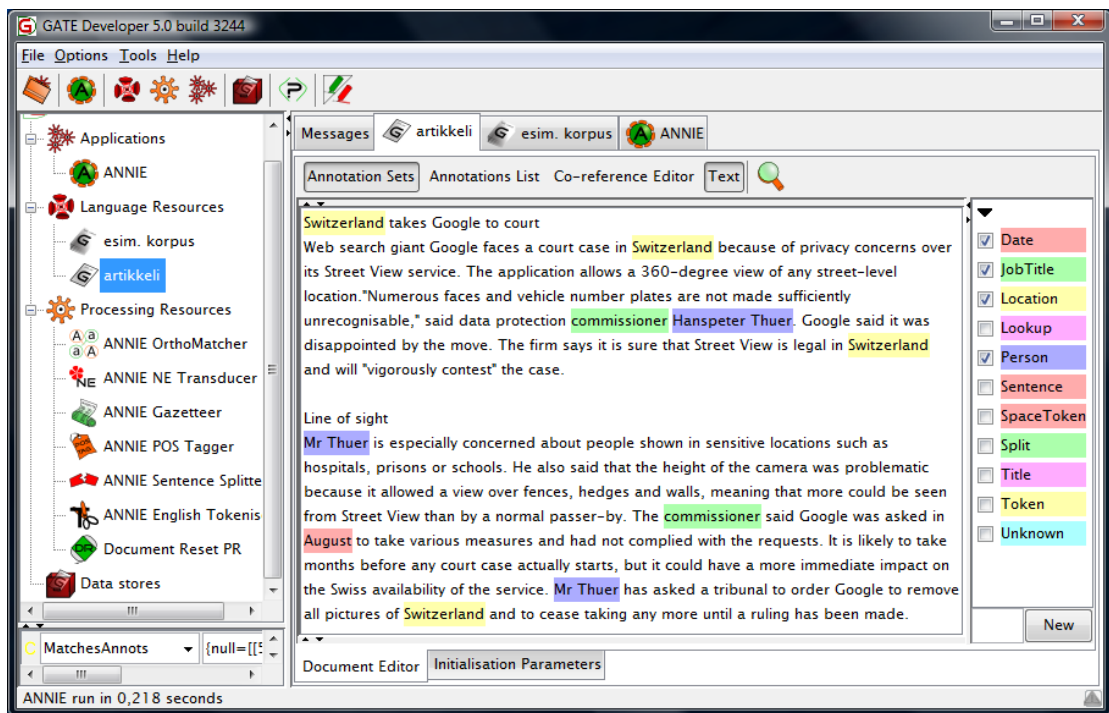
Seuraavassa esimerkissä käsitellään lyhyt teksti ANNIE-prosessiputken avulla. Tekstinä kokeilussa käytettiin BBC News -sivustolta otettua artikkelia [BBC News, 2009]. Ensin artikkeli tallennettiin GATE:n kielivarantoihin (Language Resources). Seuraavaksi luotiin esimerkikikorpus, johon teksti lisättiin. Tämän jälkeen työpöydälle ladattiin ANNIE:n perustyökalut, joiden läpi teksti ajettiin. Teksti ajettiin seuraavien prosessiputkien (ks. kohta 3.2) läpi järjestyksessä: English Tokeniser, Sentence Splitter, POS Tagger ja Gazetteer.



Kuva 3. Keltaisella merkityt merkkijonot on tunnistettu ja luokiteltu.

Kuvassa 3 näkyvät keltaisella kaikki tunnistetut merkitykselliset merkkijonot eli tokenit. Jokaisen tällaisen tokenin annotaatioita voi tarkastella ja kuvassa näkyikin sanan "says" annotaatiot. Sen sanaluokka on merkitty lyhenteellä VBZ, joka tarkoittaa presensissä olevaa yksikön kolmannen persoonan verbiä.

Lisäksi ilmenee, että se on neljä merkkiä pitkä pienillä kirjaimilla kirjoitettu sana (ei siis esim. numero). Kuvassa 4 puolestaan on korostettu eri väreillä gazetteerin tunnistamia kohtia. Keltaisella näkyvät maantieteelliset paikat, ammatinimikkeet ja violetilla henkilöt. Käsittelytyökaluja voi ladata lisää ja käyttää tarpeen mukaan. Myös annotaatioita voi käsin lisätä ja muokata.



Kuva 4. Korostetut sanat ovat Gazetteer-listoista tunnistettuja sanoja.

4. GATE:n käyttömahdollisuudet

Tässä luvussa esitellään GATE:n eri käyttötarkoituksia. Näitä ovat kielen käsittelytyökalu, ohjelmistokehitysalusta, opetustyökalu ja tiedonpoimintatyökalu. GATE on siis erittäin laaja ohjelmisto, joka palvelee monia eri käyttäjäryhmiä ja käyttötarkoituksia samanaikaisesti. Juuri GATE:n joustavuus ja laajennettavuus ovat mahdollistaneet sen monikäyttöisyyden. Tämä kertoo myös siitä, että GATE on onnistunut saavuttamaan tavoitteitaan ja on todistanut toimivuutensa myös todellisten käyttäjien piirissä.

4.1. GATE kieliteknologiatyökaluna

GATE perimmäisiin käyttötarkoituksiin kuuluu toimia kieliteknologisenä työkaluna sekä kielen käsittelyn että nyttemmin myös tiedonpoiminnan aloilla. Viime vuosina GATE onkin selvästi panostanut ohjelmiston kehittämiseen juuri IE-alalla. Esimerkiksi AguaLog-ohjelma käyttää GATE:n NL-komponentteja ja Mumis-projektissa on hyödynnetty GATE:n ANNIE kokoelmaa. AguaLog on kyselyohjelma, joka osaa toimia luonnollisella kielellä [Lopez et al., 2005]. Mu-

mis taas indeksoi ja hakee monikielisiä multimediadokumentteja [Jong and Westerveld, 2001; Saggio et al., 2004].

AquaLog käyttää GATE:n kielen käsittelykomponentteja semanttisen materiaalin työstämiseen. GATE vastaa siis kielellisen tiedon käsittelystä kysymyksiä, vastauksia ja tiedostoja käsitellessä. Käytössä on siis vain yksi GATE:n ominaisuus, jota käytetään rajapinnan kautta, eikä komponenttien sovittaminen muuhun ohjelmistoon vaatinut kovinkaan paljoa töitä. AquaLogissa on hyödynnetty myös muita tekniikoita mahdollisimman hyvän tuloksen saamiseksi. Juuri tällaiseen joustavaan ja mukautettavaan ohjelmistokehitykseen GATE on yrittänytkin pyrkiä.

Mumis kerää tietoja jalkapalloa koskevista monikielisistä (saksa, englanti, hollanti) dokumenteista, jotka on kerätty tekstilähteistä, radio-ohjelmista ja tv-lähetyksistä, ja näistä tiedoista on rakennettu hakutietokanta [Declerck et al., 2001]. Mumis käyttää GATE:n kielen käsittely ja ontologiaominaisuuksia [Kiryakov, 2003]. Kiryakovin mukaan GATE:n senhetkinen ontologia-lisäosa tuki tekstien luokkahierarkioita, suhteiden määrittelyä sekä alueen ja vaihteluvälin rajausta. Näiden semanttisten tietojen avulla teksteille voidaan antaa semanttisia ominaisuuksia, joiden perusteella voidaan suorittaa erilaisia hakuja.

Bontcheva ja muut [2003] muistuttavatkin GATE:n monista eduista monikielisen IE:n alalla. Esimerkiksi Unicode-tuki, komponenttien kieliriippumattomuus sekä algoritmien ja kielellisten resurssien erottelu tukevat monikielistä tiedonpoimintaa. Juuri Mumiksen kaltaisissa projekteissa näkyy käytännössä se, miten GATE tukee informaatiojärjestelmiä, jotka joutuvat käsittelemään suuria monikielisiä, monimuotoisia ja monimediaisia datamääriä [Bontchevan et al., 2002a]. Bontcheva ja muut korostavat, että erityisesti GATE:n laajuus ja matalan tason komponentit takaavat sen, että tarvittavat toiminnot ovat helposti saatavilla ja hyödynnettävissä.

4.2. GATE ohjelmistokehitysalustana

GATE:n suunnitteluperiaatteisiin on aina kuulunut toimia ohjelmistokehitysalustana. Tästä hyvänä esimerkkinä toimii Emille-projekti, jossa sovellettiin jo olemassa olevia GATE-komponentteja ja kehitettiin uusia. EMILLE (Enabling MInority Language Engineering) on eteläaasialaisten kielten korpus, joka sisältää 63 miljoonaa sanaa puheen ja tekstin muodossa ja jonka tarkoituksena on edistää eteläaasialaisten kielten tutkimusta ja käännöstyökaluja [McEnery et al., 2000]. Emille on jatkoa Mille-projektille, jonka tarkoituksena oli tuottaa lingvistisiä työkaluja Englannin ei-alkuperäisille vähemmistökielille [McEnery et al., 2000].

Projektin suurimmaksi ongelmaksi nousi 8-bittisen kielen kääntämisen Unicode-muotoon. Lisäksi haasteellista oli sisällyttää korpuksen eri kieliä ja eri

kirjoitusjärjestelmiä [McEnery et al., 2000; Baker et al., 2002]. Tämän seurauksena GATE:en piti rakentaa uusi selaaja (tokenizer) [McEnery et al., 2000] ja olemassa olevia merkintätyökaluja (tagger) piti muunnella ja opettaa uusiin kielijärjestelmiin sopiviksi. GATE:en piti myös luoda Unicodea tukeva editori, virtuaalinen lisänäppäimistö sekä tuki 8-bittiselle syötteelle ja tulosteelle [Baker et al., 2002].

4.3. GATE Opetuskäytössä

GATE:a voi käyttää monitasoisena opetustyökaluna kieliteknologian ja vieras-kielen opetuksessa. GATE:n soveltuvuutta opetuskäyttöön on ehkä helpoin tutkia vertaamalla sitä kohdassa 2.3 esiteltyyn NLTK:hon, joka on kehittynyt lähinnä opetuksen ympärille.

Lähtökohtaisesti siis NLTK on soveliaampi opetuskäyttöön. NLTK:n suurin etu GATE:en verrattuna on sen helppous ja yksinkertaisuus, joka samalla on myös sen suurin ongelma. Esimerkiksi NLTK:n käyttämä Python-kieli on loogista ja helposti opittavaa, mutta GATE:n käyttämä Java on huomattavasti tehokkaampaa. Varsinkin perusteista lähtevässä opetuksessa NLTK on kuitenkin GATE:en verrattuna juuri helppoutensa takia etulyöntiasemassa. Myös valmiit opetusta tukevat materiaalit ovat NLTK:n suuri etu.

GATE tarjoaa tosin muutaman opiskelussa hyödyllisen ominaisuuden, joita NLTK:sta ei löydy. GATE esimerkiksi tallentaa automaattisesti kaikkien prosessien suoritustiedot [Bontcheva 2002]. Opiskelijat pystyvät tietojen avulla helposti arvioimaan kehittämäänsä ohjelman tehokkuutta ja seuraamaan toimintojen suoritusta. GATE sisältää myös virheenkorjausominaisuuden, joka helpottaa suuresti ongelmakohtien löytämistä ja ratkaisemista [Cunningham et al., 2002]. Käyttövalmis käytännönohjelmisto toimii myös malliesimerkkinä opiskelijoille todellisuudessa käytettävästä kielenanalysointiohjelmistosta [Bontcheva et al., 2002].

Myös GATE soveltuu siis hyvin kieliteknologian opettamiseen, vaikka sen vahvuudet painottuvatkin hieman eri alueille. NLTK:n edut piilevät sen yksinkertaisuudessa, kun taas GATE tarjoaa opiskelualustaksi varsinaisessa ohjelmistokehityksessäänkin paljon käytetyn ohjelmiston. NLTK:ssa varjoon jäävät laajennettavuus, tehokkuus ja kätevyys. GATE:ssa ei ole näitä ongelmia, mutta se ei kykene täysin vastaamaan NLTK:n laajaan tukeen oppimisessa ja opetuksessa. GATE soveltuukin ehkä paremmin hieman vaativampiin opetustilanteisiin.

5. GATE:n arviointi

Kuten olemme voineet todeta, GATE on monipuolinen ja monikäyttöinen kieliteknologinen sovellus. Maynardin [2005] tekemässä vertailututkimuksessa

GATE sijoittuikin kaikkien muiden vertailussa olleiden sovellusten edelle. Mukana olivat GATE:n lisäksi KIM, MnM, Magpie ja OntoMat. Kaikki sovellukset olivat avoimen lähdekoodin XML-pohjaisia automatisoituja tekstin annotointiohjelmistoja.

Tutkimuksen mukaan GATE:n heikkouksia olivat huono skaalautuvuus, konfiguroinnin hankaluus, vaikeaselkoinen ohjeistus ja hitaus suuria tekstimääriä käsiteltäessä. Suurillakin syötemäärillä GATE:n tulokset olivat kuitenkin verrattain tarkkoja ja hieman moitteita saanut dokumentaatio oli linkitettyä ja näin helposti löydettävissä. Maynard lukee GATE:n muihin vahvuuksiin toimivuuden eri käyttöympäristöissä, monien formaattien tukemisen, helpon asentamisen, saatavuuden ja kokonaisuutena hyvän suorituskyvyn.

Vaikka GATE siis todistetusti soveltuu myös suurempien tekstimäärien käsittelyyn, tulosten laajempi tarkastelu on hankalaa. GATE ei esimerkiksi tarjoa välineitä kielen käsittelyn jälkeen tapahtuvaan analysointiin ja työstöön ainakaan lingvistisestä näkökulmasta. Esimerkiksi sanojen esiintymisyhteyden tai -yleisyyden tutkiminen ei onnistu.

GATE:n käyttöliittymä perustuu hyvin pitkälle ohjelmistoarkkitehtuurin rakenteeseen. Varsinkin aloittelijalle tämä voi tuottaa hankaluuksia, sillä hän ei välttämättä ymmärrä taustalla olevaa rakennetta eikä tunne kaikkia komponentteja. GATE:n käyttö vaatiikin huomattavaa paneutumista ohjelmiston osiin ja toimintaperiaatteisiin. Kaikki työkalut ja aineistot pitää ladata ohjelmiin erikseen ja työstää tietyssä järjestyksessä. Vaatii siis paljon taustatietoja, että käyttäjä löytää ja saa käyttöönsä oikeat työkalut oikeisiin teksteihin oikeassa järjestyksessä. Tosin GATE tarjoaa kohtuullisen hyvän linkitetyn ohjeistuksen, vaikka se ei varsinaisesti autakaan kokonaisuuden hahmottamisessa.

Mikäli kieliteknologia sovellusten kenttää katsotaan kokonaisuutena, GATE voidaan sijoittaa avointen ohjelmistojen kärkeen. Siitä on ajan mittaan tullut paljon käytetty ja näin ollen sitä on myös pystytty muokkaamaan vankan käyttäjäkokemuksen perusteella. On kuitenkin todettava, ettei GATE:a voi silti sanoa kovinkaan käyttäjäystävälliseksi ainakaan aloittelijoiden näkökulmasta. GATE:n ohjelmistoarkkitehtuuri ja suunnitteluperiaatteet ovat mahdollistaneet sen laajentamisen uusille kieliteknologisille osa-alueille (esim. IE) ja kehitys tulee varmasti olemaan myös jatkossa samansuuntaista.

6. GATE:n kehityssuuntia

Kieliteknologia on ottanut viime vuosina suuria harppauksia eteenpäin, mutta sen todelliseen läpimurtoon lienee vielä aikaa. Väyrysen [2002] mielestä esimerkiksi hakuohjelmien tarkkuus ja tehokkuus eivät vielä ole riittäviä. Väyrynen näkee tulevan kehityksen kärjessä tilastollisten menetelmien kehittämisen

luonnollisen kielen käsittelyn sijaan, mikä tosin voi tuottaa integrointiongelmia nykyisen lingvistisen tiedon kanssa. Väyrysen mielestä kieliteknologisten ohjelmistojen tulisi olla lähes täydellisiä, jotta niistä olisi hyötyä käytännössä.

Bontcheva ja muut [2004] näkevät monia kehitysalueita, joissa GATE voisi olla osallisena. Näihin kuuluvat esimerkiksi monikielisyyden haasteet, kielen tuottaminen, laajat tiedonpoimintaa ja hyödyntämistä koskevat hankkeet sekä Väyrysen [2002] mainitsema tilastomallien ja tietokoneoppimisen yhdistäminen aiempiin kielellisiin malleihin. Crane ja muut [2005] huomauttavat, että GATE tukeutuu vielä lähinnä kielellisiin malleihin, kun taas esimerkiksi BBN Identifier soveltaa itseoppivia algoritmeja (learning algorithms).

Myös semanttisen verkon kehittäminen kuuluu tämän hetken haasteisiin. Bontcheva ja muut [2004] esittävät, että helposti valjastettavat kieliteknologian sovelluksen voisivat käsitellä suuriakin tekstikokoelmia nopeasti ja näin lisätä synergiaa näiden kahden alan välillä. Tietorakenteiden ja ontologioiden tukemisella voisi olla merkittäviäkin etuja webin semanttista kerrosta rakennettaessa. Esimerkiksi kohdassa 4.1 esitelty AquaLog on osoitus juuri tällaisesta yhteystoiminnasta.

Hieman kaukokatseisempia kehityssuuntia voisivat olla esimerkiksi luonnollisen kielen hakukäyttöliittymät [Tablan et al., 2008], semanttinen sähköpostinhallinta [Laclavik and Maynard, 2009], vektorimallintamisen hyödyntäminen dokumenttien luokittelussa [Li and Bontcheva, 2008] tai vaikkapa kvanttimalintamisen soveltaminen informaation hakuun [Li and Cunningham, 2008].

7. Yhteenveto

Kuten olemme nähneet, GATE on monipuolinen ja joustava työkalu, jonka LKK- ja IE-ominaisuuksia voidaan hyödyntää lähes kaikilla kieliteknologian osa-alueilla. GATE:n suunnitteluperiaatteet ja ohjelmistoarkkitehtuuri ovat mahdollistaneet GATE:n monipuolisen käytön mitä erilaisimmissa sovelluksissa. GATE:n toimivuudesta ja hyödyllisyydestä kertovat kaikki lukuisat sovellukset, joissa GATE on osallisena. Vaikka GATE:lla onkin paljon kilpailijoita, se on onnistunut vakiinnuttamaan asemansa tieteellisessä yhteisössä.

Toistaiseksi GATE on onnistunut kohtuullisesti säilyttämään toimivuutensa, vaikka ohjelmisto on laajentunut huomattavasti. Laajennuksista huolimatta GATE on pysyttäytynyt läheisesti luonnollisen kielen käsittelyyn liittyvissä toiminnoissa, eikä ole lähtenyt rönsyilemään kaikille mahdollisille kieliteknologian osa-alueille. Onkin ehkä parempi keskittyä tiettyyn osaamisalueeseen, vaikka teoriassa GATE:n arkkitehtuuri mahdollistaisi laajentumisen lähes mihin suuntaan tahansa. Toisaalta GATE:ssa olisi potentiaalia kehittyä ensimmäi-

seksi järjestelmäksi, jossa olisi sovitettu yhteen kaikkien kieliteknologian alojen työkalut.

Viiteluettelo

- [Aswani et al., 2005] Niraj Aswani, Valentin Tablan, Kalina Bontcheva, and Hamish Cunningham, Indexing and querying linguistic metadata and document content. In: *Proc. of 5th International Conference on Recent Advances in Natural Language Processing RANLP'2005*, 35–44.
- [Baker et al., 2002] Paul Baker, Anrew Hardie, Tony McEnery, Hamish Cunningham, and Rob Gaizauskas, EMILLE, a 67-million word corpus of the indic languages: data collection, mark-up and harmonisation. In: *Proc. of the 3rd Language Resources and Evaluation Conference LREC'2002*, 819–825.
- [BBC News, 2009] BBC News, Switzerland takes Google to court, <http://news.bbc.co.uk/2/hi/business/8358908.stm>. Checked 20.11.2009
- [Bird, 2006] Steven Bird, NLTK: the natural language toolkit. In: *Proc. of the COLING/ACL on Interactive Presentation Sessions*, 69–72.
- [Bird and Loper, 2004] Steven Bird and Edward Loper, NTLK: the natural language toolkit. In: *Proc. of the ACL 2004 on Interactive Poster and Demonstration Sessions*, No. 31.
- [Blanc and Constant, 2006] Olivier Blanc and Matthieu Constant, Outilex, a linguistic platform for text engineering. In: *Proc. of the COLING/ACL on the Interactive Presentation Session*, 73–76.
- [BNC] Using BNC with Xaira/Sara, <http://www.natcorp.ox.ac.uk/tools/index.xml>. Checked 7.12.2009
- [Bontcheva, 2004] Kalina Boncheva, Open-source tools for creation, maintenance, and storage of lexical resources for language generation from ontologies. In: *Proc. of 4th Language Resources and Evaluation Conference LREC'04*.
- [Bontcheva et al., 2004] Kalina Bontcheva, Valentin Tablan, Diana Maynard and Hamish Cunningham, Evolving GATE to meet new challenges in language engineering. *Natural Language Engineering Archive* **10**, 3–4 (Sep. 2004), 349–373.
- [Bontcheva et al., 2003] Kalina Bontcheva, Diana Maynard, Valentin Tablan, and Hamish Cunningham, GATE: a unicode-based infrastructure supporting multilingual information extraction. In: *Proc. Of Recent Advances in Natural Language, Workshop on Information Extraction for Slavonic and Other Central and Eastern European Languages RANLP'2003*.
- [Bontcheva et al., 2002a] Kaling Bontcheva, Hamish Cunningham, Diana Maynard, Valentin Tablan, and Horacio Saggion, Developing reusable and ro-

- bust language processing components for information systems using GATE. In: *Proc. Of 3rd Natural Language and Information Systems NLIS'2002*, 223–227.
- [Bontcheva et al., 2002b] Kalina Bontcheva, Hamish Cunningham, Valentin Tablan, Diana Maynard, and Oana Hamza, Using GATE as an environment for teaching NLP. In: *Proc. of the ACL, Workshop on Effective Tools and Methodologies in Teaching NLP and Computational Linguistics* **1**, 54–62.
- [Burnard, 2007] Lou Burnard, XAIRA: software for language analysis. In: *Proc. of Digital Historical Corpora – Architecture, Annotation, and Retrieval*.
- [Crane et al., 2005] Gregory Crane, Kalina Bontcheva, Jeffrey A. Rydberg-Cox and Clifford Wulfman, Emerging language technologies and the rediscovery of the past. *International Journal on Digital Libraries* **5**, 4 (Aug. 2005), 309–316.
- [Crane, 1998] Gregory Crane, The perseus project and beyond: how building a digital library challenges the humanities and technology. *D-Lib Magazine* (Jan. 1998), 1-18.
- [Cole, 1997] Ron Cole, *Survey of the state of the art in human language technology*. Cambridge University Press, New York, USA, 1997.
- [Cunningham et al., 2009] Hamish Cunningham, Diana Maynard, Cristian Ursu, Marin Dimitrov, Mike Dowman, Niraj Aswani, Ian Roberts, Yaoyong Li, Andrey Shafirin, and Adam Funk, Developing language processing components with GATE version 5 (a user guide). Available as <http://gate.ac.uk/sale/tao/tao.pdf>. Checked 7.12.2009.
- [Cunningham et al., 2003] Hamish Cunningham, Valentin Tablan, Kalina Bontcheva, and Marin Dimitrov, Language engineering tools for collaborative corpus annotation. In: *Proc. of the Corpus Linguistics 2003*, 165–173.
- [Cunningham et al., 2002] Hamish Cunningham, Diana Maynard, and Kalina Bontcheva, Valentin Tablan, GATE: an architecture of robust HLT applications. In: *Proc. of Recent Advances in Natural Language Processing*, 168–175.
- [Cunningham et al., 2000a] Hamish Cunningham, Kalina Bontcheva, Wim Peters, and Yorick Wilks, Uniform language resources access and distribution in the context of a general architecture for text engineering. In: *Proc. of OntoLex'2000, Workshop on Ontologies and Language Resources*.
- [Cunningham et al., 2000b] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, and Yorick Wilks, Experiences of using GATE for NLP R&D. In: *Proc of COLING'2000, Workshop on Using Toolsets and Architectures to Built NLP Systems*.
- [Cunningham et al., 2000c] Hamish Cunningham, Diana Maynard, and Valentin Tablan, JAPE: a java annotation patterns engine (second edition).

Technical report CS-00-10, University of Sheffield, Department of Computer Science.

- [Cunningham et al., 1997a] Hamish Cunningham, Kevin Humphreys, and Robert Gaizauskas, GATE: a TIPSTER-based general architecture for text engineering. In: *Proc. of the TIPSTER Text Program (Phase III) 6 Month Workshop*.
- [Cunningham et al., 1997b] Hamish Cunningham, Kevin Humphreys, Robert Gaizauskas, and Yorick Wilks, Software infrastructure for natural language processing. In: *Proc. of the TIPSTER Text Program (Phase III), 6 Month Workshop*, 237–244.
- [Curran, 2003] James R. Curran, Blueprint for a high performance NLP Infrastructure. In: *Proc. of the Human Language Technology Conference, Workshop on Software Engineering and Architecture of Language Technology Systems HLT-NAACL'2003 8* (2003), Association for Computational Linguistics, 39–44.
- [Declerck et al., 2001] Thierry Declerck, Peter Wittenburg, Hamish Cunningham, The automatic generation of formal annotations in a multimedia indexing and searching environment. In: *Proc. of the Workshop on Human Language Technology and Knowledge Management*, No. 17.
- [Durian, 2002] David Durian, Corpus-based text analysis from a qualitative perspective: a closer look at Nvivo. *Style* **36**, No. 4, 738–742.
- [Ide, 2000] Nancy Ide, The XML framework and its implications for the development of natural language processing tools. In: *Proc. of the COLING, Workshop on Using Toolsets and Architectures to Build NLP Systems*.
- [Harman, 1992] Donna Harman, The DARPA TIPSTER project. *ACM SIGR Forum* **26** (Fall 1992), 26–28.
- [Jong and Westerveld, 2001] Franciska de Jong, Thijs Westerveld, MUMIS: multimedia indexing and searching. In: *Proc. of the Content-Based Multimedia Indexing CBMI'2001*, 423–425.
- [Kiryakov, 2003] Kiryakov Anastas, Ontology and reasoning in MUMIS: Towards the Semantic Web. Technical Report CS-03-03, Department of Computer Science, University of Sheffield, 2003.
- [Laclavik and Maynard, 2009] Michal Laclavik and Diana Maynard, Motivating intelligent email in business: an investigation into current trends for email processing and communication research. In: *Proc. Of the 11th IEEE Conference on Commerce and Enterprise Computing, Workshop on Emails in e-Commerce and Enterprise Context*, 476–482.
- [Leidner, 2003] Jochen L. Leidner, Current issues in software engineering for natural language processing. In: *Proc. of the Human Language Technology*

Conference, Workshop on Software Engineering and Architecture of Language Technology Systems HLT-NAACL'2003 **8**, 45–50.

- [Li and Bontcheva, 2008] Yaoyong Li and Kalina Bontcheva, Adapting support vector machines for F-term-based classification of patents. *ACM Trans. Asian Lang. Inf. Process.* **7**, 2 (2008), No. 7.
- [Li and Cunningham, 2008] Yaoyong Li, Hamish Cunningham, Geometric and quantum methods for information retrieval. *ACM SIGIR Forum* **42**, 2 (2008), 22–32.
- [Loper and Bird, 2002] Edward Loper and Steven Bird, NTLK: the natural language toolkit. In: *Proc. of the ACL'2002, Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics* **1**, 63–70.
- [Lopez et al., 2005] Vanessa Lopez, Michele Pasin and Enrico Motta, AquaLog: An Ontology-portable Question Answering System for the Semantic Web. In: *Proc of ESWC 2005*, 546–562.
- [Maynard, 2005] Diana Maynard, Benchmarking ontology-based annotation tools for the semantic web. In: *Proc. Of the UK e-Science Programme All Hands Meeting, Workshop on Text Mining, e-Research and Grid-enabled Language Technology AHM'2005*.
- [Maynard et al., 2002] Diana Maynard, Valentin Tablan, Hamish Cunningham, Cristian Ursu, Horacio Saggion, Kalina Bontcheva and Yorick Wilks, Architectural elements of language engineering robustness. *Natural Language Engineering* **8**, 3 (Jun. 2002), 257–274.
- [McEnery et al., 2000] Anthony McEnery, Paul Baker, Rob Gaizauskas, and Hamish Cunningham, EMILLE: building a corpus of south asian languages. *Vivek, Aquaterly in Artificial Intelligence* **13**, 3 (2002), 23–32.
- [Rösner and Kunze, 2002] Dietmar Rösner and Manuela Kunze, An XML-based document suite. In: *Proc. of the 19th International Conference on Computational Linguists* **2**, 1–5.
- [Saggio et al., 2004] Horacio Saggion, Hamish Cunningham, Kalina Bontcheva, Diana Maynard, Oana Hamza and Yorik Wilks, Multimedia indexing through multi-source and multi-language information extraction: the MUMIS project. *Data & Knowledge Engineering archive* **48**, 2 (2004), 247–264.
- [Schmid et al., 1996] Paul Schmid, Axel Theofilidis, Sibylle Rieder, Thierry Declerck, Lean formalisms, linguistic theory, and applications: grammar development in ALEP. In: *Proc. of the 16th conference on Computational linguistics* **1**, 286–291.

[Tablan et al., 2008] Valentin Tablan, Danica Damljanovic and Kalina Bontcheva, A natural language query interface to structured information. *The Semantic Web: Research and Application, Lecture Notes in Computer Science* **5021** (2008), 361–375.

[Xaira] Xaira Page, <http://www.oucs.ox.ac.uk/rts/xaira>. Checked 7.12.2009.

Ohjaaja: Jaakko Hakulinen

Älykkäät agentit oppimisen apuvälineenä

Karoliina Käki

Tiivistelmä.

Agentit ovat käyttöliittymän osia, elektronisia avustajia, jotka kommunikoivat käyttäjän kanssa. Agenteilla on monia eri käyttötarkoituksia, kuten esimerkiksi opetus, ohjeistus, tiedonhallinta, yhteydenpito ja tiedon suodatus. Tässä tutkielmassa keskityn kuitenkin vain oppimisympäristöissä käytettäviin älykkäisiin, pedagogisiin agentteihin. Pohdin tutkielmassa, miten agenttien eri roolit, ulkonäkö ja sukupuoli vaikuttavat oppilaan kokemuksiin ja oppimistuloksiin. Tutkielmassa en puutu siihen, mitä tietoa agenttien avulla voidaan opettaa, enkä agenttien tekniseen toteutukseen. Työssä kerron myös miten ja millä eri tavoin agentti voi opettaa, auttaa, tukea ja motivoida oppilastaan ja miten agentin toiminta vaikuttaa lopulta oppilaan oppimiseen.

Avainsanat ja -sanonnat: älykkäät agentit, virtuaalinen oppimisympäristö, agenttien roolit, pedagogiset agentit, personoitu opetus.

CR-luokat: I.2.11, H.5.2, K.3

1. Johdanto

Virtuaalisten oppimisympäristön käyttö on yleistynyt opetuksessa. Niiden avulla oppiminen voi tapahtua ajasta ja paikasta riippumatta. Monet oppimisympäristöt ja elektroniset oppimateriaalit eivät pidä huolta oppilaansa motivaatiosta, eivätkä oppimisesta niin kuin oikea reaali maailman opettaja yrittää. Oppilaan motivaatio on kuitenkin ehdoton edellytys oppimiselle. Opettajilla on erittäin haastava työ motivoida oppilaitaan opiskelussa. Opettajat eivät pysty antamaan kaikille oppilailleen persoonallista, juuri heidän oppimistyyliinsä sopivaa opetusta, vaikka haluaisivatkin. Onneksi kuitenkin yhä useammassa virtuaalisissa oppimisympäristöissä ja harjoituksissa oppilaan apuna toimii elektroninen avustaja, älykäs agentti. Agentti voi motivoida, seurata oppilaan edistymistä, auttaa oppilasta vaikeissa tehtävissä tai opettaa oppilasta. Nykyään on mahdollista saada persoonallista, motivoivaa opetusta iästä, paikasta ja ajasta riippumatta. Persoonallista opetusta voi antaa myös Internetissä toimiva mukautuva koulutusjärjestelmä.

Tietokonepohjaiset virtuaaliset oppimisympäristöt ovat avoimia järjestelmiä, jotka mahdollistivat vuorovaikutukset ja kohtaamiset muiden ihmisten kanssa sekä tarjoavat pääsyn laajalle toimialueelle. Virtuaalinen oppimisympäristö voi täydentää kasvokkain oppimisen metodeja tai korvata nämä opetus-

metodit kokonaan tapauksessa, jossa välimatka on esteenä perinteiselle opetukselle. Virtuaalinen oppimisympäristö tarjoaa sopivilla ja joustavilla ehdoilla myös lukuisia etuja verrattuna perinteiseen opetusympäristöön. (Buendia, Cano, & Benllock, 2009) Maantieteelliset rajat eivät muodostu esteeksi virtuaalisten oppimisympäristöjen käytölle. Potentiaaliset oppilaat kaukaisiltakin alueilta ympäri maailmaa kykenevät löytämään toisensa ja opiskelemaan oppimisympäristöissä alhaiseen hintaan. Näistä syistä virtuaalisten oppimisympäristöjen käyttö on yksi nopeimmin kasvavista koulutusteknologian tutkimuksen ja kehityksen alueista.

Ensin esittelen tutkielmassani älykkäiden agenttien joitakin yleisempiä määritelmiä lyhyesti sekä kerron älykkäiden agenttien tehtävistä. Toisessa luvussa kerron agenttien hyvinä pidetyistä ominaisuuksista. Tämän jälkeen ryhdyn pohtimaan agenttien erilaisia rooleja opetukseen liittyen ja sitä, miten ne vaikuttavat oppilaan oppimiseen. Kolmannessa luvussa käyn läpi agenttien erilaisien ulkoasujen ja sukupuolen vaikutusta oppilaisiin. Neljännessä luvussa pohdin, miten agentit pystyvät vetoamaan oppilaan tunteisiin ja saavat pidettyä oppilaan motivaation yllä sekä pohdin pystyvätkö pedagogiset agentit tunteisiin vetoamalla vaikuttamaan oppilaan oppimistuloksiin. Luvussa 5 käsittelen lyhyesti muutamaan tutkimukseen perustuen, miten älykkäät agentit antavat persoonallista opetusta oppilaalle sekä mitkä ovat sen vaikutukset oppimiseen. Esittelen myös lyhyesti, miten eri oppimistyylien edustajat kokevat oppivansa agentin kanssa työskennellessään.

2. Älykkäät agentit

Älykkäät agentit ovat ohjelmistokokonaisuuksia, jotka tukevat keinotekoisista koneen älykkyyttä. Älykäs agentti on vaikea määritellä yksiselitteisesti, koska agentteja on monenlaisia, monissa eri toimintaympäristöissä. Agentteja löytyy Internetin verkkosivuilta, erilaisista tietokoneohjelmista sekä virtuaalisista oppimisympäristöistä. Ohjelmistoissa toimivista agenteista käytetään nimitystä ohjelmistoagentti (software agent), se on osa ohjelmistoa, mikä toimii jonkun puolesta.

Älykkäät agentit ovat itsenäisesti toimivia ohjelmia, jotka suorittavat tehtäviä ympäristönään tietoverkko. Ne ovat ohjelmistoja, jotka ovat suunniteltu toimimaan itsenäisesti ja mukautuvasti, kehittäjiensä ja jokapäiväisien käyttäjien määrittelemien päämäärien saavuttamiseksi (Haynes, Cohen, & Ritter, 2009). Älykkään agentin tehtävät riippuvat sen toimintaympäristöstä ja käyttötarkoituksesta. Tiedon suodatus on agentin yksi käyttötarkoitus. Tiedon suodatuksessa agentit keräävät tietoa, tämän jälkeen arvioivat tiedon laadun ja lopuksi valikoiden esittävät tietoa käyttäjälle. Agentti toimii myös avattarena monen

käyttäjän etäyhteyksissä. Kun agentin tehtävänä on hallinnoida tietoa, se tallentaa ja järjestää tiedot tärkeysjärjestykseen sekä tarvittaessa muistuttaa tai antaa käyttäjälle datavarastostaan hänen kaipaamansa tiedon.

Tutkielmassa käsitellään vain oppimisympäristöissä olevia pedagogisia agentteja. Animoiduilla pedagogisilla agenteilla on ihmismäisiä piirteitä kuten tunteiden ilmaiseminen kasvoilla sekä eleillä, ihmisten tunteiden ymmärtäminen ja puheen tai tekstin muodossa keskusteleminen (Jaques, Lehmann, & Pesty, 2009). Pedagoginen agentti pystyy oppimaan oppilaastaan asioita, oppilaan omien mieltymyksiä, kiinnostuksen ja käytöksen perusteella. Agentti muokkaa oppilaasta opitun käyttäjämallin mukaan omaa käytöstään ja, näin pyrkii antamaan tietokoneen mahdollisuuksien rajoissa proaktiivista ja persoonallista apua oppilaalle. (Schiaffino, Amandi, Gasparin, & Pimenta, 2008) Pedagogisten agenttien tehtävänä on esimerkiksi esittää tietoja ja motivoida oppilasta oppimisympäristössä (Blanchard & Frasson, 2004). Ne parantavat virtuaaliopetuksen sisältöä ja kokemuksia, koska agentit osaavat antaa oppilaalle apua, neuvoja, palautetta sekä pystyvät tuottamaan oppilaalle persoonallista opetusta. Agentti helpottaa koneen ja oppilaan vuorovaikutusta. Se arvostelee oppilaan suorituksia ja voi näin helpottaa reaali maailmassa opettajan tai ohjaajan työtä. (Ben Ammar & Neji, 2006) Agentti voi myös toimia oppilaan opiskelutoverina ja opastaa häntä tällaisesta näkökulmasta.

Tekoäly ja ihmismäiset piirteet toimivat älykkäiden agenttien määrittämisen yhdistävänä tekijänä. Agenttitekniikan kehittyessä agenttien tekoäly kasvaa ja niiden toiminnot tulevat monipuolistumaan. Agenteille saadaan uusia määritelmiä ja vanhoja joudutaan muuttamaan. Tulevaisuudessa älykkään agentin käsite saattaa olla niin tunnettu ja selkeä, ettei sitä edes tarvitse määritellä.

2.1. Agenttien ominaisuudet

Pedagogisilla agenteilla halutaan olevan monia erilaisia ominaisuuksia, koska oppilaat usein mieltävät ne ihmismäisiksi. Agenttien toimiminen tärkeissä tehtävissä kuten opetuksessa vaatii paljon erilaisia toimintoja ja ominaisuuksia niiltä. Agenttien yhdeksi ominaisuudeksi määritellään autonomisuus, koska ne osaavat kontrolloida joitakin omia toimintojaan (Agarwal, Deo, & Das, 2004). Omien toimintojen kontrollointi tarkoittaa, ettei agentin tarvitse odottaa käyttäjältä kysyjä, vaan se osaa tehdä itsenäisesti päätöksiä ja toimia niiden mukaan. Autonomisuus on erittäin oleellinen ominaisuus pedagogiselle agentille, jotta sen kanssa työskentely olisi mielekästä ja uskottavaa. Agenteista saadaan myös kullekin oppilaalle persoonallinen opettaja, avustaja tai opiskelutoveri, koska agentti kerää oppilaastaan opiskeltaessa tietoa, jonka mukaan se muokkaa käytöstään oppilaalle sopivaksi. (Marcal & North, 2005)

Agentit sopeutuvat uusiin ja erilaisiin tilanteisiin hyvin. Ne esimerkiksi osaavat vastata oppilaan esittämiin kysymyksiin, tosin niiltä saama vastaus ei ole aina oikea. (Marcal & North, 2005) Agentin antamat väärät vastaukset johtuvat sen tekoälyn rajallisuudesta. Se voi ymmärtää väärin oppilaan esittämän kysymyksen tai agentin kapasiteetti ei riitä ymmärtämään kysymystä lainkaan. Agentin yksi tärkeimmistä ominaisuuksista on sosiaalisuus, koska niiden täytyy työskennellä yhdessä muiden opiskelijoiden ja agenttien kanssa auttaakseen heitä saavuttamaan päämääränsä (Marcal & North, 2005; Agarwal et al., 2004). Hyvä agentti on myös sinnikäs, koska agentti on väsymättömästi oppilaan opiskelussa mukana (Agarwal et al., 2004). Se on koko ajan oppilaan saatavilla sekä työstää ja ajaa sillä olevaa tietoa.

Agentin on oltava reaktiivinen huomatakseen ympäristön muutokset sekä pystyäkseen reagoimaan niihin parhaalla mahdollisella tavalla (Marcal & North, 2005). Proaktiivisuutta agentti tarvitsee pystyäkseen ohjaamaan oppilasta asetettuja päämääriä kohti. Reaktiivinen, autonominen ja ennakoiva käytös on edellytys sille että agentti voi tehdä päätöksiä itse. Mobiilisuus mahdollistaa agentin liikkumisen sen omassa toimintaympäristössään ja näin se saa haettua oppilaalleen tietoa esimerkiksi Internetistä. (Agarwal et al., 2004)

3. Agenttien roolit

Pedagogiset agentit ovat monenlaisissa rooleissa opetuksessa. Tässä esittelen, kuitenkin neljä agenttien pääroolia, mitkä ovat yleisesti käytössä oppimisympäristöissä. Ensimmäinen rooli on oppilaan henkilökohtainen avustaja (personalized assistant), joka on vuorovaikutuksessa oppilaan kanssa suoraan ja opastaa häntä täsmällisesti läpi opetettavien aihealueiden sekä tehtävien. Tällaiset agentit voivat myös toimia reaali maailman opettajan avustajina. (Sklar & Richards, 2006) Toinen agenttirooli on ekspertti, opettaja (expert agent). Opettajalla on enemmän tietoa kuin oppilaalla opetettavasta asiasta, se on alansa asiantuntija. (Baylor, 2003) Opettaja pyrkii opettamaan asiansa oppilaalle sekä neuvomaan häntä tarvittaessa. Opettajan rooli edellyttää neutraaliutta, eikä opettaja-agentti saa olla tunteikas oppilastaan kohtaan.

Kolmas tyypillinen agenttirooli on mentori (mentor), oppilaan neuvonantaja, ohjaaja, joka tekee oppilaan kanssa yhteistyötä. Mentorilla on yleensä enemmän tietämystä opiskeltavasta asiasta kuin oppilaalla itsellään. Mentori-agentti pyrkii johdattamaan oppilasta oikeaa ratkaisua kohti ja antamaan neuvoja tarvittaessa oppilaalle, paljastamatta kuitenkaan heti oikeaa vastausta. Mentori on tukena ja turvana oppilaan opiskellessa oppimisympäristössä. Mentori on hyvin samankaltainen kuin oppilas itse, ja sitä voidaankin verrata reaali maailmassa vanhempaan opiskelutoveriin. (Baylor, 2003; Sklar & Richards,

2006) Neljäs rooli on motivoija-agentti (motivator agent), havainnollistava, avoin, tunteensa näyttävä agentti (demonstrating agent). Motivoija tukee ja rohkaisee oppilasta opiskelussa eteenpäin sekä antaa oppilaalle kannustavaa palautetta. (Sklar & Richards, 2006) Motivoija-agentti on kuin reaali maailmassa oppilaan opiskelutoveri. Motivoija-agentti ei opeta oppilasta millään tavalla.

Älykkäitä agenteja on sekä nais- että mieshahmoja. Agenttien ulkoasu, sukupuoli, animaatio, käytös ja ääni vaikuttavat oppilaan kokemukseen sekä opiskelun tuloksiin agentin kanssa (Jaques, Lehmann, & Pesty, 2009). Myös oppilaan ikä ja tausta vaikuttavat siihen millaisia oppimistuloksia saadaan agenttien kanssa. Agentti saattaa olla pelkkä ääni tai käyttäjän eteen ponnahtava teksti-ikkuna, mutta ilman hahmoa oleviin agenteihin en puutu tässä tutkielmassa. Pohdin kohdassa 3.1 agentin eri roolien vaikutusta oppilaan oppimiseen ja kohdassa 3.2 ulkonäön ja sukupuolen vaikutusta oppilaan oppimiseen.

3.1. Roolien vaikutus oppimiseen

Oppilaat kokevat motivoivassa roolissa olevan agentin paremmaksi kuin ekspertin roolissa esiintyvän agentin. Motivoivan agentin kanssa oppilaat tuntevat olonsa mukavaksi ja jaksavat keskittyä paremmin opetettavaan aiheeseen. Näin ollen he saavat myös parempia oppimistuloksia kuin opettaja agentin kanssa. Motivoiva agentti osaa olla myös tunteikas ja on oppilaalle inhimillisempi vaihtoehto kuin pelkkä opettaja agentti. (Jaques et al., 2009) Oppilaat pitävät avustajan roolissa olevaa agenttia hyödyllisenä, kun on kyse esimerkiksi oppilaan muistuttamisesta jostakin tärkeästä tapahtumasta tai tehtävästä. Avustaja on myös tärkeä oppilaille etsittäessä tietoa Internetissä sekä sen antamat ehdotukset tehtävistä ja kokeista koetaan hyödylliseksi. (Schiaffino, Amandi, Gasparin, & Pimenta, 2008) Oppilaat pitävät mentoria, heidän neuvonantajansa miellyttävimpänä agenttiroolina. Mentori osaa opettaa ja auttaa oppilasta parhaiten, mutta myös motivoida ja antaa palautetta oppilaalle. (Baylor, 2003)

Baylor (2003) käytti tutkimuksessaan kolmea eri agenttia. Jokaisella agentilla oli oma roolinsa: mentori, ekspertti tai motivoija. Tutkimukseen osallistui 73 peruskoulun oppilasta. Oppilaat pitivät motivoivassa roolissa olevaa agenttia miellyttävämpänä ja ihmisläheisempänä sekä enemmän tukea antavimpana, kuin eksperttiä opettaja roolissa olevaa agenttia. Ekspertin roolissa olevaa agenttia pidettiin taas uskottavimpana ja parempana opettamaan. Eksperttiroolin uskottavuus johtuu osaksi myös agenttihahmon vaatetuksesta ja iästä. Agentti oli siististi puettu ja iältään vanhempi kuin motivoija- ja mentoriroolis- sa olevat agentit. Tutkimuksessa käytettyjen agenttien ulkomuodot on esitetty kuvassa 1. Oppilaat pitivät mentoria miellyttävämpänä ja auttavaisempana

kuin eksperttiä ja mentorilla saatiin merkittävästi parhaimpia tuloksia myös oppimisen suhteen.



Kuva 1. Tutkimuksessa käytetyt agentit: Motivoija, Ekspertti ja Mentori (Baylor, 2003).

Jos oppilaat saavat itse valita agentilleen roolin, he valitsevat mieluummin motivoivan opiskelutoverin kanssaan opiskelemaan kuin opettajaroolissa olevan agentin, jolla on ekspertin tiedot (Kim, Xu, Wei, & Ko, 2007). Agentin siis halutaan osoittavan tunteitaan oppilastaan kohtaan, sekä tukevan ja kannustavan oppilasta opiskelussaan eteenpäin. Agentin taidolla opettaa ja tiedon määrällä ei ole niinkään väliä oppilaiden oppimistuloksien kanssa. Oppimisympäristö, missä on jokin agentti oppilaan apuna, antaa kuitenkin aina parempia tuloksia kuin oppimisympäristö, jossa oppilas opiskelee yksin (Jaques et al., 2009).

3.2. Agentin sukupuolen ja ulkoasun vaikutus oppimiseen

Mies- ja naisagentit herättävät oppilaiden parissa erilaisia tuntemuksia. Opiskeltaessa osa oppilaista kokee kanssaan samaa sukupuolta olevan agentin miellekkäämmäksi ja motivoivammaksi kuin toisen sukupuolen edustajan. (Jaques et al., 2009; Rosenberg-Kima, Baylor, Plant, & Doerr, 2008) Ainoastaan muutamissa tutkimuksissa käy ilmi, että naispuolinen agentti olisi parempi kuin miespuolinen. Kim ja muut (2007) sai tutkimustuloksissaan merkitseviä eroavia tuloksia sukupuolen vaikutuksesta oppilaisiin. Heidän tutkimukseensa osallistui 58 tyttöä lukion matematiikan kurssilta. Tyttöjen asenne matematiikkaa kohtaan oli erittäin kielteinen ja opiskelumotivaatio alhainen. Osa tytöistä sai itse valita neljästä agenttihahmosta itselleen sopivalta tuntuvan vaihtoehdon. Vaihtoehtoina olivat nais- ja miesopettaja, sekä nais- ja miestoveri agentti. Suurin osa tytöistä valitsi naisen hahmossa olevan agentin. Niiden tyttöjen, jotka saivat itse valita agenttinsa, asenne ja työskentelytehokkuus matematiikan parissa paranivat. Osalle tytöistä määrättiin jokin neljästä agenttihahmosta. Tällä ei huomattu olevan vaikutusta asenteisiin eikä työskentelytehokkuuteen. Tytöt

pitivät naishahmossa olevia agenteja merkittävästi parempina lisäämään heidän motivaatiotaan ja työskentelyään matematiikan parissa, mutta miehen hahmossa olevat agentit jopa laskivat oppilaiden motivaatiota ja työskentelytehokkuutta. Tuloksissa täytyy kuitenkin huomioda, että siinä osassa tyttöjä, jotka saivat itse valita agenttinsa, oli mies agenttihahmojen käyttö merkittävästi vähäisempää kuin naispuolisten agenttien. Voidaan myös olettaa, että tuloksiin vaikuttaa oppilaiden homogeenisuus sekä matematiikka tieteenalana, jota kohtaan miespuoliset opiskelijat osoittavat yleensä enemmän kiinnostusta kuin naiset.

Agenttien ulkoasut ja animaatiohahmot herättävät oppilaissa ristiriitaisia tunteita. Usein halutaan agentin olevan ihmishahmossa enemmän kuin, esimerkiksi jokin eläinhahmo tai satuolento. Ihmishahmossa agentit ovat helposti lähestyttäviä ja ystävällisen näköisiä (Veletsianos, Yerasimou, & Doering, 2005). Schiaffino ja muut (2008) antoivat opettajaroolin omaavasta taikuri agenttihahmosta (ks. kuva 2) negatiivista palautetta. Viidennes oppilaista koki agentin toiminnot erittäin tärkeiksi, mutta olisivat halunneet jonkin muun hahmon opettamaan heitä. Agentti ilman minkäänlaista hahmoakin olisi ollut heistä parempi kuin taikuri.



Kuva 2. Tutkimuksessa käytetty agenttihahmo eTeacher (Schiaffino et al., 2008).

Jaques ja muut (2009) päätyivät tulokseen, että ihmismäistä naisagentti hahmoa pidettiin erittäin miellyttävänä. Agentti oli nimeltään Pat (ks. kuva 3). Patin vaatteet ja niiden värit miellyttivät oppilaita, ja Pat hahmoa pidettiin tavallisen ja uskottavan näköisenä.



Kuva 3. Tutkimuksessa käytetty agentti Pat (Jaques et al., 2009).

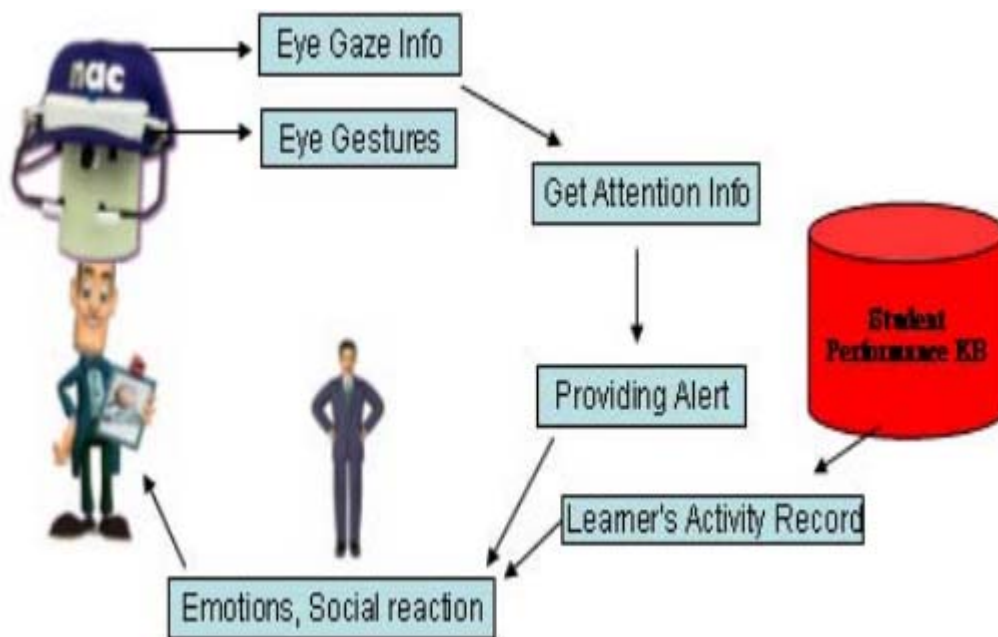
Tuloksia tutkittaessa täytyy kuitenkin muistaa, että oppilaiden ikä, tausta ja omat mieltymykset vaikuttavat siihen, kuinka he kokevat erilaiset agenttihahmot.

4. Miten tunteisiin vetoavat agentit vaikuttavat oppilaaseen?

Tunteita on tutkittu psykologian, fysiologian ja tekoälyn aloilla paljon. Emotionaaliset mallit auttavat meitä simuloimaan tunteita ja erilaisissa tapauksissa vastaamaan tunteisiin sekä auttavat toteuttamaan agenteilla ihmisten tunteita. Ihmisen perustunteet ovat ilo, pelko, suru ja viha (Ghasem-Aghee, Somayeh, & Ören, 2008). Nämä neljä perustunnetta kuitenkin voivat sekoittua ja voimme kokea myös esimerkiksi inhoa, hämmästyä, ahdistusta, innokkuutta, kateutta ja kiintymystä. Tunteet ovat kaikenlaisissa vuorovaikutustilanteissa läsnä, joten niitä on myös hyödyllistä sisällyttää virtuaalisiin oppimisympäristöihin (Ben Ammar & Neji, 2006). Agentit ovat ihmisten kaltaisia persoonallisuuksia tai animaatiohahmoja, jotka toimivat oppilaan kanssa luonnollisella tavalla ja tarjoavat todentuntuksia oppimiskokemuksia oppilaalle. Edellä on jo todettu, että oppilaat ovat pitäneet eniten tunteensa avoimesti näyttävästä, tukea antavasta ja motivoivasta agenttihahmosta. Tällaisella hahmolla on myös saatu parhaat oppimistulokset.

Jotkin älykkäät agentit osaavat lukea oppilaan kasvoilta joitakin perustunteita ja näin pystyvät motivoimaan oppilastaan ja antamaan hänelle palautetta. Osa agenteista voi reagoida oppilaan katseeseen, hiiren tai näppäimistön käyttöön ja saada tätä kautta selville oppilaan tunnetilan. Agentit myös tallentavat oppilaalta jo aikaisemmin saatuja tietoja. Yhdistelemällä vanhaa ja uutta tietoa he saavat oppilaasta perustoimintamallin ja osaavat reagoida oikein oppilaan käytökseen. Agentti voi muistuttaa oppilasta keskittymään aiheeseen, jos hän katselee näytöstä pois päin kauan. Agentti huomaa oppilaan kiinnostuksen vähenemisen aiheita kohden, kun oppilaan pupillit laajenevat tai silmänliikkeet

muuttuvat vähemmän aktiivisiksi (ks. kuva 4). Silloin agentin on reagoitava ja kyseltävä oppilaalta, onko hän kyllästynyt tai väsynyt. Väsymykseen ja ikävystymiseen viittaa myös vähentynyt näppäimistön käyttö ja vähentyneet klikkaukset hiirellä. Tällöinkin agentin on reagoitava ja vaihdetta esimerkiksi aihetta. Kun taas oppilas osoittaa mielenkiintoa opetettavaa aihetta kohti, agentti antaa oppilaalle positiivista palautetta. Agentti osaa myös näyttää oppilaalle tunteitaan, esimerkiksi kun oppilas menettää keskittymisen, agentti saattaa osoittaa hieman vihastuneensa tai antaa hälytysmerkin oppilaalle. Oppilas voi saada myös empatiaa agentilta, kun hän vastaa väärin. (Wang, Chignell, & Ishizuka, 2006) Agentit pystyvät ilmaisemaan tunteitaan ja mielialojaan, esittämään opittuja taitoja ja käyttämään ei-verbaalisia vihjeitä kiinnittääkseen oppilaan huomion opiskeltavaan asiaan (Ben Ammar & Neji, 2006). Katseenseurantaa hyödyntävillä ja kasvojen ilmeitä lukevilla agenteilla on saatu positiivisia oppimistuloksia, ja niiden on todettu auttavan oppilasta keskittymään paremmin (Ben Ammar & Neji, 2006; Wang et al., 2006).



Kuva 4. Katseenseurannassa oppilaalta saatu tieto silmien liikkeistä yhdistyy ja tallennettuun tietoon ja käyttötilanteeseen, mihin agentti sitten reagoi (Wang et al., 2006).

Jaques, Lehmann ja Pesty (2009) esittelivät edellä mainitun Pat-agentin. Heidän tutkimukseensa osallistui 39 oppilasta, jotka olivat iältään 12–19 -vuotiaita. Pat oli kyvykäs tunnistamaan tunteita kuten iloa, surua, vihaa, pettymystä, häpeää, kiitollisuutta ja tyytyväisyyttä. Patin tunteiden ilmaisemiseen sekä tunteiden riittävyteen ja kommunikointiin ei kuitenkaan oltu täysin tyytyväisiä, vaan suurimmasta osasta oppilaista se oli keskitasoa. Oppilaat olivat tyytyväi-

siä Patin käytökseen, motivointiin, liikkeisiin, reagointiin ja siihen, että Pat osasi muuttaa asennettaan, sekä keinojaan ilmaista tunteitaan. Oppilaiden mielestä vain pelkkä ääni ei riittäisi tunteiden ilmaisemiseen. Kuvassa 5 on esimerkki Patin antamasta kannustuksesta.



Kuva 5. Pat motivoi oppilasta jatkamaan vaikeuksista huolimatta ja antaa palautetta onnistuneesta suorituksesta (Jaques & et al., 2009).

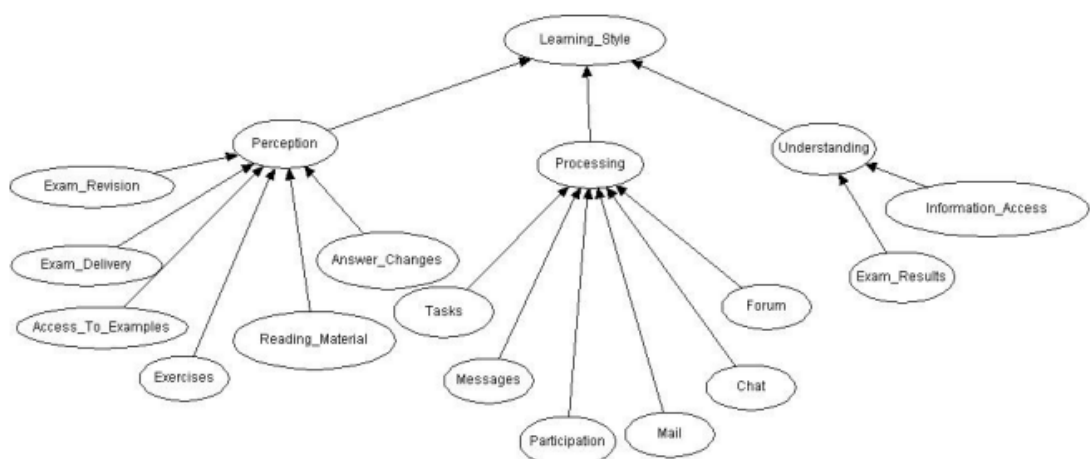
Tunteisiin vetoavien agenttien kanssa opiskelu koetaan mielekkäimmäksi, koska ne ovat eniten ihmismäisiä. Miten tekoälyllinen animaatiohahmo voi olla samankaltainen kuin oikea ihminen? Tämä kysymys herättää ja lisää oppilaiden kiinnostusta työskennellä tunteisiin vetoavien agenttien kanssa. Agentti, joka tunnistaa ilmeesi, rekisteröi silmänliikkeesi ja näiden perusteella kannustaa sekä auttaa sinua opiskelussa eteenpäin on oppilaalle kuin hyvä opiskelutoveri reaali maailmassa.

5. Miten agentin personoitu opetus koetaan

Ihmisillä on erilaisia oppimistyyplejä, joiden avulla he oppivat ja omaksuvat opetettavat asiat. Päästäkseen parhaisiin oppimistuloksiin heidän on siis saatava juuri heidän oppimistyylilleen sopivaa opetusta. Oppimistyylyt voidaan jakaa kahdeksaan eri kategoriaan yleisesti käytetyn Felderin ja Silvermanin (1988) kehittämän mallin mukaan, mutta suurin osa oppilaista ei omaa vain yhtä oppimistyyliä, vaan heidän oppimisessaan on osia monista eri oppimistyyleistä. (Reategui & Zattera, 2008; Schiffino, Amandi, Gasparini, & Pimenta, 2008) Aktiivisen oppimistyylin omaavat oppilaat oppivat harjoituksia aktiivisesti tekemällä, keskustelemalla ja opettamalla toisilleen asioita, kun taas mieteliäiden oppilaiden on saatava rauhassa pohtia ja ajatella asioita. Visuaaliset

oppilaat oppivat parhaiten näköaistinsa avulla, kun taas verbaliset oppilaat oppivat parhaiten puhumalla ja kirjoittamalla. Intuitiiviset oppilaat oppivat välittömästi, muodostamalla uusia mielleyhtymiä vanhojen kanssa. Sensitiiviset oppilaat eivät opi helposti uusia asiakokonaisuuksia, ja he perustavat oppinsa vain faktoihin. Järjestyttä noudattavat oppilaat oppivat lineaarisia ja loogisia yhteyksiä parhaiten, kun taas globaalit oppijat omaksuvat asioita satunnaisesti sieltä täältä kunnes yhtäkkiä ymmärtävät asiakokonaisuuden. (Reategui & Zattera, 2008; Schifffino et al., 2008) Agentti seuraa itse oppilasta ja muodostaa päätelmiä tämän mallin avulla siitä, miten oppilaat vastaanottavat ja käsittelevät tietoa.

Miten agentti voi sitten valita käyttäjälleen sopivan oppimistyylin? Agentti selvittää riippuvuudet oppilaan käytöksen ja oppimistyylin välillä Bayes-verkon avulla (ks. kuva 6). Felder ja Silverman malliin kuuluu vielä oppimistyylien lisäksi neljä eri näkökantaa oppimistyyleistä: syötteisiin liittyvät (input), havaintokykyyn liittyvät (perception), prosessointiin liittyvät (processing) ja ymmärrykseen liittyvät (understanding). Esimerkiksi havaintokyky määrittää sitä, millaista tietoa oppilas haluaa vastaanottaa. Prosessoinnissa käydään läpi, miten oppilas työstää tietoa, tekemällä tehtäviä, keskustelemalla vai kirjoittamalla muistiinpanoja. Ymmärrys määrittelee sitä, miten oppilas oppii asian, välittömästi vai esimerkiksi vähitellen. Syötteet kertovat siitä, mitä oppilas esimerkiksi antaa vastaukseksi johonkin tehtävään. Oppimistyylistä riippuen agentti neuvoo ja tekee ehdotuksia oppilaalle, jotta opiskelu etenisi parhaalla mahdollisella tavalla.

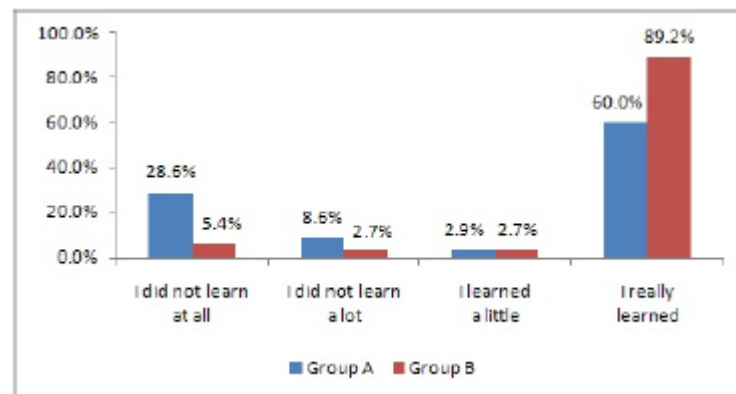


Kuva 6. Bayes-verkko, jota eTeacher-agentti käyttää (Schiaffino et al., 2008).

Schiaffino ja muut (2008) esittelee, miten oppilaat kokivat agentin, joka antaa heille persoonallista opetusta. Suurin osa oppilaista koki agentin antamat

neuvot ja ehdotukset hyödyllisenä itselleen ja hieman yli puolet kokivat agentin antamat muistutukset tehtävistä ja tapahtumista tarpeellisina. Oppilaat olivat kaiken kaikkiaan hyvin tyytyväisiä eTeacher agenttiin. Agentti oppii ajan myötä aina enemmän asioita oppilaastaan, ja mitä kauemmin sillä on aikaa tarkkaila oppilasta, sitä paremmin se pystyy auttamaan ja opettamaan oppilasta.

Myös Reategui ja Zattera (2008) huomasivat, että personoidun agentin kanssa työskentelevät oppilaat oppivat paremmin kuin ilman agenttia työskentelevät oppilaat. Heidän tutkimukseensa osallistui 72 yliopistossa opiskelevaa nuorta. Opiskelijat arvioivat itse oppimistuloksiaan. Kuvasta 7 näkyy, kuinka ilman agenttia työskennellyt ryhmä A arvioi oppimistuloksensa selvästi huonommaksi kuin agentin kanssa työskennellyt ryhmä B.



Kuva 7. Oppimistulokset agentin kanssa ja ilman agenttia (Reategui & Zattera, 2008).

Reategui ja Zattera (2008) testasivat myös, miten eri oppimistyyliä vaikuttivat siihen, minkälaisia oppimistuloksia saatiin agentin kanssa ja ilman agenttia. Kuvassa 8 esitetään tästä tulokset, oppilaat ovat itse arvioineet omaa oppimistaan.

Pelkästään näiden kahden edellä mainitun tutkimuksen perusteella ei voida vielä päätellä, osaako agentti todella antaa oppilaille persoonallista ja juuri heidän oppimistyyliilleen sopivaa opetusta. Reateguin ja Zatteran (2008) tutkimukseen osallistui 72 opiskelijaa, joka on pieni joukko tutkittaessa agenteja, jotka pyrkivät antamaan oppilaille persoonallista opetusta oppimistyylien mukaan. Intuiitiivisesti ja verbaalisesti oppivia opiskelijoita on vain kaksi kummasakin A ryhmässä, joissa opiskelija työskentelee yksin. Verbaalisessa ja intuitiivisessa B ryhmässä ei työskentele ketään agentin kanssa, joten tutkimus ei pysty osoittamaan, että agentit palvelisivat jokaista oppimistyyliä yhtä hyvin. Tutkimus antaa kuitenkin suuntaa päätelmälle, että agentin käyttö palvelee varsinkin aktiivisia, sensitiivisiä ja visuaalisia opiskelijoita hyvin.

	Group	I did not learn	I did learn	Total
<i>Active</i>	A	7 (70%)	3 (30%)	10
	B	0	13 (100%)	13
<i>Reflective</i>	A	0	3 (100%)	3
	B	2 (33.3%)	4 (66.7%)	6
<i>Sensing</i>	A	5 (38.5%)	8 (61.5%)	13
	B	2 (8.7%)	21 (91.3%)	23
<i>Intuitive</i>	A	0	2 (100%)	2
	B	0	0	0
<i>Visual</i>	A	6 (40%)	9 (60%)	15
	B	0	18 (100%)	18
<i>Verbal</i>	A	1 (50%)	1 (50%)	2
	B	0	0	0
<i>Sequential</i>	A	5 (45.5%)	6 (54.5%)	11
	B	0	9 (100%)	9
<i>Global</i>	A	0	3 (100%)	3
	B	0	2 (100%)	2

Kuva 8. Oppimistulokset oppimistyylin mukaan (Reategui & Zattera, 2008).

Agentit pystyvät luomaan paljon erilaisia harjoituksia aktiivisille oppilaille ja antamaan helposti visuaalista opetusta virtuaalisessa oppimisympäristössä. Sensitiivisten oppilaiden hyvät tulokset johtuvat agentin sinnikkyydestä opettaa heitä faktoihin perustuen väsymättä. Sensitiiviset oppilaat saavat tarvitsemaansa aikaa oppia asioita agentilta enemmän kuin reaali maailmassa opettaja voisi heille antaa. Agentin antama yksilöllinen opetus ja sen väsymätön työskentely oppilaan kanssa, parantavat ainakin joidenkin oppilaiden oppimiskokemuksia sekä oppimistuloksia.

6. Lopuksi

Oppilaiden motivaatio ja keskittyminen saadaan paremmin pidettyä yllä älykkäiden, pedagogisten agenttien avulla kuin ilman niitä virtuaalisissa oppimisympäristöissä. Agenttitekniologia voi auttaa löytämään ratkaisuja oppilaille, joille on todettu keskittymishäiriöitä. Pedagogiset agentit ovat olleet myös mukana autististen lasten kuntoutuksessa. Nämä agentit ovat monella tapaa hyödyllisiä, mutta eivät vielä voi korvata kokonaan reaali maailman opettajaa oppilaalle. Agentit ovat kuitenkin todella hyvä lisä virtuaalisissa oppimisympäristöissä.

Tässä tutkielmassa pohdittiin eri näkökulmista, mitkä asiat vaikuttavat oppilaiden kokemuksiin ja oppimistuloksiin agenttien kanssa. Tutkielmaa voisi

laajentaa vertailemalla, miten erityisen ja yleisen tiedon opettaminen agenteilla toimii. Tutkielmaa saisi myös laajennettua paneutumalla syvällisemmin persoonallisiin ja tunteisiin vetoaviin agentteihin sekä niiden eri toteutustekniikoihin opettaa oppilasta.

Agenttien erilaiset roolit ja toiminnot mahdollistavat oppilaalle reaali maailman opetuksen samankaltaisuuden sekä persoonallisen opetuksen. Agenttien tekoälyn kehittyessä selviää, voivatko agentit jopa korvata oikean reaali maailman opettajan.

Viiteluettelo

- Agarwal, R., Deo, A. & Das, S. (2004). Intelligent agents in e-learning. *SIGSOFT Software Engineering Notes*, 29(2), 1-3.
- Baylor, A. (2003). The impact of three pedagogical agent roles. *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems*, ACM Press, 928-929.
- Ben Ammar, M. & Neji, M. (2006). A multi-agent based system for affective peer-e-learning. *Proceedings of the MobiMedia 2nd International Conference on Mobile Multimedia*, ACM Press, 1-6.
- Blanchard, E. & Frasson, C. (2004). An autonomy-oriented system design for enhancement of learner's motivation in e-learning. *Intelligent Tutoring Systems, Lecture Notes in Computer Science 3220*, Springer, 34-44.
- Buendia, F., Cano, J. & Benlloch, J. (2009). An Instructional approach to drive computer science courses through virtual learning environments. *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*, ACM Press, 6-10.
- Chen, K.J. & Barthés, J.P. (2007). MemoPA: Intelligent personal assistant agents with a case memory mechanism. *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence Lecture Notes in Computer Science 4682*, Springer, 1357- 1367.
- Ghasem-Aghaee, N., Fatahi, S. & Ören, T.I. (2008). Agents with personality and emotional filters for an e-learning environment. *Proceedings of the Simulation Multiconference*, ACM Press, 816-821.
- Haynes, S., Cohen, M. & Ritter, F. (2009). Designs for explaining intelligent agents. *Internal Journal of Human-Computer Studies*, 67(1), 90-110.
- Jaques, P., Lehmann, M. & Pesty, S. (2009). Evaluating the affective tactics of an emotional pedagogical agent. *Proceedings of the 2009 ACM Symposium on Applied Computing*, ACM Press, 104-250.
- Kim, Y., Xu, B., Wei, Q. & Ko, Y. (2007) Creating social affable learning environments. *Technology and Teacher Education Annual*, 3277-3284.

- Macal, C. & North, M. (2005). Tutorial on agent-based modeling and simulation. *Proceedings of the 2005 Winter Simulation Conference*, ACM Press, 2-15.
- Reategui, E. & Zattera, C. (2008). Do learning styles influence the way students perceive interface agents? *Proceedings of the IHC VIII Brazilian Symposium on Human Factors in Computing Systems*, ACM Press, 108-116.
- Rosenberg-Kima, R., Baylor, A., Plant, E. & Doerr, C. (2008). Interface agents as social models: The impact of appearance on females' attitude toward engineering. *Computers in Human Behavior*, 25(6), 526-531.
- Schiaffino, S., Amandi, A., Gasparin, I. & Pimenta, M. (2008). Personalization in e-learning: the adaptive system vs. the intelligent agent approaches. *Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems*, ACM Press, 186-195.
- Sklar, E. & Richards, D. (2006). The use of agents in human learning systems. *Proceedings of the AAMAS Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, ACM Press, 767-774.
- Veletsianos, G., Yerasimou, T. & Doering, A. (2005). The role of intelligent agents on learner performance. *Proceedings ED-MEDIA 2005—World Conference on Educational Multimedia, Hypermedia & Telecommunications*, 3638-3643.
- Wang, H., Chignell, M. & Ishizuka, M. (2006). Empathic tutoring software agents using real-time eye tracking. *Proceedings of the ETRA 2006 Symposium on Eye Tracking Research & Applications*, ACM Press, 73-78.

Ohjaaja: Pentti Hietala

Sähköisen vaalijärjestelmän kokeilusta kunnallisvaaleissa 2008 äänestäjän tietoturvan näkökulmasta

Jorma Laajamäki

Tiivistelmä.

Suomessa toteutettiin sähköisen äänestyksen pilottihanke vuoden 2008 kunnallisvaaleissa. Tässä tutkielmassa on tarkoitus selvittää pilotin toteutumista tietojärjestelmänä painottaen tietoturvan näkökulmaa. Ensin paneudutaan järjestelmän määrittelyn ongelmaan ristiriitaisten oletuksien alla, minkä jälkeen selvitetään varsinaista tiedonkulkua järjestelmän sisällä tietovirtojen avulla. Vertaamalla järjestelmän tiedonkulusta saatua informaatiota järjestelmän käyttäjävaatimuksiin ja tavoitetilään saadaan aikaan rajattu innovaationarviointi sähköisen äänestyksen pilotoinnista Suomessa.

Avainsanat ja -sanonnat: Tietoturva, vaalit, sähköinen äänestäminen

CR-luokat: J.1, K.4.1, K.5.2

1. Johdanto

Edustuksellisessa demokratiassa vaalit ovat ainoa tunnettu järjestelmä, joka siirtää legitimiä vallan eduskuntalaitoksessa toimiville edustajille kansalta, jolle se määrittelyn mukaan demokratiassa kuuluu. Vaalit ovat laaja kokonaisuus demokraattisissa maissa ja se käsittää suurimman osan kansalaisista. Vaalissa äänten antamiseen ja laskentaan liittyvä tietojärjestelmä on ainakin yhtä vanha kuin antiikin Kreikan demokratia, joidenkin tutkimusten mukaan vieläkin vanhempi. Esimerkiksi muinaisen Intian demokraattisessa perinteessä 600-luvulla ennen ajanlaskun alkua järjestettiin vaaleja [Muhlberger, 1998]. Nykykaisen muotonsa vaalit saivat 1700- ja 1800-lukujen aikana [Anttiroiko, 2003].

Vaaleihin predikoidaan demokraattisen järjestelmän toimivuus [Euroopan neuvosto, 2004]. Vaalijärjestelmän toiminta on yhteiskunnallisessa mielessä vähintään yhtä tärkeää kuin yhdenkään toisen järjestelmän. Jos esimerkiksi lentokoneessa toimivan epäonnistuneen järjestelmän seurauksena menehtyisi tuhansia ihmisiä, se ei sinällään olisi uhka yhteiskunnalle, vaikka olisi kriittinen lentomatkatustajille ja muille järjestelmään liittyville. Vaalijärjestelmän toimimattomuus tai alttius väärinkäytökselle voi teoriassa kaataa koko yhteiskuntajärjestelmän.

Useissa Euroopan maissa sekä Yhdysvalloissa on viimeksi kuluneen vuosikymmenen aikana siirrytty joko kokeilemaan tai käyttämään sähköistä äänestystä, joissakin – kuten Alankomaissa – jo vuosikymmenen aiemmin. Suomessa

kokeilu sähköisestä äänestyksestä järjestettiin vuoden 2008 kunnallisvaaleissa kolmella paikkakunnalla ja vaalit päädyttiin uusimaan sähköiseen äänestyksen pilotoineiden kuntien osalta. Vaalit uusittiin kuitenkin käyttöliittymän oletetun toimimattomuuden perusteella [KHO 2009], mikä ei ole tämän tutkimuksen tarkastelun piirissä.

Tämä tutkimus rajautuu sähköisen ja sille vertailuna toimivan manuaalisen eli perinteisen lippuvaali äänestysjärjestelmien tiedon ja sen takaisinkytkentöjen analysointiin pääasiallisesti äänestäjän näkökulmasta. Yhteiskunnallisesta näkökulmasta äänestäjän tarpeisiin kiteytyy koko järjestelmän toimivuus, sillä vaalijärjestelmän olemassaolon syy on äänestäjien oikeus ja velvollisuus luovuttaa päätäntävalta kansanedustuslaitokselle.

Tutkielma etenee kaksijakoisesti. Ensimmäisessä osiossa pohditaan vaalijärjestelmien ontologiaa ja käsitellään kriittisesti niitä muutospaineita, joita vaalijärjestelmään, erityisesti Suomen sähköisen järjestelmän pilotoinnissa, on kohdistunut. Näin pyritään löytämään sen kaltainen tila ja tilan täyttymiset ehdot, jotka voitaisiin tulkita yleisiksi ja päteviksi säännöiksi. Lisäksi tutkimuksessa määrittelen oman käsitykseni tietoturvasta, joka tietojenkäsittelytieteellisessä tutkimuksessa on usein hyvin pluralistinen termi ja sellaisena tieteellisesti hankala viitattava.

Ideaalitalan saavuttamista tutkitaan toisessa osiossa analysoimalla tietovirtoja, jotka järjestelmässä kuvaavat tiedon liikkumista ja takaisinkytkentöjä. Sekä sähköinen että manuaalinen vaalijärjestelmä käydään läpi käyttäjävaatimusten kautta, ja näiden yhtäläisyyksien ja eroavaisuuksien vertailulla pyritään selvittämään sähköisen äänestyskokeilun onnistumista tietojärjestelmänä.

2. Vaalijärjestelmä

Vaalijärjestelmä on monitahoinen tietojärjestelmä. Jos sitä pohditaan pelkän mekaanisen toiminnan kannalta, on se hyvin yksinkertainen toiminnoiltaan. Esimerkiksi tasavallan presidentin vaalien toinen kierros: kerätään dataa eli äänet, jotka tallennetaan tietokantaan eli vaaliurna. Datalla voi olla vain kolme eri arvoa: kahden eri ehdokkaan äänet ja hylätyt eli kummallekaan kuulumattomat äänet. Arvoista käytännössä vain kaksi eli ehdokkaiden äänet ovat ensisijaisen tuloksen kannalta merkityksellisiä. Tietokannasta eli vaaliurnasta johdetaan yksinkertaisella laskuperiaatteella tietoa eli kummankin ehdokkaan äänten summat ja katsotaan, kumpi ehdokkaista sai enemmän ääniä eli tuli valituksi.

Toisaalta vaalijärjestelmän on erittäin kriittinen [McGaley 2008], vaikka se ei suoranaisesti aiheuta uhkaa esimerkiksi ihmisten terveydelle tai hyvinvoinnille. Annettujen tehtävien monimutkaisuus ja kattavuus tekee demokraattisesta vaalijärjestelmästä monimutkaisen. Sen tulee kattaa kaikki sille esimerkiksi perus-

tuslaissa [L731, 1999] ja vaalilaissa [L714, 1998] annetut velvollisuudet vaaleista ja äänestämisestä, joista osa on hyvin abstrakteja kuten "vaalien reiluus" ja osa hyvin kattavia kuten "yleinen ja yhtäläinen äänioikeus" [Jääskeläinen 2002].

Vaalijärjestelmän tehtävä on siis laajempi kuin pelkkä äänien antaminen ja laskenta. Johtuen demokraattisen päätöksenteon ja vastuullisuuden valvontaan liittyvistä ominaisuuksista, kuten äänen uudelleen laskemisen mahdollisuudesta tai oikeudesta valittaa vaalien oikeasta järjestyksestä, tulee järjestelmän toteuttaa demokraattista ideaa. Ei riitä, että vaalijärjestelmä toimii, sen tulee myös kyetä rakenteellaan ja toiminnallaan todistamaan, että äänet on annettu niiden sääntöjen puitteissa, jotka demokraattiselle äänestysjärjestelmälle on annettu [Gritzalis, 2002].

2.1. Järjestelmän määrittelyn ja tavoitetilan ongelma

Järjestelmän määrittely ja tavoitetila nousee ongelmaksi eri lähteiden ristiriitaisuuksista johtuen. Suomen sähköisen äänestysjärjestelmän toteuttaja Tieto- ja viestintäministeriön Oikeusministeriön hyväksymän järjestelmän esimäärittelyksen [Huitti et al., 2003], joka ei kaikilta osin vastaa perinteistä näkemystä demokraattisen vaalijärjestelmän vaatimuksista. Esimäärittelyksen liitteessä tarkastellaan eri kriteerien kriittisyyttä, joista vain yhtäläinen äänioikeus saa korkeimman kriteerin eli yksi asteikolla yhdestä viiteen. Helsingin hallinto-oikeus yritti linjata uutta näkökulmaa esittäessään hieman vajaan 98 % onnistumisen äänestämisessä olevan riittävä [HHO, 2009]. Koska tämä selvästi rikkoo yleistä ja yhtäläistä äänioikeutta, joka osaltaan on yksi vaalijärjestelmän perustavista periaatteista, on kyseessä ilmeinen virhe. Vaikka joissakin lähteissä tämä reilu 2 % osuus on rinnastettu vaaleissa hylättyihin ääniin, ei kysymys ole tästä. Manuaalisessa vaalijärjestelmässä nämä äänet olisivat niitä, jotka katoavat äänestyskopin ja vaaliuurnan välillä. Tämänkaltaista äänien katoamista ei ole osoitettu tapahtuneen yksissäkään aiemmissa vaaleissa.

Perinteinen oletus vaalijärjestelmän virhemarginaalista on ollut lähellä nolaa. Yleistä ja yhtäläistä äänioikeutta on haasteellista selittää siten, että olisi jonkinlainen hyväksytty määrä kadotettuja ääniä. Tätä käsitystä tukee myös hallituksen esitys vaalilain muuttamisesta [HE 14/2006] sähköisen äänestämisen vuoksi. Sähköiselle äänestykselle ei sinällään ole annettu erilaista määritelmää tai virhemarginaalia kuin perinteiselle äänestykselle. Euroopan neuvoston [2004] suositus sähköisen äänestyksen laillisista, toiminnallisista ja teknisistä normeista toteaa, että "Sähköisen äänestyksen tulee olla yhtä luotettavaa ja turvallista kuin äänestys sellaisissa demokraattisissa vaaleissa tai kansanäänestyksissä, joissa sähköisiä äänestysmahdollisuuksia ei käytetä."

Demokraattisille äänestysjärjestelmille on olemassa periaatteet, jotka muun muassa McGaley [2008] on kategorisoinut:

1. Yleisyys; kaikilla täysivaltaisilla kansalaisilla on oikeus äänestää.
2. Yhtäläisyys; jokaisella äänestäjälle on sama määrä ääniä.
3. Vapaus; äänestäjällä on oikeus painostamattomaan äänestämiseen.
4. Salaisuus; jokaisella äänestäjällä on oikeus ja velvollisuus antaa äänensä salaisesti niin, ettei toinen ihminen voi todeta annettua ääntä.
5. Välittömyys; äänestyksen tulos pitää johtaa suoraan äänestäjien antamista äänistä.
6. Luottamus; äänestäjien tulee luottaa näiden periaatteiden noudattamiseen.

2.2. Periaatteista johdetut käyttäjävaatimukset

Grizalis [2002] on induktoinut yhtenäistetyn ohjelmistokehitysprosessin (IBM Rational Unified process) keinoin sähköisen vaalijärjestelmän järjestelmävaatimusten ja äänestäjien käyttäjävaatimusten listan, jota käytän pohjana sekä analyysin kysymyksille että järjestelmän tavoitetilalle. Grizaloksen oletamat periaatteet järjestelmän vaatimuksille ovat samat sekä Suomen Oikeusministeriön [Jääskeläinen, 2002] että aiemmin mainitun McGaleyn näkemyksen kanssa. Grizaloksen käyttäjävaatimukset ovat seuraavat:

1. Osallistumisen äänestämiseen tulee olla todennettavissa.
2. Pakottamattomuus pitää olla taattua.
3. Mahdollisuus antaa kelvoton ääni tarkoituksella pitää olla mahdollista.
4. Vain äänioikeutettujen pitää pystyä äänestämään ja vain kerran.
5. Annettua ääntä ei saa pystyä muuttamaan tai toistamaan.
6. Äänestäjän tulee kyetä todentamaan että annettu ääni on laskettu mukaan lopputulokseen.
7. Äänestäjillä tulee olla syrjimätön mahdollisuus käyttää äänestysjärjestelmää.
8. Äänioikeus- ja äänestysjärjestelmien tulee olla todistettavasti eriytettyjä.
9. Äänet tulee varmentaa erillään äänestäjien varmentamisesta.
10. Äänestyksessä ei saa käyttää välikäsiä, ts. toinen ihminen ei voi päättää toisen ihmisen äänestä.
11. Jokainen äänestyslipuke tulee tallentaa ja laskea oikein.

3. Tietoturva ja tietovirrat

Tietoturva on käsitteenä hankala johtuen termin moniselitteisyydestä. Tietoturvalle ei löydy yhtenäistä määritelmää tietojenkäsittelytieteen piiristä, vaan useimmiten se perustuu joko kontekstiin tai se määritellään erikseen. Tässä tut-

kimuksessa määrittelen tietoturvan itsenäisesti ja palautan sen Platonin klassisen tietokäsityksen kautta ymmärretyn tiedon turvaksi – tiedon turva tarkoittaa tiedon oikeellisuuden ehtojen turvaamista. Klassisen tiedon määritelmä voidaan palauttaa kolmeen ehtoon: tiedon haltija uskoo tiedon olevan tosi, tieto on tosi ja tiedon totuusarvo on perusteltavissa.

Turvalla käsitetään ne asiat, jotka edesauttavat säilyttämään edellä mainittuja ehtoja, eli joko tiedon säilymistä, uskon säilymistä tiedon oikeellisuuteen ja perusteltavuuden säilymistä.

Tätä tietoturva-käsitystä vasten peilataan äänestysjärjestelmän tietovirtoja niiden kulkiessa vaalijärjestelmän läpi eri toimijoiden näkökulmista. On huomattava, että vaikka tieto olisikin tosi, jos ulkopuolisella tarkkailijalla kuten julkisella yleisöllä ole tästä tietoa, se ei ole perusteltavissa eli se on uskoa, ei totta.

Tieto voidaan esittää kolmen totuusarvon kautta: se on totta eli tietoa vain, jos sen kaikki kumuloituvat muuttujat ovat totta koko ketjun ajan. Koska tieto on klassisen tietokäsityksen mukaan tietoa vain, jos se täyttää kolme ehtoaan – totuus-arvo, usko ja perusteltavuus – voidaan päätellä, että jokaisen ketjun lenkin jokaisen ehdon tulee olla tosi, muuten ketjun päässä ei ole tietoa vaan jotakin muuta. Näin tiedon epätotuuteen ei tarvita kuin yksi epätosi ehto yhdessä sen käsittelyn vaiheessa, jotta lakkaa olemasta tietoa.

Tietovirrat toimivat työkaluna tulkittaessa tiedon leviämistä. Niistä voidaan johtaa kahdenlaista tietoa: onko tieto saavuttanut jonkin toimijan vai ei, ja onko sen totuusarvo tai perusteltavuus jostain syystä muuttunut matkalla.

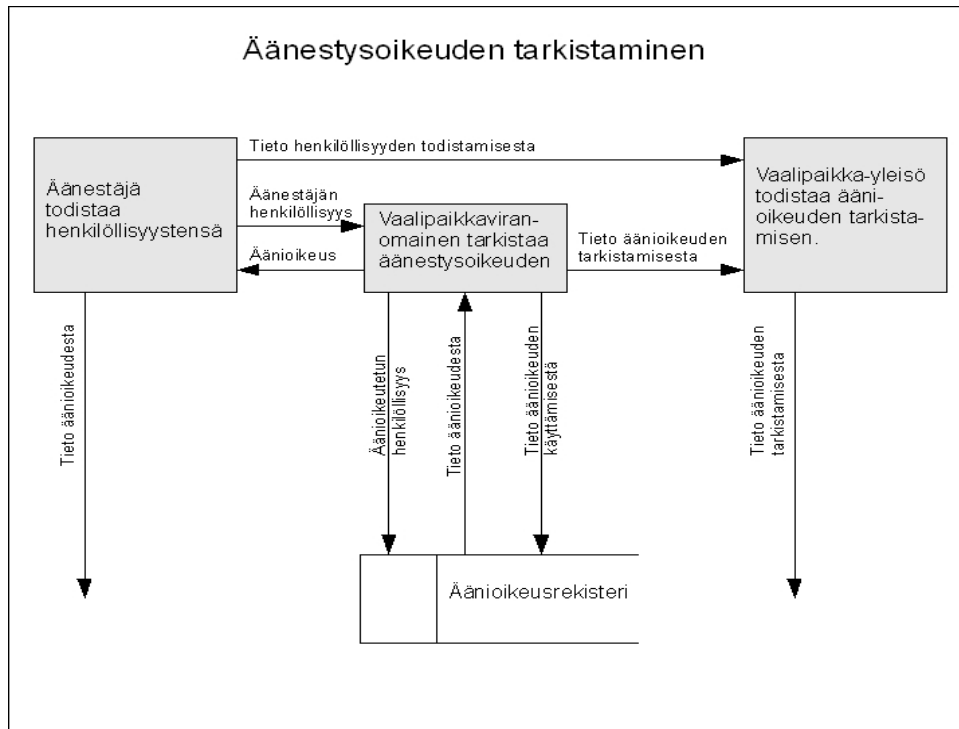
Tietovirrat ovat olennaisia, kun analysoidaan tietojen totuusarvoa ja perusteltavuutta. Jos tiedon perusteltavuus perustuu sen julkisuuteen ja tämä tieto on tai käy julkisuudelta tai äänestäjältä piilossa, on argumentoitavissa että tiedon perusteltavuus katoaa tai on vähintään kriittisesti tarkasteltava ennen kuin se voidaan hyväksyä tiedoksi.

4. Äänioikeuden tarkastaminen ja äänioikeus

Äänioikeuden tarkastamisen osalta äänestysprosessi ei poikkea sähköisen ja manuaalisen äänestysjärjestelmän välillä lainkaan. Kummassakin tapauksessa äänestäjä saapuu paikalle, jakaa henkilöllisyytensä vaalivirkailijalle, joka tarkistaa äänestäjän äänioikeuden. Tämän perusteella vaalivirkailija päättää äänioikeuden toteuttamisesta äänestäjälle (kuva 1).

Poikkeus tässä vaiheessa on, että manuaalisessa äänestyksessä voidaan todeta äänestäjän saavan välineet äänestämiseen eli äänestyslipun, kun sähköisessä äänestämässä äänestäjä saa äänestysavaimen. Tämä avain mahdollistaa äänestämisen kopissa. Jos oletamme olevan yleistä tietoa, että avain tosiasiallisesti mahdollistaa äänestämisen, eivät vaiheet eroa äänioikeuden toteutumisen

näkökulmasta toisistaan. Kummallakin tavalla äänestäjistä voimme todeta, että äänestäjä on oikeutettu pitämään totena tietoa äänioikeudestaan, vaaliviranomainen samoin, ja vaalipaikan yleisö todistaa sekä äänioikeuden tarkastamista että lupaa varsinaiseen äänestämiseen.



Kuva 1: Tietovirtakaavio äänioikeuden tarkastamisesta

Kriittisesti ajatellen sähköisen äänestyksen kortti on palauttavissa äänestäjään ja näin ollen luo sillan äänioikeuden ja äänen välille, mikä rikkoo periaatetta äänen ja äänestäjän eriyttämisestä. Äänestäjällä ei voi olla tietoa käytetäänkö kortin äänestäjää tunnistavia tietoja, kun kortti syötetään äänestyskopissa.

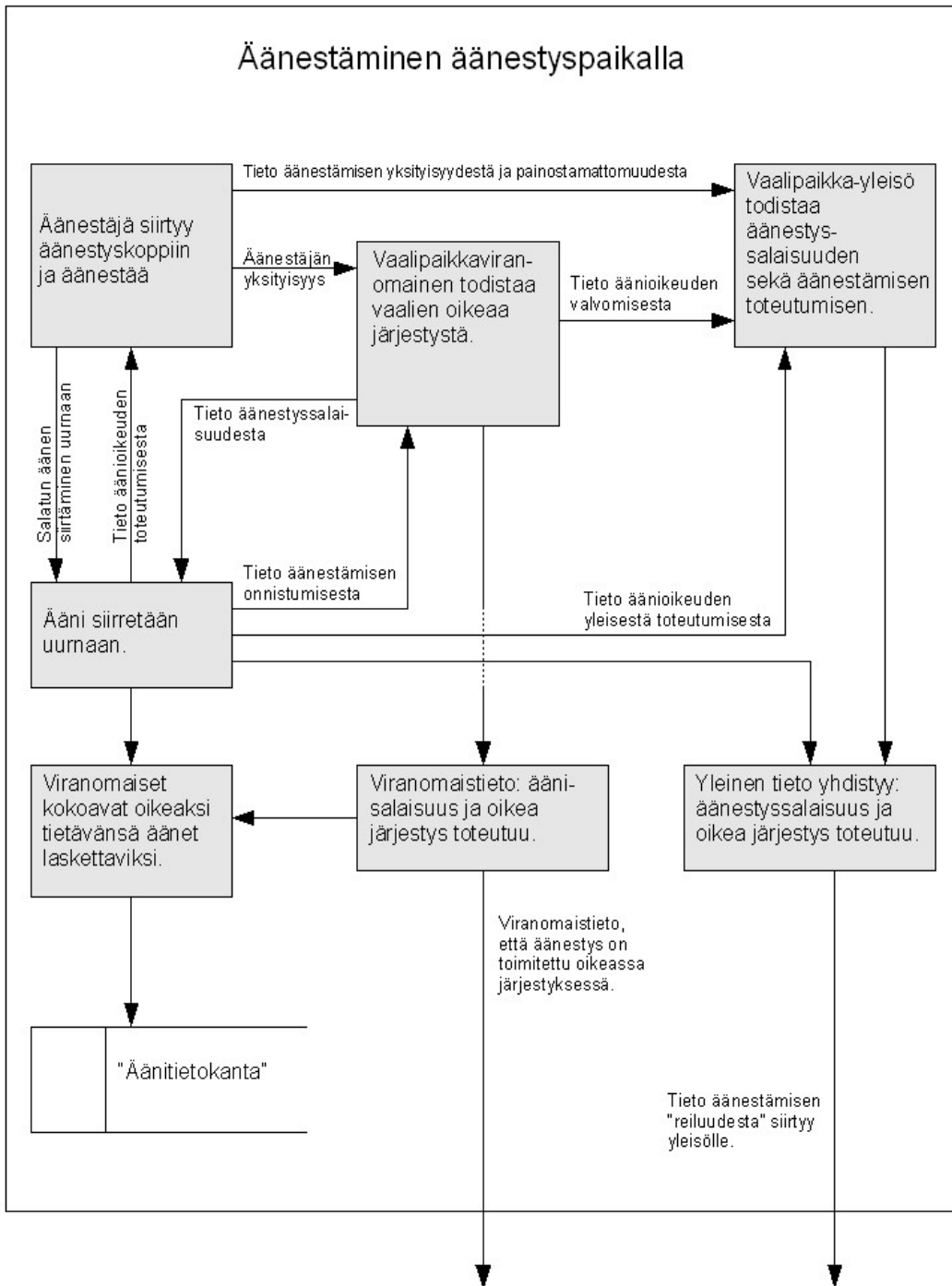
5. Äänestäminen ja äänen tallentaminen

Äänestämisen ja äänen tallentamisen osalta manuaalinen ja sähköinen äänestäminen poikkeavat siinä mittakaavassa, että niitä on syytä tarkastella erikseen.

5.1. Äänestäminen ja äänen tallentaminen manuaalisessa järjestelmässä

Äänestäjä siirtyy äänestyskoppiin ja salaa näin toimintansa eli äänen antamisen yksityiskohdat. Äänestyskopissa äänestäjä antaa äänensä kirjoittamalla ehdokkaan numeron äänestyslipukkeelle. Äänestäjä voi antaa myös tyhjän tai muutoin kelvottoman äänen halutessaan. Äänestäjä sulkee lipukkeen ja siirtyy ulos äänestyskopista. Äänestäjä laittaa äänensä uurnaun viranomaisen leimalla va-

rustettuna, näin siirtäen äänensä todennettavasti äänien muodostamaan data-rykelmään, joista äänntenlaskennassa muodostetaan äänestäjiä erittelemätön tietokanta (kuva 2).



Kuva 2: Manuaalisen äänestämisen tietovirrat

Kaikki manuaalisen äänestämisen vaiheet ovat tarkasteltavissa ja todennettavissa sekä äänestäjän, viranomaisten että yleisön toimesta. Vain itse äänestyskoppikäyttäjän suhteen on olemassa rajaus: tätä toimenpidettä ei voi suoranaisesti tarkastella kuin toimiessaan itse äänestäjänä, sitä ei voida todentaa

jokaisella suorituskerralla. Äänestyskopin ulkopuolisella tiedolla yhdessä henkilökohtaisella tiedolla äänestyskopin sisältä tapahtuvista asioista voidaan kuitenkin perustella hyvin kattavasti tarvittava tieto äänestysjärjestelmän toiminnasta.

Äänestäjän ja äänen välinen yhteys katkeaa kaksivaiheisesti. Annettua ääntä ei voi yhdistää kehenkään äänestäjään missään vaiheessa, sillä ääni on kapseloitu perinteisen äänestyslipukkeen kautta anonyymiksi. Äänestyslipun kuulamista henkilölle ei voida osoittaa muutoin kuin siinä vaiheessa kun ääntä siirryttään antamaan tai ääni siirretään urnaan. Ääntä ja äänestyslippua ei voida yhdistää äänestyksen aikana kuin äänestyskopissa ja vain äänestäjä itse. Kun äänestyslipuke lopulta avataan ääntenlaskun yhteydessä, ei ole jäljellä mitään kohtuullista keinoa äänen ja äänestäjän yhdistämiseen, kun yhteys äänestäjän ja äänestyslipukkeen välillä on kadotettu. Jos äänestysalueella katsotaan olevan liian vähäinen määrä äänestäjiä nimettömyyden saavuttamiseksi, yhdistetään äänet toisen äänestysalueen äänien kanssa. Tällä taataan tosiasiallinen anonymitteetti.

Osallistuminen ja pakottamattomuus todentautuvat sekä henkilökohtaisen että julkisen yleisön tarkkailun kautta. Jokainen äänioikeutettu voi todeta äänestäneensä rauhassa äänestyskopissa ja ääni siirretään urnaan julkisesti, minkä perusteella voidaan perustellusti sanoa äänestämisen tapahtuneen ja olleen pakottamatonta. Sama pätee myös äänestysoikeuden käyttämiseen kerran ja vain kerran.

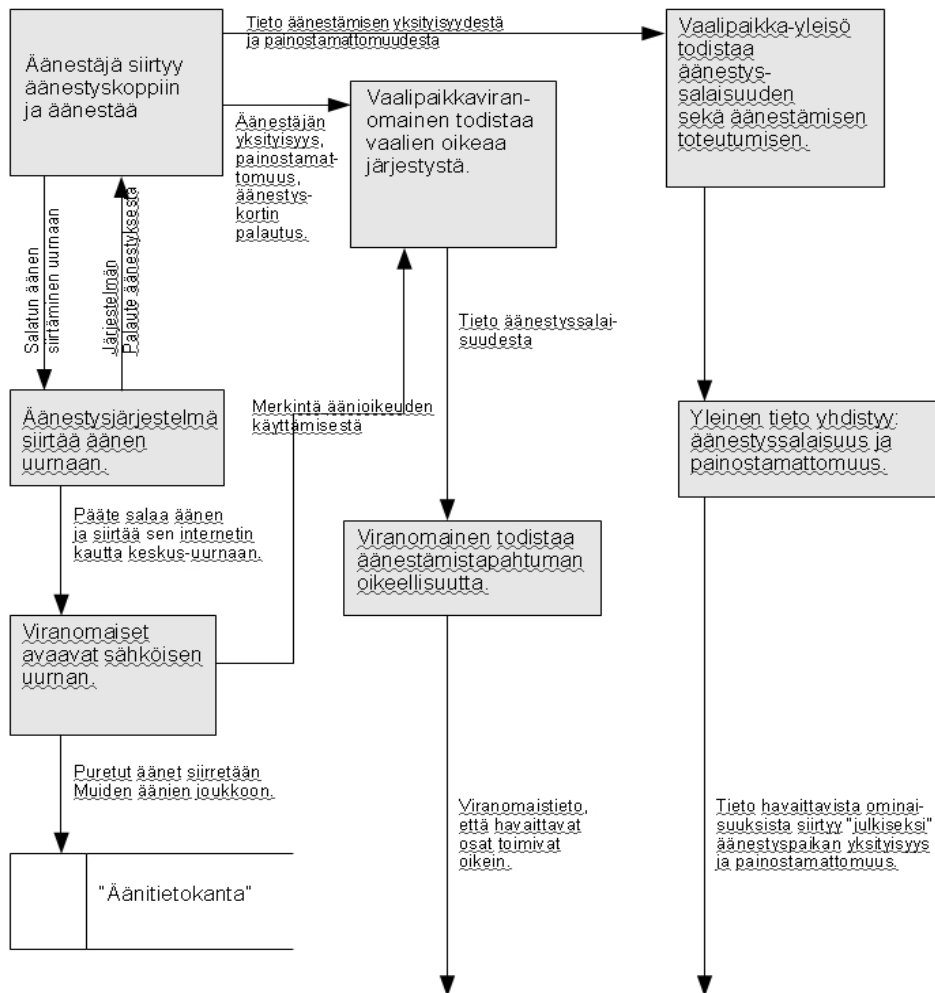
Koska äänestysoikeus toteutuu vain kerran ja ääni ei ole jäljitettävissä, voidaan muuttamattomuus johtaa näiden kautta. Ei ole mahdollista vaihtaa ääntänsä, jos ei ole mahdollisuutta äänestää kuin kerran ja kerran annettua ääntä ei voi antamisen jälkeen jäljittää edes sen antaja.

Vakuuttuminen äänen tulemisesta lopputulokseen tapahtuu johtamalla eri tietoja. Koska voidaan vakuuttua vaalipaikalla tapahtuneen äänestyksen oikeellisuudesta, voidaan perustellusti todeta äänien olevan urnassa. Äänet siirretään ääntenlaskupaikalle luottamusmiesten valvonnassa. Äänet lasketaan luottamusmiestoimintana ja tulos on tarkastettavissa. Jos tässä toiminnassa olisi jotakin, joka antaisi aiheen epäillä lopputuloksen oikeellisuutta, voidaan kohtuudella odottaa, että se tulisi esille, koska kaikki vaiheet ovat nähtävissä.

Perinteisen lippuäänestyksen kohdalla takaisinkytkennöistä suurta osaa näyttelee asioiden havaitsettavuus. Tietoa on mahdollista olla, koska tietojenkäsittely toimii sellaisella tasolla, josta kaikilla äänioikeutetuilla on mahdollisuus saada tietoa. Tämä puolestaan antaa oikeutuksen tiedolle, koska se on perusteltavissa olevaa, empiirisesti havaittavaa.

5.2. Äänestämisen ja äänen tallentaminen sähköisessä järjestelmässä

Sähköisessä äänestyksessä äänestäjä siirtyy äänestyskoppiin ja syöttää sähköisen äänestyskortin äänestyspääteeseen. Graafisen, kosketusohjattavan käyttöliittymän avulla äänestäjä valitsee kenelle hän antaa äänensä, vai äänestääkö tyhjää. Hän vahvistaa valintansa ja poistuu äänestyskopista kortin kanssa. Vahvistettu ääni siirtyy sähköiseen urnaan järjestelmän toimittajan tiloissa ja äänen siirtymisestä jää merkintä. Äänestyskortti palautetaan äänestysviranomaisille. Äänestäjä voi halutessaan tarkistaa äänestämisen onnistumisen viranomaiselta myös jälkikäteen, niin että viranomainen voi tarkistaa, onko järjestelmä tallentanut kyseisen äänestäjän äänen siirtymisen sähköiseen urnaan (kuva 3).



Kuva 3: Sähköisen äänestämisen tietovirtoja

Sähköisen äänestämisen ulkoiset puitteet ovat samankaltaiset kuin perinteisessä äänestystavassa äänestyskopin osalta. Äänestäjä on yhtälailla ulkoisesti suojattuna painostamiselta ja yksityisyys on turvattu äänestyskopin luoman

esteen avulla. Ero syntyy pääosin äänestyskopin ulkopuolisista ja sähköisistä järjestelmän osista.

Sähköisessä äänestystavassa ääntä ja äänestäjää ei eroteta samoin kuin perinteisessä. Sähköisessä järjestelmässä äänestäjän ääni tai sen viitetiedot, on yhdistettävissä äänestäjään myös jälkikäteen [Karhumäki ja Meskanen, 2008]. Koska laitteisto ja lähdekoodi, joiden perusteella voitaisiin päätellä, missä vaiheessa äänen ja äänestämisen yhdistäminen häviää, ovat liikesalaisuuksia, ei äänestäjällä voi olla varsinaista tietoa näiden tietojen erottamisesta. Viranomaislausunnossa [Jääskeläinen, 2008] todetaan, että "Äänestäjän on mahdollista tarkistuttaa vaalivirkailijalta, onko hänen antamansa ääni sähköisessä vaaliuurnassa." Tämä viittaisi siihen, että ääntä ja äänestäjää ei ole eriytetty, vaikka ääni olisikin salattu.

Äänestäjä ei voi todentaa äänensä siirtymistä urnaan, vaan äänestäjän tulee luottaa järjestelmän antamaan palautteeseen. Tiedon perusteltavuus on täysin järjestelmän varassa, eikä ole olemassa keinoja tuon järjestelmän oikeellisuuden tarkasteluun. Tiedämme kriittistenkin sähköisten järjestelmien tekevän virheitä [McGaley and McCarthy, 2004], kuten äänestyskokeilussa kävi ilmi. On siis olemassa kohtuullinen oletus, että sähköinen vaalijärjestelmä on virheillemme altis.

Pilotoitu äänestysjärjestelmä ei toteuttanut tiedollisuutta yhden ja vain yhden kerran äänestämistä. Järjestelmä antoi äänestää toistamiseen, mikäli äänioikeus ei kirjautunut käytetyksi eli viitetietoa äänen siirtymisestä urnaan ei löytynyt. Näin palaamme samaan ongelmaan kuin aiemmin: emme voi tietää, onko ääni urnassa vai oliko virhe mahdollisesti viitetiedoissa.

Äänten tosiasiallista uudelleenlaskentaa ei ole mahdollista tehdä sähköisessä järjestelmässä. Ei ole työkalua sen tarkistamiseen, onko kone tulkinut oikein tai tallentanut tiedon oikein, vaan "tarkastuslaskenta" tuottaa saman tiedon yhä uudestaan. Oikeusministeriö on esittänyt, että äänten oikeellisuus voitaisiin varmistaa vertaamalla niitä muun muassa äänioikeutettuihin. Tämänkaltaiset keinot eivät kuitenkaan kerro mitään siitä, onko yksittäisen äänestäjän ääni mennyt oikein tai onko esimerkiksi data muuttunut niin että ääni menee väärälle ehdokkaalle. Vertailu ei myöskään millään tavalla mahdollista virheenkorjausta eli uudelleenlaskentaa, vaan korkeintaan kertoo että jossakin on sattunut virhe.

6. Manuaalisen ja sähköisen äänestysjärjestelmän vertailu ja analyysi käyttäjävaatimusten toteutumisesta

Kuten aiemmin on todettu, tulee sähköisen järjestelmän toteuttaa samat vaatimukset kuin manuaalisen äänestyksen, joten sähköisen äänestyspilotin toimin-

taa voidaan arvioida vertailemalla sitä manuaaliseen vaalijärjestelmään. En lähde kyseenalaistamaan perinteisen vaalijärjestelmän lähtökohtia, vaan vertaan näitä kahta sillä olettamalla, että käytössä olevan manuaalisen vaalijärjestelmän vaatimukset ovat kohdallaan ja näille vaatimuksille on olemassaolon oikeutus.

6.1. Viranomaistiedon ja liikesalaisuuksien probleema

Ongelmalliseksi jäävä yksittäinen probleema on sähköisen järjestelmän luotettavuus. Kuinka paljon voimme luottaa järjestelmään, jonka toimintaa emme voi tarkastella siinä määrin kuin toisenlaisen, omin silmin todistettavan järjestelmän toimintaa? Voimmeko pitää perusteltuna tietoa, joka syntyy toisen käden tietona, viranomaisten ja yritysten vakuuttelun tuloksena?

Koska viranomaiset ovat tiedollisesti pitkälti järjestelmätoimittajan varassa, on ensisijainen informaation lähde järjestelmätoimittaja. Toimittajan intressit eivät kuitenkaan ole välttämättä samansuuntaiset kuin demokratian tarpeet, vaan voivat olla voimakkaassakin ristiriidassa näiden kanssa [McGaley and McCarthy, 2004]. Koska toisen käden tiedon perustelut syntyvät pikemminkin rationaalisesti kuin empiirisesti, voidaan perustelut kyseenalaistaa ristiriitaisilla motiiveilla.

Arvioidessa viranomaistietoa tai informaatiota järjestelmän toimittajalta voidaan kiinnittää kriittistä huomiota annettuihin lausuntoihin. Oikeusministeriö väitti lausunnossaan perustuslakivaliokunnalle, että "Käytettävien ohjelmistojen virheetön toiminta on varmistettu." [Jääskeläinen, 2008], mitä voidaan pitää sinällään kyseenalaisena väitteenä. Teoria virheettömästi toimivista ohjelmistoista on tietojenkäsittelytieteen piirissä vasta idea, jonka toteutumiseen ei ole mitään tieteellistä todistusaineistoa.

Samassa lausunnossa todetaan, että "järjestelmään ei perustu avoimeen lähdekoodiin ja tällöin yleinen käytäntö, että järjestelmän koodin tarkastaa – äänestäjien puolesta – riippumaton auditoija, jonka raportti on julkinen". Sisäisessä selvityksessä Oikeusministeriö[2008] toteaa, että "Ohjelmiston ja laitteiston testaus oli selkeästi jätetty auditoinnin ulkopuolelle." Tämän perusteella voidaan kohtuullisesti väittää, että ohjelmiston varsinaisesta toiminnasta tai toimimattomuudesta ei ole sen kaltaista viranomaistietoa, jota voisi käyttää perusteluna tiedolle järjestelmän toimivuudesta.

6.2. Manuaalisen ja sähköisen järjestelmien vaatimusten toteutuminen ja vertailu demokraattisten vaalien periaatteisiin

Vaalit eivät ole vain demokratiassa viitekehyksessä toimiva järjestelmä, vaan vaaleja käytetään sekä edustuksellisen demokratiaa muistuttavissa järjestelmissä että täysin demokratiaa tunnustamattomissa järjestelmissä. Vaaleja voidaan

käydä muun muassa osakeyhtiön hallitusta valittaessa, jolloin ei ole tarkoituksenmukaista tai edes viitekehykseen sopivaa toteuttaa esimerkiksi yksi ääni per äänestäjä -periaatetta. Tässä tutkimuksessa käsitellään kuitenkin edustuksellisen demokratian vaaleja, jolloin vaalien demokraattisten periaatteiden toteuttaminen on olennaista.

Kun tietoa oletetaan olevan vain jokaisen äänestäjän henkilökohtaisen kriittisen tarkastelun alainen tieto, jonka käyttäjän on mahdollista henkilökohtaisesti varmistaa, on sähköisen ja manuaalisen vaalijärjestelmän ero valtaisa. Kokeilu ei toteuttanut juuri ollenkaan demokraattiselta vaalijärjestelmältä vaadittavaa luottamus-periaatetta.

6.2.1 Yleisyyden periaate

Yleisyyden periaatteen alle kuuluu vain ensimmäinen käyttäjävaatimus osallistumisen todennettavuudesta. Sähköisessä äänestyskokeilussa se jäi toteutumatta. Osallistumisen todennettavuutta ei voida perustella, koska käyttäjällä eli äänestäjällä ei ole tietoa siitä, tuliko ääni tallennetuksi. Sen sijaan 2,3 % epäonnistuminen viittaa siihen suuntaan, että osallistuminen ylipäättään on jäänyt toteutumatta, eikä pelkästään sen todentuminen.

6.2.2 Vapauden periaate

Yleisyyden vaatimukset näyttävät toteutuneen pilotoinnissa. Pakottamattomuus on yhtä ilmeistä kuin manuaalisessa äänestämisessä ja saatavilla olevan perustellun tiedon mukaan järjestelmä on antanut tosiasiallisen mahdollisuuden antaa tarkoituksenmukainen kelvoton ääni. Kriittisesti lähestyen voitaisiin todeta, että kelvottoman äänen mahdollisuus jää toteutumatta, koska äänten tallentumisesta ja laskemisesta ei itsessään voida olla varmoja, mutta se ei sinällään kuulu yleisyyden vaan muiden periaatteiden piiriin. Näin ollen yleisyyttä voidaan pitää toteutuneena.

6.2.3 Yhtäläisyyden periaate

Yhtäläisyyden vaatimus sisältää melkein puolet käyttäjävaatimuksista, kysymykset neljästä kahdeksaan. Yksikään kysymys ei saa vastaukseen totta, joten vaatimuksen toteutumisen voidaan katsoa epäonnistuneen täysin.

Sähköisestä järjestelmästä ei ole saatavissa tietoa, saivatko kaikki äänioikeutetut tosiasiallisesti äänestää. Yhden ja vain yhden kerran äänestäminen on myöskin epäselvä. Koska järjestelmä on antanut äänestää uudestaan, mikäli äänioikeutta ei ole merkitty käytetyksi, on osa äänestäjistä tosiasiallisestikin äänestänyt useamman kuin yhden kerran. Äänioikeusmerkinnän oikeellisuudesta ei ole tietoa, joten mikä tahansa vaihtoehto ei yhdestäkään äänestä useampaan ääneen voi pitää paikkansa.

Samankaltainen ongelma vaivaa kaikkia tämän periaatteen kysymyksiä: toteutumisesta ei voida sanoa mitään, koska se ei ole todennettavissa. Yksittäisten ongelmien, kuten äänestyskertojen tai äänten katoamisen, lisäksi on siis olemassa yksi yleisempi syy, joka estää periaatteen toteutumista, vaikka muut ongelmat korjattaisiin.

6.2.4 Vaalisalaisuuden periaate

Vaalisalaisuuden periaate toteutuu niiltä osin, miltä pilotin järjestelmä noudattaa perinteisen äänestysjärjestelmän toimintaa, eli äänen antamista. Voidaan pitää totena, että ääni on annettu julkisuudelta piilotettuna äänestyskopin ansiosta.

Vaalisalaisuuden piiriin kuuluu kuitenkin myös vaatimukset äänten ja äänestäjien erottelemisesta ja äänestys- ja äänioikeus-järjestelmien erottelemisesta. Näiltä osin pilotti ei toteuttanut vaalisalaisuutta, vaan äänestäjien ja äänen erottelu jää lopulta kokonaan arvoitukseksi ja äänioikeusjärjestelmä linkittyy sähköisen äänestyskortin kautta äänestysjärjestelmään ja äänien identiteettiin.

6.2.5 Välittömyyden periaate

Välikäsien käyttö näyttää kohtuullisesti estetyltä pilottihankkeessa, joten välittömyys toteutuu tältä osin. Välittömyyden periaate sisältää myös vaatimuksen äänten oikeellisesta tallentamisesta ja laskemisesta, mikä ei tule todistetuksi. Näin tämä periaate toteutuu osittain.

6.2.6 Luottamuksen periaate

Luottamuksen periaate kerää alleen kaikkien periaatteiden toteutumisen, joita toteuttamalla äänestäjällä olisi oikeus ja velvollisuus luottaa järjestelmän toimintaan. Vaalijärjestelmän viidestä periaatteesta toteutuu pilotoinnissa kokonaan yksi ja toinen osittain. Luottamusperiaate vaatisi [Pieters, 2006], muiden vaatimusten ohella, kaikkien periaatteiden toteutumista, joten sen voidaan sanoa jääneen todistetusti toteutumatta.

6.2.7 Käyttjävaatimusten yhteenveto

Taulukko 1 selventää järjestelmien eroa kokoamalla toteutuneet ja toteutumattomat vaatimukset.

Käyttjävaatimusten toteutuminen		
Vaatus	Manuaali järjestelmä	Sähköinen järjestelmä
Osallistumisen äänestämiseen tulee olla todennettavissa	kyllä	ei
Pakottamattomuus pitää olla taattua	kyllä	kyllä
Mahdollisuus antaa kelvoton ääni tarkoituksella pitää olla mahdollista	kyllä	kyllä
Vain äänioikeutettujen pitää pystyä äänestämään ja vain kerran	kyllä	ei
Annettua ääntä ei saa pystyä muuttamaan tai toistamaan	kyllä	ei
Äänestäjän pitää kyeta todentamaan, että annettu ääni on laskettu lopputulokseen	kyllä	ei
Äänestäjillä tulee olla syrjimätön mahdollisuus käyttää äänestysjärjestelmää	kyllä	ei
Äänioikeus- ja äänestysjärjestelmien tulee olla todistettavasti eriytettyjä	kyllä	ei
Äänet tulee varmentaa erillään äänestäjien varmentamisesta	kyllä	ei
Äänestyksessä ei saa käyttää välikäsiä, ts. Toinen ihminen ei voi päättää toisen ihmisen äänestä (ei valtakirjoja)	kyllä	kyllä
Jokainen äänestyslipuke tulee tallentaa ja laskea oikein	kyllä	ei

Taulukko 1. Käyttjävaatimusten toteutuminen manuaalisessa ja sähköisessä äänestysjärjestelmässä.

Kuten taulukosta voidaan havaita, ei sähköinen äänestyspilotti toteuta kuin kolme vaatimusta demokraattisista vaaleista. Toteutuneet ominaisuudet palautuvat vaalien vapauteen, jonka kummatkin alakohdat tulevat toteutuneeksi, sekä välittömyyteen, josta täyttyy toinen kahdesta alakohdasta. Näin pilotin toteutumisen ulkopuolelle jäivät vaatimuksista yleisyys, yhtäläisyys, vaalisalaisuus sekä osittain välittömyys verrattuna manuaaliseen äänestykseen.

6.3. Viranomaistiedon ja liikesalaisuuksien oletaminen perusteluiksi

Vaikka olettaisimme Tietoenatorin ja viranomaisten antamien tietojen olevan tosia, ei pilotoinnin äänien katoamista koskevaa virhettä ole kohtuullista laittaa perustelemattomasti käyttäjien vastuulle. Virheellisestä toiminnasta ei ole olemassa mitään julkista tutkimusta. Näin ollen osallistumisen todennettavuus, äänioikeutettujen äänestämisen vain kerran, annetun äänen muuttaminen tai toistaminen, äänen todentaminen lopulliseen tulokseen, syrjimättömyys, äänioikeus- ja äänestysjärjestelmien eriyttäminen, äänen varmentaminen erillään ja jokaisen äänestyslipukkeen laskeminen eivät edelleenkään toteudu.

Tästä päätellen viranomaistiedon tai liikesalaisuuksien oletaminen tiedon perusteluiksi muuttaisi tämän tutkimuksen tulosta lähinnä läpinäkyvyyden osalta. Se ei kuitenkaan poistaisi lukuisia ongelmia sekä väärässä informaatio-

sa, jota viranomaistahoilla on järjestelmästä tai kyseenalaista järjestelmänmäärittelyä. Se ei myöskään poistaisi tapahtuneita virheitä.

7. Yhteenveto

Tutkimuksessa tarkasteltiin Suomen sähköistä äänestyskokeilua rajatusta näkökulmasta - viitekehyksenä toimii järjestelmän käyttäjälle eli äänestäjälle annettu tieto sen toiminnasta. Tulokset ovat suhteellisen yksitulkintaisia, Suomen sähköisen äänestyksen pilotointi vuoden 2008 kunnallisvaaleissa oli tämän tutkimuksen viitekehyksessä kategorinen epäonnistuminen. Osan syystä kantaa varmasti se, ettei aidosti demokraattisia vaatimuksia järjestelmän toiminnasta esitetty missään vaiheessa suunnittelua. Samankaltaisia ongelmia sähköisten äänestysjärjestelmien kanssa on ollut muuallakin, esimerkiksi Yhdysvalloissa [Dill et al., 2003].

Eri periaatteiden eritasoinen toteutuminen kertoo ongelmista, joihin tulisi kiinnittää huomiota. Vapauden periaate toteutui moitteettomasti, mitä pidetään näkyvimpänä osana vaalijärjestelmää. Vapaudelle läheinen välittömyyden periaate toimi myös osittain, mutta loput ominaisuudet jäivät toteutumatta. Näyttää siltä, että ohjelmistokehitysprosessi on itsessään ollut virheellinen, eikä välttämättä ole ymmärretty, että monimutkaisen tietojärjestelmän suunnittelu ei ole sama kuin yksittäisen laitteen ohjelmiston suunnittelu.

Jos äänestysjärjestelmä olisi toiminut avoimen lähdekoodin pohjalta suljetun sijaan ja antanut käyttäjälle ja järjestelmälle varmuuden äänestyksestä esimerkiksi tallennettavan paperitulosten muodossa, olisi iso osa ongelmista vältetty. Kaikkia ongelmia sekään ei olisi ratkaissut. Esimerkiksi avoimen lähdekoodin käyttö ei todista sen aktuaalista käyttöä äänestystilanteessa käytössä olevassa äänestystilanteessa. Kysymykseksi jää, voiko sähköisestä äänestysjärjestelmästä ylipäättään saada niin varmaa palautetta toiminnasta, että se voitaisiin keskeisessä demokraattisessa vaalissa hyväksyä käytettäväksi.

Viiteluettelo

[Anttiroiko, 2003] Ari-Veikko Anttiroiko, Building Strong E-Democracy – The Role of Technology in Developing Democracy for the Information Age, *Communications of the ACM* **46**, 9 (Sep, 2003), 121-128.

[Dill et al., 2003] David L. Dill, Bruce Schneier and Barbara Simons, Voting and Technology: Who Gets to Count your Vote?, *Communications of the ACM*, **46**, 8 (Aug 2003), 29-31.

- [Euroopan neuvosto, 2004] Euroopan neuvoston suositus koskien sähköisen äänestyksen laillisia, toiminnallisia ja teknisiä normeja, Euroopan neuvosto, epävirallinen käännös, 2004, Saatavilla <http://www.vaalit.fi/uploads/sv74q1ff29czyb.doc>, 10.9.2009.
- [Gritzalis, 2002] Dimitris A. Gritzalis, Principles and requirements for a secure e-voting system, *Computers & Security*, **21**, 6/2002, 539-556.
- [HE 14/2006] Hallituksen esitys Eduskunnalle laiksi vaalilain muuttamisesta, Saatavilla <http://www.finlex.fi/fi/esitykset/he/2006/20060014>, 17.9.2009.
- [HHO, 2009] Helsingin hallinto-oikeus, Päätös sähköisen äänestyksen uusimisesta jätetystä valituksesta, Diaarinumero 06788/08/2201, 2009. Saatavilla <http://www.effi.org/blog/hhao-2009-01-29.html>, 10.9.2009.
- [Huitti et al., 2003] Ari Huitti, Joe Niva, Eino Kainulainen ja Samuli Haikonen, Vaalitietojärjestelmän esimäärittely, Loppuraportti (versio 1.31), TietoEnator, 2003. Saatavilla <http://www.effi.org/yksityisyys/vaalitietojarjestelma/vaalitietojarjestelma-esimaarittely.pdf>, 10.9.2009.
- [Jääskeläinen, 2002] Arto Jääskeläinen, Oikeusministeriön julkaisu: Suomen vaalijärjestelmä, Muistio, Oikeusministeriö, 2002.
- [Jääskeläinen, 2008] Arto Jääskeläinen, Selvitys sähköisen äänestysmenettelyn luotettavuudesta, Selvitys, Oikeusministeriö, 2008.
- [Karhumäki ja Meskanen, 2008] Juhani Karhumäki ja Tommi Meskanen, Auditointiraportti kunnallisvaalien sähköisen äänestyksen pilotista, Turun yliopisto, 2008, Saatavilla <http://www.vaalit.fi/uploads/6d8qgeom5g.pdf>, 17.9.2009.
- [KHO, 2009] Korkein hallinto-oikeus, Päätös sähköisen äänestyksen uusimisesta jätetystä valituksesta, Diaarinumero 687/1/09, 2009. Saatavilla <http://www.kho.fi/paatokset/46372.htm>, 10.9.2009.
- [L714, 1998] Vaalilaki, 1998, Saatavilla <http://www.finlex.fi/fi/laki/ajantasa/1998/19980714>, 17.9.2009.
- [L731, 1999] Suomen Perustuslaki, 1999, Saatavilla <http://www.finlex.fi/fi/laki/ajantasa/1999/19990731>, 17.9.2009.
- [McGaley, 2008] Margaret McGaley, E-voting: an Immature Technology in a Critical Context, *PhD Dissertation, Dept. of Computer Science, NUI Maynooth*.
- [McGaley and McCarthy, 2004] Margaret McGaley and Joe McCarthy, Transparency and e-Voting: Democratic vs. Commercial Interests, *Electronic Voting in Europe - Technology, Law, Politics and Society*, European Science Foundation, 153-163.

- [Muhlberger, 1998] Steve Muhlberger, Democracy in Ancient India, 1998, Saatavilla
<http://www.nipissingu.ca/department/history/muhlberger/histdem/in-diadem.htm>, 24.12.2009.]
- [Oikeusministeriö, 2008] Oikeusministeriön sisäisen tarkastuksen yksikkö, Selvitys sähköisen äänestyksen kokeilu, Selvitys, Diaarinumero OM 11/0152/2008, 2008.
- [Pieters, 2006] Wolter Pieters, Acceptance of Voting Technology: Between Confidence and Trust, In: *4th International Conference, iTrust 2006, Lecture Notes in Computer Science* **3986**, Springer, 283-297.
- [Tietoenator, 2008] Sähköisen Äänestyksen Pilotti 2008, Tekninen Toteutus ja Tietoturvaratkaisut, Tietoenator, 2008, Saatavilla
<http://www.vaalit.fi/uploads/7aanqsm6czk.pdf>, 17.9.2009.

Virtuaalinen tuotekokemus

Taina Lempiäinen

Tiivistelmä.

Internetin alati kasvavien kävijämäärien myötä yhä useampi yritys markkinoi tuotteitaan myös sähköisesti verkossa. Kehittyvät teknologiat mahdollistavat tuotteen esittelyn lukuisilla eri keinoilla. Tässä tutkielmassa käsittelen virtuaalista tuotekokemusta ja sitä, miten erilaiset tuote-esitykset vaikuttavat siihen, kuinka hyvin asiakas ymmärtää tuotteen ominaisuudet ja toiminnot. Kuvat, videot, 3D-esitykset ja erilaiset interaktiiviset ominaisuudet ovat esimerkkejä tavoista, joiden kautta yritykset voivat antaa asiakkaalle tietoja tuotteista. Käytän esimerkkeinä kolmea erilaista tuote-esittelyä kolmesta eri verkkokaupasta. Esittelen aikaisempia tutkimuksia aiheesta sekä vertaan niitä omiin esimerkkeihini. Teksti ja kuvat osoittautuivat parhaimmiksi kertomaan asiakkaalle yksityiskohtaisia tietoja, kun taas videot ja interaktiiviset teknologiat olivat tehokkaimpia välittämään itse koettavia ominaisuuksia, kuten esimerkiksi ulkonäköä.

Avainsanat ja -sanonnat: virtuaalinen tuotekokemus, sähköinen kauppa, verkkokauppa, tuote-esittely

CR-luokat: H.4, H.5

1. Johdanto

Internetin käyttäjämäärät ja verkkoteknologiat kasvavat jatkuvasti ja internetissä tapahtuva elektroninen kaupankäynti (e-kaupankäynti) mielletään yhdeksi tietotekniikan ja viestintäteknologioiden tärkeimmistä sovelluksista [Manvi and Venkataram, 2005]. Internetin yleistymisen vuoksi tuotteiden myyntiä ja markkinointia hoidetaan yhä enemmän sähköisesti verkon kautta. Verkkokaupankäynnin tarkoitus on auttaa ostajia tutkimaan tuotetta, josta on kiinnostunut, ja päättämään hinnan, maineen, tuotteen saatavuuden ja palvelun perusteella kieneltä jälleenmyyjältä sen haluaa ostaa [Manvi and Venkataram, 2005]. Asiakkaat saavat kaiken tiedon kiinnostavasta tuotteesta verkkosivuilla olevan tuote-esittelyn perusteella. Ne suunnitellaan esittelemään tuote asiakkaille, auttaamaan heitä ymmärtämään tuotteen toimintaa ja vakuuttamaan asiakkaat siitä, että tuote on muita toimivampi ja houkuttelevampi [Hoch and Deighton, 1989].

Tässä tutkielmassa keskityn tutkimaan erilaisten tuotekuvien vaikutusta siihen, kuinka hyvin tuotteen ominaisuudet ja toiminnot välittyvät asiakkaalle. Käytän esimerkkeinä kolmea erilaista tuotekuvaa eri verkkokaupoista. Yksi

tuotekuva koostuu pelkästään kuvasta ja tekstistä, yhdessä on mukana video ja kolmannessa on 3D-tekniikalla toteutettu kuva, jota pystyy pyörittelemään ja klikkaamaan hiirellä valitakseen osat, joita haluaa tutkia tarkemmin. Tuotteet eivät kuulu samaan tuoteryhmään, mutta pyrin valitsemaan monimutkaisuuskiltaan ja ominaisuuksien määriltään samantyyppiset tuotteet. Jiang ja Benbasat [2007] huomasivat tutkiessaan kahden monimutkaisuudeltaan erilaisen tuotteen erilaisia tuotekuvia, että tulokset vaihtelivat eri määrän ominaisuuksia sisältävien tuotteiden välillä. Siksi pyrin valitsemaan tässä suhteessa samankaltaiset tuotteet.

Tutkielman eteneminen on seuraavanlainen. Luvussa kaksi käsitellään aiempia tutkimuksia ensin yleisemmin ja sen jälkeen tuotekuvan ominaisuuksien näkökulmasta. Kolmosluvussa esitellään ensin tämän tutkielman esimerkkeinä käytetyt tuotteet ja sen jälkeen kerrotaan erilaisista tuote-esittelyistä. Luvussa neljä verrataan esimerkkien arviointeja aikaisempiin tuloksiin. Viidennessä luvussa on lyhyt yhteenveto tutkielmasta sekä pohdintaa tutkielman rajoituksista sekä siitä, miten tutkimusta voisi jatkaa tulevaisuudessa esimerkiksi pro gradu -tutkielmaksi.

2. Teoreettinen tausta

2.1. Taustaa

Fyysisten tuotteiden ostaminen internetistä on haastavaa, koska tuote ei ole asiakkaan nähtävillä ennen ostopäätöstä [Lightner and Eastman, 2002]. Asiakkaat eivät voi koskea, tuntea ja kokeilla tuotteita niin kuin he tekisivät fyysisessä kaupassa ennen ostopäätöstä. Tuotteen ominaisuuksien ymmärtämisen kannalta on tärkeää, että asiakas saa tuotesivulta riittävän tiedon ostopäätöksen tekemiseen, ja yksityiskohtaisen tuoteinformaation esittäminen onkin välttämätöntä verkkokaupan menestymisen kannalta [Jiang & Benbasat, 2007b]. Niin yritykset, jotka toimivat verkossa, kuin siellä käyvät kuluttajatkin ovat huolestuneita siitä, miten tuotteiden tiedot esitetään verkkosivuilla ja miten asiakkaat ne ymmärtävät [Burke, 2002]. Szymanski ja Hise [2000] selvittivät tutkimuksessaan, että tuotteen tietojen esitys vaikuttaa merkittävästi asiakkaiden tyytyväisyyteen verkkokaupankäynnissä. Huono tai epäselvä tuotekokemus ajaa asiakkaat helposti toiseen verkkokauppaan. Samoin myös asiakkaan kuvitelma siitä, että tuote-esittely on tarpeeksi informatiivinen, parantaa heidän suhtautumistaan verkkokauppaan [Jiang and Benbasat, 2007b].

Suuri osa verkkokaupoista yhä esittää tuotteidensa tiedot pelkillä kuvilla ja teksteillä [Lightner and Eastman, 2002]. Tekstiä käytetään kertomaan tuotteen ominaisuuksista, kuten koosta tai painosta. On kuitenkin selvää, ettei pelkällä

tekstillä ja kuvilla pystytään hyvin kuvailemaan tuotteen koettavia ominaisuuksia [Jiang and Benbasat, 2007b], esimerkiksi sitä, miltä kangas tuntuu tai miten jokin laite reagoi käytettäessä. Kehittyvien teknologioiden vuoksi myös tuote-esittelyillä on mahdollisuus kehittyä vastaamaan jatkuvan kysynnän tarpeita.

Multimedia ja interaktiiviset teknologiat mahdollistavat yrityksille tuotteiden esittelyn monilla eri tavoilla. Tuotteesta voi olla kuvien lisäksi esimerkiksi videoita, ääntä tai 3D-esityksiä. Lisäksi asiakkaan voi antaa vaikka kokeilla tuotteen käyttöä virtuaalisesti. Kim ja kumppanit [2007] kertovat esimerkkejä eri määrstä interaktiivisuutta. Kaksiulotteinen, klikkaamalla suurentuva kuva antaa käyttäjälle vähän vuorovaikutteisuutta. Tuotteiden yhdistelymahdollisuus tai yksityiskohtiin tarkentaminen luovat jo enemmän interaktiivisuutta. Uusimmat teknologiat, joissa käyttäjä voi virtuaalisesti kokeilla tuotteen toimintoja, luovat jo erittäin suuren vuorovaikutuksen asiakkaan ja tuotteen välille. Tutkimukset ovat osoittaneet, että virtuaalitekniologioiden ja multimedian käyttö tuotteiden esityksissä parantaa asiakkaan suhtautumista verkkokauppaan ja kasvattaa ostopäätöksen tekemisen todennäköisyyttä [Schlosser, 2003; Izak and Benbasat, 2007]. Erään tutkimuksen mukaan jopa 98 % ihmisistä, jotka olivat kiinnostuneita ostamaan tuotteita verkkokaupoista, suosivat jonkinlaisia multimediaesityksiä tai demoja pelkän tekstin sijaan [Manvi and Venkataram, 2005].

Monet yritykset ovat jo alkaneet käyttää uusia teknologioita verkkosivullaan. Esimerkiksi navigaattorien valmistajayritys TomTom (www.tomtom.com) on laittanut internetsivuilleen navigaattorimallien tietojen ja kuvien lisäksi videoita eri ominaisuuksista, kuten edistyneestä kaistaopastuksesta. Videosta käyttäjät näkevät, miten tuotetta käytetään ja miten se toimii tietyissä tilanteissa. Samoin monilta autonvalmistajien sivuilta on löydettävissä useita videoita, joissa kuuluu selkeästi auton ääni ja näkyy miten auto liikkuu. Näin käyttäjät kuulevat auton oikeat äänet ja pystyvät paremmin ”kokemaan”, miltä auto tuntuu, näyttää ja kuulostaa. Kameravalmistaja Kodakin sivuilla (www.kodak.com) käyttäjä voi interaktiivisen demon avulla kokeilla, miten kamera käyttäytyy eri toiminnoissa. Lisäksi kameraa voi käänellä miten haluaa ja tarkastella sitä eri suunnista. Näiden esimerkkien kaltaiset tuote-esittelyt vangitsevat käyttäjän tutkimaan tuotetta pidempään ja siten saamaan tuotteesta enemmän yksityiskohtaista tietoa [Jiang *et al.*, 2005].

2.2. Tuotekuvan ominaisuuksia

Tuotekokemusta pohdittaessa on oleellista ymmärtää se, että asiakkaat eivät välttämättä osta varsinaisesti tuotetta verkkokaupasta, vaan vain tuotekokemuksen tuotteesta [Li *et al.*, 2001]. Esimerkiksi asiakas ei ole televisiomainok-

sessä haistanut, miltä hajuvesi tuoksuu, vaan nähnyt minkälaiselle elämäntyyliin tuoksu sopii ja minkä tyylinen tuoksu on. Kun asiakas tutkii tuotetta verkkokaupassa, hän miettii minkälaisen tuote-elämyksen se voisi hänelle tarjota ja miten hyvin tuote pystyy tarjoamaan odotetun elämyksen [Hoch and Deighton, 1989].

Tuote-esittelyt tulee suunnitella niin, että ne eivät johda asiakasta harhaan, vaan antavat mahdollisimman oikeellisen kuvan tuotteesta. Tuotekuvan laadukkuutta voidaan osaksi mitata sillä, kuinka elävä se on, eli kuinka paljon virtuaalinen tuote muistuttaa varsinaista fyysistä tuotetta [Jiang and Benbasat, 2007b]. Elävyyteen liittyy nonverbaalista kieltä, ja sitä voi havainnoida eri aisteilla [Lim *et al.*, 2000]. Elävät tuotekuvat kertovat käyttäjälle enemmän, koska ne esittävät tuotteen konkreettisemmin, kuin suuripiirteinen ja vaisu kuva [Lim *et al.*, 2000]. Siitä seuraakin, että mitä elävämpi tuotekuva on, sitä enemmän tietoa se kertoo asiakkaalle. Jiang ja Benbasat [2007b] osoittivat, että elävyys on yli kaksi kertaa niin tehokas välittämään tuotetietoa asiakkaalle kuin interaktiivisuus. NykYTEknologioilla elävyyden luominen on lisäksi yleisesti paremmin tuettua kuin interaktiivisuuden lisääminen. Esimerkiksi Macromedia Flash ja Apple Quicktime -teknologiat mahdollistavat melko helpolla tavalla multimedial, kuten animaation, äänen tai videon, lisäämisen internetsivuille.

Hochin ja Deightonin [1989] mukaan tuote-esittelyyn voi tutustua kahdella eri tavalla: passiivisesti tutkien tai aktiivisesti etsien. Tutkijoiden mukaan aktiivisessa etsimisessä käyttäjä on motivoituneempi oppimaan tuotteen tietoja ja hänen huomionsa ja sitoutumisensa saadaan heräämään helpommin. Näin käyttäjälle jää paremmin muistiin hänen tekemänsä asiat kuin passiivisesti katsomalla [Hoch and Deighton, 1989]. Kun käyttäjä tutkii itse aktiivisesti tuotetta, hänen mieleensä jää paremmin malli siitä, miten tuote toimii. Käyttäjän on tällöin usein helpompi hahmottaa tuotteen ominaisuudet ja toiminnot kuin vain katsomalla [Jiang and Benbasat, 2007a]. Hän oppii tekemällä, tutkimalla, erehtymällä ja liittämällä uutta tietoa jo mielessä oleviin malleihin. Tämän perusteella interaktiiviset tuotekuvat opettavat käyttäjää enemmän, sillä käyttäjä toimii niissä itse aktiivisesti.

Jotta kahta tai useampaa erilaista tuote-esittelyä voi luotettavasti vertailla, on tuotteiden oltava ominaisuuksien määrältään suurin piirtein samanlaisia. Jahng ja kumppanit [2000] havaitsivat, että monimutkaisen tuotteen suuret määrät ominaisuuksia voivat vaatia enemmän aikaa tiedon prosessointiin ja havainnointiin kuin yksinkertaisemman tuotteen vähemmät ominaisuudet. Siten tuotteen ymmärtäminen olisi haastavampaa ja monimutkaisempaa, mikäli ominaisuuksien määrä on suuri. Esimerkiksi videota seurattaessa käyttäjän täytyy keskittyä ja pysyä tarkkaavaisena koko videon ajan ymmärtääkseen tuot-

teen toiminnan, kun taas kuvia katsellessa käyttäjä voi itse tauottaa toimimisen ja edistyä omaan tahtiin [Jiang and Benbasat, 2005]. Kun tuotteen monimutkaisuus ja ominaisuuksien määrä kasvavat liian suureksi, ei käyttäjä enää jaksakaan keskittyä pitkään videoon ja tiedon ymmärtäminen huononee. Myös interaktiivinen esittely monimutkaisesta tuotteesta suuremmalla todennäköisyydellä uuvuttaa käyttäjän, jolloin hän ei jaksakaan tai kykene sisäistämään kaikkea näkemäänsä tietoa [Jiang and Benbasat, 2007]. Käyttäjä ei myöskään välttämättä käy läpi kaikkia ominaisuuksia, jotka hän esimerkiksi videossa näkisi.

3. Erilaisia tuotekuvia

3.1. Esimerkkituotteet

Valitsin esimerkkituotteiksi eri verkkokaupoista kolme erilaista tuotetta, jotka ovat suurin piirtein yhtä monimutkaisia ja ominaisuuksien määrältään samantyyppisiä. Yksinkertaisimmalla tuotesivulla on vain kuva ja teksti. Toinen tuote esitetään kuvan ja tekstin lisäksi ääninäytteillä ja 3D-kuvalla, jossa tuotetta voi tarkastella eri kulmista. Viimeisen tuotteen sivulla on nähtävänä useampia videoita tuotteen kuvan ja tekstin lisäksi.

Ensimmäinen tuote on kansalaisboksi Kombotallennin, eli tallentava digiboksi. Tuotteen jälleenmyyjä ja verkkokauppa, joista tuote-esittelyä tutkitaan, on Verkkokauppa.com [Verkkokauppa.com, 2009]. Tuotteen pieni kuva on sivun vasemmassa laidassa. Kuvaa on mahdollisuus klikata, jolloin saman kuvan hyvin suuri ja tarkka versio avautuu uuteen ikkunaan. Pienen kuvan oikealla puolella on runsaasti tekstiä, jossa on tuotteen tarkat tekniset tiedot. Tuotesivu on suomenkielinen.

Toinen tuote on Thomann SP-5500 kosketinsoitin. Tätä tuotetta myy Thomann Cyberstore [Thomann, 2009]. Tuotteesta on kuva ja sen alapuolella ovat tuotteen tekniset tiedot. Sivun muuten suomenkielinen, mutta tekniset tiedot ovat englanniksi. Tuotesivulla on linkki, josta aukeaa ikkuna tarkempaan tuotekuvaan. Tässä ikkunassa voi tuotteen kuvaa tarkastella yksityiskohtaisemmin, tarkentaa haluttuihin kohtiin, pyöritellä ja tutkia tuotetta mistä kulmasta tahansa. Lisäksi tuotteesta on kolme erilaista ääninäytettä asiakkaan kuultavaksi.

Viimeisen tuotteen esittely on kaikista monipuolisin. Tuotteen nimi on Kindle Wireless Reading Device ja sitä myy Amazon [Amazon, 2009]. Laitteesta on useita kuvia sekä paljon englanninkielistä tekstiä. Lisäksi tuotesivulla on kolme videota: kaksi laitteesta itsestään ja yksi, jossa muut asiakkaat kertovat tuotteesta mielipiteitään. Tuotteesta on myös 3D-esitys, jossa tuotetta voi pyöritellä ja tarkastella lähempää.

3.2. Teksti ja kuvat

Lightner ja Eastman [2002] tutkivat asiakkaiden tyytyväisyyttä tuote-esittelyihin vertailemalla esittelyjä, joista ensimmäisessä oli pelkkää tekstiä, toisessa pelkkä kuva ja kolmannessa molemmat. Nykypäivänä heidän tutkimustuloksensa ovat kuitenkin jo melko triviaaleja. Harvassa kaupassa on enää pelkkää tekstiä tai pelkkää kuvaa, vaan suurimmassa osassa on sekä teksti että kuva. Tulokset osoittivat, että käyttäjät suhtautuvat tuotteeseen myönteisemmin, jos tuote-esittelyssä on suunnilleen tasavertaisesti yksityiskohtaista (search) tietoa tekstimuodossa sekä kokemusperäistä (credence) tietoa, kuten esimerkiksi ulkonäkö tai muiden käyttäjien kokemukset [Lightner and Eastman, 2002]. Koehenkilöiltä pyydettiin vapaamuotoisia kommentteja parannusehdotuksiksi. Pelkän kuvallisen tuote-esittelyn saamia kommentteja oli 28, joista 23 toivoivat lisätietoja tuotteesta. Tekstimuodossa olevan esittelyn kommentteja oli 36, joista 24 toivoivat näkevänsä tuotteen kuvan. Kaikista tyytyväisimpiä olivat ne koehenkilöt, jotka näkivät sekä kuvan että tekstin tuotteesta. Näin he saivat sekä tekstimuodossa olevan tiedon, kuten tuotteen nimen ja ominaisuudet, että näkivät, miltä tuote näyttää.

Kun käyttäjä etsii tietoa haluamastaan tuotteesta ostopäätöksen tekemistä varten, haluaa hän ensisijaisesti nähdä tekstimuodossa tuotteen ominaisuudet ja mielellään omalla äidinkielellään [Lightner and Eastman, 2002]. Kuvat sisältävät kokonaisvaltaista ja kokemusperäistä tietoa, josta ei löydy tarvittavaa tietoa, kuten esimerkiksi yksityiskohtaisia teknisiä tietoja [Lightner and Eastman, 2002].

Kombotallentimen tuote-esittelyssä kuvalla on hyvin vähän merkitystä asiakkaan tuotteen ymmärtämisen kannalta, koska oletettavasti suurin osa kuluttajista tietää suurin piirtein, miltä digiboksi näyttää. Tuoteryhmissä, joissa tuotteet näyttävät hyvin pitkälti samalta, on käyttäjälle oleellista pystyä vertailemaan yksityiskohtaisia tietoja. Teknisissä laitteissa, joissa tärkein informaatio on koneen kuoren sisällä, ei kuvalla ole juurikaan merkitystä. Tällöin on oleellista, että käyttäjälle kerrotaan yksityiskohtaiset tiedot tekstimuodossa, josta ne on helppo lukea tai silmäillen etsiä itseä kiinnostava yksityiskohta.

3.3. Video

Nykyään monessa verkkokaupassa on tekstin ja kuvan lisäksi myös videoita tuotteen toiminnasta. Video koostuu ikään kuin useista paikallaan olevista kuvista, jotka näkyvät hyvin nopeasti peräkkäin. Kuvan ja videon oleellinen ero onkin siinä, että yhdessä kuvassa käyttäjä näkee vain yhden kuvan, mutta videossa hän näkee niitä useita, jotka kaikki antavat lisätietoja tuotteesta. Tästä onkin pääteltävissä, että video hyödyntää käyttäjää eniten silloin, kun kaikki

videolla näkyvät kuvat ovat oleellisia tuotteen tiedon esittämisen kannalta [Jiang and Benbasat, 2007a]. Lisäksi videolla on mahdollista esittää ääntä, jolla voi antaa vielä lisää hyödyllistä tietoa tuotteen toiminnasta. Jiang ja Benbasat [2005] huomasivat, että videot auttavat kuvia paremmin asiakkaita ymmärtämään tuotteiden toimintaa ja siten rakentamaan positiivisia mielikuvia verkkosivuja kohtaan.

Jos tuotteen dynaamiset ominaisuudet esitetään usean kuvan sarjana, käyttäjä pakotetaan luomaan tuotteen toiminnasta mielessään malli, joka voi vaihdella esityksen tyylin ja kuvien lisäksi tekstimuodossa olevien selitysten mukaan [Jiang and Benbasat, 2007]. Valmis video samasta toiminnasta antaa käyttäjälle valmiin mallin, eikä käyttäjän tarvitse itse päätellä kuvien perusteella, miten toiminto tapahtuu. Video paitsi nopeuttaa toiminnon ymmärtämistä, se myös poistaa mahdolliset virhetulkinnat, jotka voivat aiheutua kuvien ja tekstien erityyillisistä esittämisistä.

Kindle Reading Devicen esittelyssä videot ovat oleellisessa osassa tuotteen tietojen ymmärtämisessä. Sivulla on niin runsaasti tekstimuotoista tietoa laitteesta, että on vaikea päätellä, mikä tieto on kaikista oleellisinta. Verkkokauppa selvästi luottaa videoihin, sillä heti vasemmassa yläreunassa, siinä missä yleensä on totuttu näkemään kuva tuotteesta, on video. Videossa on englanninkielinen selostus, joka kertoo perustietoja laitteesta. Samalla videolla voi nähdä Kindlen tietoja ja niihin liittyviä kuvia. Esimerkiksi melko alussa on kuva Kindlestä ja vieressä lukee laitteen koko. Samaan aikaan selostus kertoo tuotteen koon myös ääneen. Näin videosta hyötyy myös, vaikka ei olisikaan mahdollista kuunnella selostusta. Läpi videon ruudulle ilmestyy selostuksen kannalta oleellisimmat sanat. Tämä perustietoja kertova video kestää vajaat kaksi minuuttia.

Alhaalla on lisäksi toinen video, kestoltaan yli kuusi minuuttia. Tämän videon tarkoitus ei niinkään ole antaa tietoja laitteesta, vaan vaikuttaa kuluttaja siitä, että Kindle sopisi juuri hänelle. Video markkinoi nimenomaan tuotekokemusta, ei itse tuotetta. Esimerkiksi videon alussa nainen istuu mukavasti rannalla, ottaa Kindlen laukustaan ja alkaa lukea. Videolla on enemmän kokemusperäistä tietoa kuin lyhyemmällä ”tietoisku”-tyylisellä sivun yläreunassa olevalla videolla. Kolmannella videolla verkkokaupan entiset asiakkaat kertovat mielipiteitään Kindlestä. Luonnollisestikaan kenelläkään ei ole mitään pahaa sanottavaa, vaan asiakkaat kertovat, kuinka laite on nopeuttanut ja helpottanut lukemista ja on jopa parempi kuin kirja. Tämä video pyrkii erityisesti vaikuttamaan katsojaa siitä, että se voi tehdä kaikkien lukemisesta vielä miellyttävämpää. Tuotteen sivu on selvästi suunniteltu niin, että ylhäällä on videon muodossa oleellinen tieto siitä, mikä tuote on ja mitä se tekee, ja halutessaan asiakas voi selata alaspäin lukemaan tarkempia tietoja.

3.4. Interaktiiviset ominaisuudet

Fyysistä tuotetta ostaessaan asiakkaalle on tärkeää nähdä tuote edessään ja päästä tutkimaan sen muotoja ja materiaalia eri kulmista [Li *et al.*, 2001]. Tämän tyyppiset toiminnot voidaan simuloida elävästi 3D-ympäristössä, jossa käyttäjä voi vapaasti tutkia, käänellä, pyöritellä ja tarkentaa tuotteen yksityiskohtiin. Asiakas saa sen vaikutelman, että tuote olisi fyysisesti läsnä tilanteessa. Fiore ja Jin [2003] huomasivat, että 3D-ympäristö ja tuotteiden yhdistelymahdollisuus antoivat asiakkaalle enemmän visuaalista tietoa myös esimerkiksi siitä, miten tuotteet sopivat keskenään yhteen. Näin interaktiivinen tuote-esittely pystyy antamaan kuvia ja videoita enemmän samaa tietoa, kuin mitä asiakas saisi ostaessaan tuotetta tavallisesta kaupasta.

Interaktiivinen tuotekuva eroaa tavallisista kuvista ja videosta siinä, että interaktiivisessa esittelyssä käyttäjä voi passiivisen katselemisen lisäksi tutkia itsenäisesti tuotteesta haluamiaan tietoja. Käyttäjä voi itse valita tuotteen tutkimisen nopeuden ja tahdin. Hän voi pitää taukoja, silmäillä muita tietoja välillä ja sitten taas palata interaktiivisen esittelyn pariin tutkimaan uusia kiinnostavia ominaisuuksia. Käyttäjä voi siis etsiä tietoa omalla yksilöllisellä tavallaan. Interaktiivisessa ympäristössä käyttäjä ikään kuin käyttää oikeaa tuotetta ja testaa sen ominaisuuksia, aivan kuin tuote olisi fyysisesti läsnä [Jiang and Benbasat, 2007].

Schlosser [2003] huomasi tutkimuksissaan, että interaktiivisuus tuote-esittelyssä helpottaa käyttäjää muodostamaan mielikuvan tuotteesta ja siitä, miten sitä käytetään. Voisikin olettaa, että käyttäjä ymmärtää tuotteen toiminnan paremmin interaktiivisessa ympäristössä kuin pelkkien kuvien tai videon perusteella. Interaktiivisten ominaisuuksien ja videoiden välillä ei ole kuitenkaan löydetty merkittävää eroa tiedon välittämisessä. Tutkimukset [Jiang & Benbasat, 2005] ovat osoittaneet, että valmiit videoleikkeet eri ominaisuuksista ovat tarpeeksi tehokkaita saamaan asiakkaat ymmärtämään tuotteen toimintaa, eikä interaktiivisia ominaisuuksia välttämättä tarvita.

Interaktiivinen tuote-esittely toimii parhaiten silloin, kun tuotteen ominaisuuksien määrä ei nouse liian suureksi. Jiang ja Benbasatin [2005] tutkimuksessa ominaisuuksien määrältä suuremman tuotteen kaikista ominaisuuksista koehenkilöt kävivät läpi interaktiivisessa tuotekuvassa vain 76 %, kun yksinkertaisemmasta tuotteesta tutustuttiin peräti 92 %:iin ominaisuuksista. Koska käyttäjä toimii interaktiivisessa tuotekuvassa aktiivisemmin, hän oppii kuitenkin paremmin niistä ominaisuuksista, jotka hän käy läpi.

Thomannin kosketinsoittimien tuote-esittelyssä on mahdollista tutkia kuvaa 3D-mallin kautta eri kulmista, tarkentaa yksityiskohtiin ja pyöritellä tuotetta. Käyttäjä voi tarkastella kosketinsoitinta aivan kuin se olisi hänen edessään

kaupassa. Kuvat ovat tarkkoja ja käyttäjä voi myös valita oikean reunan valmiista listasta eri osien kuvia lähempään tarkasteluun. Kuvan pyörittely on aluksi hieman hankalaa, kunnes siihen tottuu. 3D-kuvan lisäksi tuotteen tekniset tiedot on listattu tuotesivulle. Käyttäjällä on myös mahdollisuus kuunnella neljää erilaista ääninäytettä kosketinsoittimen toiminnasta. Ääninäytteet on integroitu hyvin käyttöliittymään ja ne voi kuunnella heti tuotesivulta ilman, että tarvitsee ladata tiedostoa ensin omalle koneelle ja sitten vasta kuunnella. Kindle Wireless Reading Devicen sivulla on myös 3D-esitys tuotteesta, mutta koska videot olivat niin kattavia, ei esitys tuo käyttäjälle uutta tietoa tuotteesta.

4. Keskustelu

Empiiriset tutkimukset [Suh and Lee, 2005; Jiang and Benbasat, 2007a] osoittavat, että edistyneet multimediateknologiat, kuten videot ja interaktiiviset ominaisuudet, ovat tehokkaita välittämään kokemuksen kautta ymmärrettäviä ominaisuuksia, esimerkiksi ääntä tai dynaamista toimintaa. Sen sijaan yksinkertaiset esittelyt, joissa on vain kuvia ja tekstejä, ovat parempia kertomaan asiakkaalle teknistä tietoa. Mikäli kuvien lisäksi tuotteen esittelyssä on mukana ääntä, voi se tuotteesta riippuen parantaa merkittävästi käyttäjän tuotteen ymmärrystä.

Yleisesti tuotteiden esittelyssä tutkimukset puoltavat videoiden ja interaktiivisten teknologioiden käyttöä kuvien sijaan [Jiang and Benbasat, 2007b]. Interaktiivisten ominaisuuksien ja videoiden välillä ei ole löydetty merkittävää eroa siinä, missä määrin tieto välittyy asiakkaalle. Poikkeuksen muodostavat erittäin monimutkaiset tuotteet, joissa ominaisuuksien määrä nousee hyvin suureksi. Tällöin on parasta välttää monimutkaisia vuorovaikutteisia esittelyjä ja suosia kuvia ja videoita [Jiang and Benbasat, 2007a].

Elävyyden lisääminen videoilla, äänillä tai kuva-animaatioilla on tutkimusten mukaan tehokkaampaa välittämään tietoa kuin interaktiiviset ominaisuudet [Jiang and Benbasat, 2007b]. Kun ottaa huomioon, että vuorovaikutteisten ominaisuuksien hankkiminen verkkokauppaan on todennäköisesti kalliimpaa ja tehottomampaa, Jiang ja Benbasat [2007a] suosittelevatkin yrityksiä hankkimaan hyvälaatuisia videoita tuote-esittelyihin vuorovaikutteisten teknologioiden sijaan.

Tämän tutkimuksen esimerkkituotteiden tuote-esittelyiden arvioinnit puoltavat aikaisempia tutkimuksia. Kaikki tuotteet ovat olleet teknisiä tuotteita, joissa on ollut runsaasti yksityiskohtaista tietoa. Tekstimuotoinen informaatio tuotesivuilla on ollut välttämätön kaikilla näillä tuotteilla. Ainoastaan Kindle Wireless Reading Devicen esittelyssä videot auttoivat jäsentelemään tietoa, jota tekstimuodossa oli hyvin paljon. Kombotallentimen kohdalla teksti ja kuva oli-

vat tarpeeksi kattavat kertomaan tuotteen toiminnot ja ominaisuudet asiakkaalle. Koska lähes kaikilla ihmisillä on nykyään tietoa siitä, mikä digiboksi on, ei sitä tarvinnut tuote-esittelyssä selittää. Riitti, että listataan ne ominaisuudet, jotka erottavat juuri tuon digiboksin muista vastaavista.

Ääniä oli kuunneltavissa kosketinsoittimen esittelyssä sekä Kindlen esittelyssä videon muodossa. Kosketinsoittimen kohdalla on varmasti hyvin tärkeää, että asiakas pääsee kuuntelemaan soitinta, onhan kyseessä musiikin soittamisen väline. Oli myös hyvä, että vaihtoehtoina oli neljä erityyppistä ääninauhaa, jotka antoivat käyttäjälle kuvan siitä, mitä erilaisia ääniä tuotteella voi soittaa. Kosketinsoittimessa oli nähtävillä runsaasti tarkkoja kuvia eri kulmista sekä lisäksi 3D-esitys, jossa katsoja voi itse tarkastella tuotetta mistä kulmasta ja kuinka läheltä tahansa haluaa. Esitys oli hieman vaikeakäyttöinen ja mielestäni 3D-kuva ei ollut lainkaan välttämätön, kun nähtävillä oli muutenkin runsaasti kuvia eri kulmista. Mikäli kyseessä olisi ollut suurempi esine, esimerkiksi auto, olisi 3D-esityksen merkitys ollut aivan toinen. Tilanne olisi ollut sama, jos muita kuvia ei olisi ollut yhtä runsaasti tai ne eivät olisi olleet yhtä eläviä ja hyvälaatuisia. Nyt kuitenkin tuntui siltä, että esitys oli mukava lisä, mutta ei varsinaisesti tuonut käyttäjälle suurta lisäinformaatiota tuotteesta.

Kindle Wireless Reading Device on tuotteena varmasti tuntemattomin ja olikin hyvä, että siitä oli saatavilla kattavat videot tuotteen sivulla. Tuotteen sivu oli kokonaisuudessaan hyvin pitkä, mikä teki olennaisen tiedon löytämisen vaikeaksi. Ilman videoita olisi ollut hankalaa löytää sivulta tieto siitä, mitä laite oikein tekee. Videossa oli tietoa sekä kuunneltavana että luettavana. Näin asiakas voi tutustua tuotteeseen, vaikka hän ei pystyisikään hyödyntämään sekä näkö- että kuuloaisteja. Lisäksi asiakkaalla oli valittavanaan kaksi tuotteesta kertovaa videota, yksi lyhyt ja ytimekäs ja toinen, jossa kerrottiin laitteen ominaisuuksista tarkemmin. Tämän tuotteen kohdalla 3D-esitys oli melko turha, koska videot kertoivat jo kaiken oleellisen tuotteesta. 3D-kuvalla ei ollut mitään uutta tietoa kerrottavana asiakkaalle.

Sekä aikaisemmat tutkimukset että tässä tutkielmassa esitetyt tuotearviot ovat samaa mieltä siitä, mitkä tavat ovat tehokkaimpia välittämään tuotteen tietoa asiakkaalle. Kyseisissä teknisissä tuotteissa kaikista tärkeimpiä olivat tekstimuodossa olevat yksityiskohtaiset tiedot. Tuotteen kuuluessa yleisesti tunnettuun tuoteryhmään, esimerkiksi televisioihin tai digitallentimiin, yksityiskohtainen kuva on riittävä kertomaan, miltä tuote näyttää. Mikäli on oleellista näyttää tuotteesta eri kulmia, kannattaa suosia useampia tarkkalaatuisia kuvia tai videoita. Jos tuote on vähemmän tunnettu tai erittäin kallis, ovat videot hyvä keino kertoa asiakkaalle tuotteen perustoiminnot, jotta asiakkaan ei tarvitse alkaa niitä itse päättelemään kuvista. Videota kannattaa suosia interak-

tiivisten toimintojen sijaan, koska se on yhtä informatiivinen ja helpommin toteutettavissa. Ei kannata pyrkiä liian hienoihin ja monimutkaisiin tuoteesittelyihin, vaan suosia niitä, jotka ovat helppoja käyttää ja kertovat asiakkaalle realistisen ja todenmukaisen kuvan tuotteesta. Näin asiakas ei tule pettymään tilauksen saapuessa, vaan tietää jo etukäteen, minkälaisen tuotteen tulee vastaanottamaan.

5. Yhteenveto

Tässä tutkielmassa käsiteltiin virtuaalista tuotekuvaa ja sitä, miten erilaiset multimedia ja interaktiiviset teknologiat välittävät tuotteen tietoja asiakkaalle. Vaikka tutkielmassa keskityttiin siihen, miten hyvin asiakas tuotteen ominaisuudet ymmärtää, on tärkeää muistaa, että monet muutkin asiat vaikuttavat asiakkaan tuotekokemukseen. Verkkosivujen viihtyvyys, houkuttelevuus ja mainonta ovat kaikki asioita, joita pitää tutkia lisää ennen kuin voi tehdä lopullisia johtopäätöksiä siitä, minkälaisilla tuotekuvilla yritykset pystyisivät parhaiten edistämään myyntiään verkossa. Lisäksi voitaisiin tutkia, vaihtelevatko tulokset, mikäli tutkittavana on eri tuoteryhmään kuuluva tuote kuin mitä tässä käsiteltiin, esimerkiksi vaatteet tai huonekalut. Tässä tutkielmassa rajoituttiin tutkimaan teknisiä laitteita, eikä voikaan yleistää näiden päätelmien koskevan kaikkia muitakin tuoteryhmiä. On hyödyllistä ymmärtää, miten tuotteen tietoja pystyy parhaiten välittämään asiakkaalle, mutta ostotapahtuman syntymiseen vaikuttavat monet muutkin asiat ja se loppujen lopuksi on aina yritysten tavoitte elektronisessa kaupankäynnissä.

Virtuaalisesta tuotekuvasta on tehty paljon tutkimuksia viime vuosina, koska ne myös vanhenevat suhteellisen nopeasti alati kehittyvän teknologian myötä. Uusia tutkimuskysymyksiä tulee jatkuvasti esille. Aiheeseen liittyy lisäksi läheisesti eri tieteiden tutkimusalueita, kuten markkinointia ja asiakkaan psykologista käyttäytymistä. Aihe on niin laaja, että siitä löytyisi varmasti mielenkiintoisia tutkimuksen kohteita myös esimerkiksi pro gradu -tutkielmaa varten, mikäli siinä vaiheessa riittää vielä kiinnostusta laajentaa aiheen tietämystä.

Viiteluettelo

- [Amazon, 2009] Amazon, Kindle Wireless Reading Device, <http://www.amazon.com/Kindle-Wireless-Reading-Display-Generation/dp/B0015T963C/>, Checked 15.12.2009.
- [Burke, 2002] R.R. Burke, Technology and the customer interface: What consumers want in the physical and virtual store, *Journal of the Academy of Marketing Science* 30, 4 (2002), 411-432.

- [Fiore and Jin, 2003] A.M. Fiore and H.-J. Jin, Influence of image interactivity on approach responses towards an online retailer, *Internet Research: Electronic Networking Applications and Policy* 13, 1 (2003), 38-48.
- [Hoch and Deighton, 1989] S.J. Hoch and J. Deighton, Managing what consumers learn from experience, *Journal of Marketing* 53, 2 (1989), 1-20.
- [Jahng et al., 2000] J. Jahng, H. Jain and K. Ramamurthy, Effective design of electronic commerce environments: A proposed theory of congruence and an illustration, *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans* 30, 4 (2000), 456-471.
- [Jiang and Benbasat, 2007a] Z. Jiang and I. Benbasat, The effects of presentation formats and task complexity on online consumers. Product understanding, *MIS Quarterly* 31, 3 (2007a), 475-500.
- [Jiang and Benbasat, 2007b] Z. Jiang and I. Benbasat, Research note - investigating the influence of the functional mechanisms of online product presentations, *Information Systems Research* 18, 4 (2007b), 454-470.
- [Jiang et al., 2005] Z. Jiang, W. Wang and I. Benbasat, Multimedia-based interactive advising technology for online consumer decision support, *Communications of the ACM* 48, 9 (2005), 92-98.
- [Kim et al., 2007] J. Kim, A.M. Fiore and H.H. Lee, Influences of online store perception, shopping enjoyment, and shopping involvement on consumer patronage behavior towards an online retailer, *Journal of Retailing and Consumer Services* 14, 2 (2007), 95-107.
- [Li et al., 2001] H. Li, T. Daugherty and F. Biocca, Characteristics of virtual experience in electronic commerce: a protocol analysis, *Journal of Interactive Marketing* 15, 3 (2001), 13-30.
- [Lightner and Eastman, 2002] J.N. Lightner and C. Eastman, User preference for product information in remote purchase environments, *Journal of Electronic Commerce Research* 3, 3 (2002), 174-185.
- [Lim et al., 2000] K.H. Lim, I. Benbasat and L.M. Ward, The role of multimedia in changing first impression bias, *Inform. Systems Res* 22, 2 (2000), 449-471.
- [Manvi and Venkataram, 2005] S.S. Manvi and P. Venkataram, An intelligent product-information presentation in e-commerce, *Electronic Commerce Research and Applications* 4, 3 (2005), 220-239.
- [Schlosser, 2003] A.E. Schlosser, Experiencing products in the virtual world: the role of goal and imagery in influencing attitudes versus purchase intentions, *Journal of Consumer Research* 30(September) (2003), 184-198.
- [Suh and Lee, 2005] K.-S. Suh and Y.E. Lee, Effects of virtual reality on consumer learning: An empirical investigation in web-based electronic commerce, *MIS Quarterly* 29, 4 (2005), 673-697.

- [Szymanski and Hise, 2000] D.M Szymanski and R.J. Hise, e-Satisfaction: An initial examination, *Journal of Retailing* 76, 3 (2000), 309-322.
- [Thomann, 2009] Thomann Cyberstore, Thomann SP-5500, http://www.thomann.de/fi/thomann_sp5500.htm, Checked 15.12.2009.
- [Verkkokauppa.com, 2009] Verkkokauppa.com, Kansalaisboksi kombotallenin, <http://www.verkkokauppa.com/popups/prodinfo.php?id=6238>, Luetu 15.12.2009.

Ohjaaja: Taina Kaapu

Second Life liiketoiminnallisena toimintaympäristönä

Mervi Ollikainen

Tiivistelmä.

Tässä tutkielmassa käsitellään virtuaalimaailmojen, tarkemmin Second Lifen, liiketoimintaa. Web 2.0:an on kehittynyt uudenlaista liiketoimintaa, varsinkin ns. osallistumistaloutta, jonka toimivuutta virtuaalisessa ympäristössä tarkastellaan. Pohditaan myös, minkälaisia riskejä ja hyötyjä liiketoiminnalla on, sekä sen tulevaisuutta uudessa ympäristössä. Esille tulee lainsäädännöllisiä kysymyksiä esimerkiksi tekijänoikeuksista sekä mahdollisesta verotuksesta. Ohjelmoijatkin joutuvat ottamaan enemmän vastuuta ohjelmistojen turvallisuudesta rahaliikenteen vuoksi. Toisaalta virtuaalimaailmat tarjoavat paljon uusia mahdollisuuksia sekä yrityksille että käyttäjille.

Avainsanat ja -sanonnat: Virtuaalimaailma, Web 2.0, liiketoiminta, Second Life, osallistumistalous.

CR-luokat: H.5.1, J.4, K.4.4, K.5.1, K.5.2

1. Johdanto

Internet oli alkuaikoinaan enemmänkin tiedonjakelukanava kuten televisio, johon harvat tuottivat sisältöä. Verkko on kasvanut räjähdysmäisesti niistä ajoista ja samalla se on 2000-luvun puolella muuttunut toiminnaltaan täysin erilaiseksi – sosiaalisesti mediaksi. Nykyistä Internetiä kutsutaan usein Web 2.0:ksi. Sosiaalinen media on antanut kenelle tahansa mahdollisuuden tuottaa sisältöä ja jakaa sitä kenelle tahansa. Erilaiset firmat ovat omaksuneet uusia tapoja lähestyä kuluttajia ja uusia yrityksiä on syntynyt palvelemaan nousseita tarpeita.

Tässä tutkielmassa käsitellään Web 2.0:n syntyä liiketaloutta. Keskityn Second Life -nimiseen virtuaalimaailmaan, joka simuloi reaalia maailmaa ilman peleille tyypillisiä tavoitteita. Koska Second Lifessa toimivat lähes samat säännöt kuin reaalia maailmassa, siihen on syntynyt monenlaista liiketoimintaa. Tarkastelen, kuinka perinteiset reaalia maailmassa pitkään olleet liiketoimintamallit toimivat virtuaalimaailmassa, sekä mitä uutta liiketoimintaa on syntynyt Internetin tarjoamien uusien mahdollisuuksien myötä.

Virtuaalimaailmoja, erityisesti Second Lifea, on tutkittu laajasti viime vuosina. Kiinnostuksen kohteena ovat olleet myös uudentyyppiset liiketalousmallit, perinteisen liiketoiminnan, kuten markkinoinnin, toimivuus ja tulevaisuus erilaisessa ympäristössä. Esillä ovat olleet myös virtuaalimaailmoissa käytävään liiketoimintaan liittyvät riskit ja haasteet, kuten yksityisyyskysymykset ja verotus, sekä hyödyt. Syitä siihen, minkä takia ihmiset ostavat virtuaalisia asi-

oita, on myös tutkittu. Onko virtuaalimaailmojen talous vain hetken villitys ja ylittääkö tai peräti korvaako uudentyyppinen liiketalous perinteisen liiketalouden? Tämä tutkielma kokoaa yhteen virtuaalimaailmojen liiketalouteen keskittyneitä aiheita tarjotakseen niistä kokonaiskuvan.

Toisessa luvussa käsitellään Web 2.0:n mukana kehittynyttä osallistumistaloutta, joka haastaa perinteiset hierarkkiset liiketalousmallit. Kolmannessa luvussa tutkitaan, mitä liiketalousmalleja virtuaalimaailmoissa on ja mitkä niistä kuuluvat osallistumistalouteen. Neljännessä luvussa siirrytään tarkastelemaan lähemmin Second Life -virtuaalimaailman liiketaloutta ja viidennessä luvussa arvioidaan virtuaalimaailmoissa tapahtuvan rahaliikenteen riskejä sekä hyötyjä. Kuudennessa luvussa pohditaan liiketalouden tulevaisuutta virtuaalimaailmoissa. Lopuksi seitsemännessä luvussa tehdään yhteenveto tässä tutkielmassa käsitellyistä asioista.

2. Web 2.0 – Osallistumistaloudet

Osallistumistaloudella tarkoitetaan hyödykkeiden tuotantoa, jakelua, vaihtoa ja kulutusta, joka syntyy sosiaalisen median piirissä [Melakoski et al., 2007]. Esimerkiksi monille tutulle Sims-pelisarjalle, elämysimulaattorille, pelaajat ovat tuottaneet omia asuja ja hiustyylejä yms., joita toiset pelaajat ovat voineet ladata omille simeilleen eli pelattaville hahmoilleen. Osa hankkii jopa tuloja tekemällä virtuaalisia hyödykkeitä. Osallistumistalous tehdään tutummaksi seuraavissa kappaleissa, joissa käsitellään sen eri osa-alueita: crowdsourcingia, joukkotuotantoa ja jakelua sekä virtuaalisia hyödykkeitä ja ammatteja.

Crowdsourcingilla tarkoitetaan työn ulkoistamista määrittelemättömälle ihmisjoukolle avoimella kutsulla. Hintikan [2008, s. 35] mukaan voidaan vaihtoehtoisesti crowdsourcing määritellä avoimen lähdekoodin periaatteiden soveltamiseksi ohjelmistokehityksen ulkopuolella. Tavanomaisin esimerkki lienee yrityksen kilpailu, jossa etsitään uutta logoa. Näin yritykset saavat ideansa kehitettyä ja käyttöön nopeasti, sen sijaan, että joutuisivat kehittämään kaiken alusta asti itse. Myös tarjonta on suurempi kuin omassa organisaatiossa [Hintikka, 2008], joten yritykset voivat saada uusia ideoita yrityksen organisaation ulkopuolelta. Se tarjoaa myös kenelle tahansa mahdollisuuden osallistua yrityksen toimintaan ja saada sitä kautta esimerkiksi parhaimmassa tapauksessa työtarjoja.

Kolmanteen osallistumistalouden osaan kuuluvat synteettiset eli virtuaaliset hyödykkeet ja ammatit. Aiemmin mainitut Sims-pelisarjaan tehdyt hyödykkeet ovat osa tätä taloutta. Virtuaalisista ammattiteistäkin, kuten konsultanteista, löytyy monia esimerkkejä Second Life -virtuaalimaailmasta.

Osallistumistalous tavoittaa kohderyhmät, joita ei muuten tavoitettaisi. Se on luonut uusia tuotantomalleja (massaulkoistaminen), jakelumalleja (vertaisverkot), ansaintamalleja, markkinoita uudentyypisille tuotteille ja palveluille (virtuaaliset hyödykkeet) sekä ammatteja ja toimialoja. Lyhyesti sanottuna osallistumistalous eroaa perinteisestä tekemällä kuluttajasta passiivisen vastaanottajan sijaan aktiivisen ja kehitykseen osallistuvan.

3. Liiketoimintamallit ja niiden toimivuus virtuaalimaailmassa

Ennen osallistumistaloutta Internetiin levisi perinteistä taloutta, kun huomattiin verkon suosio ja sitä kautta tavoitettavissa olevien ihmisten määrä. Perinteisellä taloudella tarkoitetaan tässä tutkielmassa hierarkkista liiketaloutta, jossa kuluttaja nähdään pelkkänä ostajana, jota yritys yrittää houkuttaa asiakkaakseen markkinoimalla tuotteita tai palveluja. Kuluttaja ei siis osallistu tuotteiden ja palvelujen kehitykseen millään lailla. Virtuaalimaailmoista, kuten Second Lifesta, löytyy sekä perinteistä että osallistumistaloutta, jonka Noam [2007] jakaa kahdeksaan eri liiketoimintamalliin.

Ensimmäisessä liiketoimintamallissa markkinoidaan reaali maailman hyödykkeitä ja palveluja. Markkinointi ei kuitenkaan välttämättä toimi virtuaalimaailmoissa kuten Second Life, sillä kaikki käyttäjät eivät ole aina sisäänkirjautuneena ja virtuaalimaailman koko on suuri. Samoin muissa virtuaalimaailmoissa, esim. World of Warcraftissa, ongelmana on myös se, että esimerkiksi uuden auton mainostaminen fantasiamaailmassa särkisi pelin tunnelman. Toinen Noamin malli on muuten sama kuin ensimmäinen, mutta siinä myydään virtuaalisia hyödykkeitä ja palveluja. Malli voi olla kannattava, jos pystyy tekemään ainutlaatuisen tuotteen. Käsittelen tarkemmin markkinointia ja immateriaaleja luvussa neljä.

Kolmannessa liiketoimintamallissa myydään palveluja ja työkaluja virtuaalisten tuotteiden tekijöille ja myyjille. Yksi esimerkki tästä on ItemBay, yritys joka varmistaa turvallisen rahaliikenteen käyttäjien välillä, kun virtuaalisia hyödykkeitä myydään ja ostetaan. Neljännessä mallissa on myös kyse uudentyypisistä palveluista, kuten virtuaalimaailmassa tapahtuvasta koulutuksesta tai markkinatutkimuksesta. Muun muassa Harvardin oikeustieteellisessä tiedekunnassa osa "CyberOne"-kurssin opetuksesta tapahtuu Second Lifessa [Bray and Konsynski, 2007].

Mediasisällön jakaminen, eli viides Noamin liiketoimintamalli, on virtuaalimaailmoissa yleistä. Media voi olla esimerkiksi musiikkia ja videoita. Varsinkin aloittelevat artistit yrittävät tehdä itseään tunnetuksi usein verkon kautta, ja se voi olla kannattavaakin. Liiketoimintamallina se on riskialtista, koska sisältö voi olla mitä vaan.

Kuudennen mallin osalta Noam [2007] toteaa, että virtuaalinen talous on kulutustalous tuottavan talouden sijasta. Kukaan ei esimerkiksi pese ikkunoita, koska ikkunat eivät likaannu eikä kukaan tarvitse lääkäreitä, koska kukaan ei sairastu. Virtuaalimaailmaan kirjaututaan oikeassa elämässä tienatun rahan kanssa ja sitä kulutetaan niin kuin kauppakeskuksissa. Toisin sanoen virtuaalimaailmoissa joko käydään vaihtokauppaa, esitelmöidään tai kulutetaan.

Noamin [2007] mukaan yhteisöllisyydellä houkutteellaan kuluttajia, sillä siihen perustuu mm. Facebook-sivuston suosio. Second Lifessa on olemassa jopa ns. "hengailijoita", joita esimerkiksi virtuaaliympäristössä olevan klubin omistaja voi palkata pyörimään klubilla tai sen lähellä. Tällä tavalla paikasta saadaan mielenkiintoisemman oloinen muille käyttäjille. Tulevaisuudessa käyttäjiä saatetaan kannustaa käymään virtuaalimaailmoissa esimerkiksi maksamalla, mikä voi olla tuottoisaa, jos se houkuttelee muita käyttäjiä.

Viimeisin liiketoimintamalli on virtuaalimaailman omistaminen. Paras tulos saadaan rajoittamalla kilpailua sekä myymällä monopoliasemia. Tällöin hinnat eivät pääse karkaamaan liian alas. Ainoa rajoitus on pitää hinnat kuluttajille järkevinä. Tulevaisuudessa hintoja voitaneen nostaa, jos kynnys virtuaalimaailmasta lähtemiseen kasvaa. [Noam, 2007]

4. Second Life – liiketoiminnan "kehityslaboratorio"

Second Life on Linden Lab -yhtiön luoma ja omistama virtuaalimaailma, joka jäljittelee reaali maailmaa ilman kummempia tavoitteita. Koska virtuaalimaailmassa on lähes samat säännöt kuin oikeassakin, siihen kuuluu myös talous. Monet yritykset, kuten uutistoimisto Reuters, ovat tehneet omia saarekkeita Second Lifeen. Yritysten saarissa pidetään mm. yritystapaamisia, kuten IBM on tehnyt [Bray and Konsynski, 2007]. Lukuisat ja monenlaiset yritykset toimivat Second Lifessa, mutta virtuaalimaailmassa on myös syntynyt yrityksiä kuten Electric Sheep, joka tarjoaa sisältöä kolmiulotteisiin virtuaalimaailmoihin. Liikevaihto Second Lifessa on 1-2 miljoonaa dollaria päivittäin ja siellä on yli 9 miljoonaa asukasta, mikä osaltaan selittää yritysten kiinnostusta [Barnes, 2007].

Linden Lab on kehittänyt rahayksikön nimeltään LindeX, joka on verrattavissa oikeaan valuuttaan. Toisin kuin muissa virtuaalimaailmoissa, LindeX-dollareiden arvoa ei ole kiinnitetty, eli rahan arvo muuttuu aivan kuin oikeilla valuutoilla. Lisäksi LindeX-dollareita voi vaihtaa oikeisiin Yhdysvaltojen dollariin, toisin kuin monen muun virtuaalimaailman valuuttaa [Bray and Konsynski, 2007]. Second Lifen kotisivujen (<http://secondlife.com>) mukaan tällä hetkellä 250 LindeX-dollaria (L\$) vastaa yhtä Yhdysvaltojen dollaria. Oikean rahan liikevaihto tarkoittaa myös sitä, että Second Lifessa toimii pankkeja. Yksi näistä pankeista on MetaBank, joka on täysin virtuaalinen pankki, jossa asiakkaat voi-

vat tallettaa ja nostaa LindeX-dollareita. Pankki tekee voittoa lainaamalla rahaa korkoineen Second Lifen asukkaille maanostoa varten [Bray and Konsynski, 2007].

Tämän luvun kohdissa käsittelen Second Lifen talouden mielestäni kahta tärkeintä tekijää: markkinointia ja virtuaalista omaisuutta. Pohdin lopuksi mitä kaikkea uuden yrityksen perustamiseen kuuluu Second Lifessa.

4.1. Markkinointi

Mainostamiseen kuuluvat 3D-esineet, kuten esimerkiksi uuden auton virtuaalisen mallin asettaminen näytteille. Mainostaulut, radio, musiikkiesitykset, elokuvat ja videot, minipelit ja aktiviteettien, kuten leirien, tarjoaminen ovat myös eräitä markkinoinnin muotoja Second Lifessa [Barnes, 2007].

Virtuaalimaailmassa mainostetaan sekä reaali maailman tuotteita että virtuaalisia tavaroita. Markkinointi voi koskea mitä tuotteita tahansa, esim. monet autoyhtiöt (Mercedes, Mazda), matkailufirmat (STA Travel) ja kuluttaja elektroniikkayhtiöt (Nokia, Intel) ovat maksaneet tuotesijoittelusta Second Lifessa [Barnes, 2007]. Samaten firmat, kuten American Apparel, ovat kokeilleet julkaista uusia tuotteitaan ensin virtuaalisesti Second Lifessa saadakseen osviittaa siitä, kuinka hyvin tuote tulee pärjäämään [Lui et al., 2007].

Kaikenlainen markkinointi ei toimi virtuaaliympäristöissä, sillä emme pysty maistamaan, tuntemaan tai haistamaan virtuaalisesti. On siis hyödytöntä tarjota avatarille, eli pelattavalle hahmolle, esimerkiksi makupalaa jostain tuotteesta. Lui ja muut [2007] kuitenkin huomauttavat, että vaikka tuotetta ei voi digitaalisesti nauttia, sitä voi silti markkinoida parantamalla tuotetietoutta esimerkiksi kokoontumisilla, joissa tuotteesta puhutaan, sillä tuotetietous on yhteydessä ostospäätökseen. Toisaalta Second Lifessa pystymme näkemään ja kuulemaan sekä puhumaan. Markkinoijat käyttävätkin näkö- ja kuuloaistia hyväkseen mainostamisessa. Avatarin voi viedä esimerkiksi taidegalleriaan, jossa on oikeiden töiden virtuaalikopioita. Lisäksi, vaikka Second Lifessa on paljon asukkaita, sisään kirjautuneita on yleensä vähän suhteessa virtuaalimaailman kokoon [Noam, 2007]. Osa näistä käyttäjistä on kehitysmaiden asukkaita tai alaikäisiä, jotka eivät ole markkinoijien näkökulmasta kiinnostavia.

Mainostaminen virtuaalimaailmoissa on monipuolisempaa kuin television tai radion kautta markkinointi, koska se tarjoaa myös interaktiivisen elementin. Esimerkiksi avatarin voi viedä ajelulle uudella Nissanilla, vaikka elämys ei ole täysin sama kuin reaalielämässä.

Clemons ja muut [2007] pohtivat myös markkinoinnin osuutta Web 2.0:n yhteisöpalveluihin, kuten Facebook ja Second Life. He toteavat, että mainostaminen on nykyään paljon vaikeampaa kuin ennen, koska se on niin yleistä ja peräti ylitsepursuavaa. Täten kuluttajien luottamus maksettuihin sanomiin on

heikentynyt todella paljon, joten tuotesijoittelun kannattavuus virtuaalimaailmoissa on kyseenalaista. Samoin voi pohtia, menettääkö virtuaalimaailma käyttäjien luottamuksen, jos se yrittää ohjata ostoskäyttäytymistä.

4.2. Virtuaalinen omaisuus

Virtuaalinen omaisuus on omaisuutta, joka on olemassa pelkästään virtuaalisesti. Barfield [2009] määrittelee virtuaalisen omaisuuden kolmella vaatimuksella. Ensimmäisessä vaatimuksessa todetaan, että jos joku omistaa synteettistä omaisuutta, kukaan muu ei sitä omista. Toinen on, että omaisuus ei häviä lopullisesti, kun käyttäjä sulkee tietokoneen, jolloin siihen uskaltaa sijoittaa rahaa. Kolmannessa muutkin virtuaalimaailman asukkaat voivat olla tekemisissä omaisuuden kanssa, mikä saattaa kasvattaa omaisuuden arvoa.

Second Lifessa käyttäjät ja yritykset tuottavat monenlaisia virtuaalisia esineitä, kuten vaatteita avatareille. Tuotteita ostetaan esimerkiksi oman hahmon kustomointia varten. Suurempia virtuaaliesineitä ovat kiinteistöt, tai jopa kokonaiset saaret. Yksi Second Lifen rikkaimmista asukkaista, Anshe Chung, on kerännyt usean sadan tuhannen Yhdysvaltojen dollarin omaisuuden Second Lifen kautta. Hän osti Second Lifelta suuria maa-alueita, joita hän koristeli lisäämällä mm. puita ja jokia, jonka jälkeen hän möi maa-alueet pienempinä kokonaisuuksina muille virtuaalimaailman asukkaille [Barfield, 2009].

Virtuaalisista tuotteista puhuttaessa on myös tärkeää ymmärtää, minkä takia niistä maksetaan. Käytännössä synteettisten asioiden ja esineiden viehätys on sama kuin esimerkiksi keräilykorteilla. Niillä ei ole reaali maailmassa käytännön hyötyjä, mutta ne voivat olla hyvinkin arvokkaita omistajalleen. Guo ja Barnes [2007] ovat selvittäneet tarkemmin syitä virtuaalimaailmoissa tapahtuvaan rahaliikenteeseen. Kärjistettynä, maksavat käyttäjät luottavat rahaliikenteen suojaukseen sekä haluavat useimmiten nostaa sosiaalista statustaan muiden käyttäjien silmissä. Käytännössä he uskovat virtuaalisen omaisuuden mm. lisäävän peliympäristön hauskuutta. Shin [2008] on päätenyt samoihin tuloksiin tutkiessaan ihmisten ostoskäyttäytymistä virtuaalimaailmassa. Luottamus Web 2.0:aan, virtuaaliseen valuuttaan ja tätä kautta virtuaalimaailmoin korreloi positiivisesti ostopäätöksen teon kanssa. Tuotteen helppokäyttöisyys, yhteisöllisyys, käytännöllisyys ja kaupankäynnin vaivattomuus vaikuttavat myös positiivisesti päätökseen. Riskit, kuten epäluottamus rahansiirtoon, ihmiskontaktin puute (myyjä ei ole fyysisesti läsnä) ja ostosten teon vaikeus taas heikentävät ostopäätöstä. Kaikki em. seikat kuuluvat *technology acceptance* -malliin (TAM), josta kerrotaan tarkemmin Shinin [2008] artikkelissa.

Second Life tekee voittoa myymällä maata ja perimällä kiinteistöveroaa. Pelin pelaaminen on muutoin ilmaista. Monet muut pelit, kuten World of Warcraft

raft, hankkivat tuottoa vaatimalla kuukausimaksun. Molemmat ovat osoittautuneet tuottoisiksi virtuaalimaailmoiksi.

4.3. Yrityksen perustaminen

Otetaan konkreettinen esimerkki, jossa aloitteleva artisti myy omatekoista musiikkiaan omassa levykaupassa. Myytävä tuote on tässä tapauksessa virtuaalimaailmaan sopiva, koska sitä kautta pystymme kuulemaan. Musiikkia voi myös myydä virtuaalisena, toisin kuin esimerkiksi leivoksia. Kustannukset jäävät alhaisiksi, koska kun kappale on tehty, sitä voidaan kopioida ja lähettää ilmaiseksi loputtomiin. Ainoastaan laitteisto äänittämiseen ja soittamiseen voi maksaa.

Varsinkin aloittelevan Second Lifen käyttäjän kannattaa aluksi opetella käyttämään virtuaalimaailmaa sekä verkostoitua muiden käyttäjien kanssa. Seuraavaksi Linden Labilta voi ostaa maata ja tehdä vaikkapa itse 3D-levykauppa. Artisti voi myös palkata asiaan erikoistuneen yrityksen, kuten ElectricSheepin, ”rakentamaan” levykaupan. Kun levykauppa on valmis, sinne täytyy houkutella muita virtuaalimaailman asukkaita. Tämä onnistuu markkinomalla musiikkia joko mainoskylteillä tai multimediaesityksillä paikoissa, joissa on paljon liikennettä. Väkeä voi houkutella myös palkkaamalla muita käyttäjiä hengamaan liikkeen luona. Markkinoinnissa artisti voi käyttää hyödykseen myös edellä esiteltyjä uusia liiketoimintamalleja. Hän voi pyytää yhteisöä auttamaan esimerkiksi oman brändin luomisessa ja haastaa muita tekemään uusia versioita kappaleistaan. Artisti voi pitää myös virtuaalisia konsertteja promovoidakseen musiikkiaan.

Musiikin myyminen kannattaa turvallisuussyistä hoitaa virtuaalimaailmoissa tapahtuvien rahasiirtoihin erikoistuneen yrityksen, kuten edellä esitellyn ItemBayn, kautta. Kun LindeX-dollareita on kerääntynyt tarpeeksi, artisti voi muuntaa sen oikeaksi rahaksi Linden Labin tarjoaman palvelun kautta, joka löytyy Second Lifen virallisilta sivuilta (<http://secondlife.com/>).

Yrityksen perustaminen virtuaalimaailmaan ei siis eroa paljon normaalista, ja vaikka aloittaminen voi olla helpompaa ja edullisempaa, sekään ei ole riskitöntä. Käsittelen seuraavassa luvussa ongelmia, joita kuvitteellinen artistimme voi kohdata liiketoimintaa hoitaessaan.

5. Liiketoiminnan ongelmat, riskit ja hyödyt virtuaalimaailmassa

Virtuaalimaailmassa liikkuu paljon rahaa, jota ollaan valmiita hankkimaan myös epärehellisesti. Reaalimaailmassa on aina ollut ihmisiä, jotka ovat hankkineet osansa rahaliikenteestä huijaamalla ja varastamalla. Virtuaalimaailmassa kukaan ei voi varastaa konkreettisesti käsilaukkua, mutta esimerkiksi hakke-

roimalla voi päästä käsiksi toisten rahoihin. Toisaalta edellä käsitellyillä uusilla liiketoimintamalleilla on paljonkin hyötyjä niin kuluttajan kuin yrityksen näkökulmasta.

5.1. Ongelmat ja riskit kaupankäynnissä

Kaupankäynti virtuaaliomaisuudella on melko uusi ilmiö yhä, joten siihen ei ole vielä olemassa selkeitä lakeja. Second Lifea lukuun ottamatta hyvin monessa tapauksessa virtuaalimaailman omistaja omistaa lopulta kaikki virtuaaliset hyödykkeet. Käytännössä tällaisissa virtuaalimaailmoissa käyttäjät eivät voisi myydä esineitä ilman virtuaalimaailman omistajan lupaa, mutta sitäkin tapahtuu. Yhtenä esimerkkinä tästä on World of Warcraft, joka on poistanut yli miljoona dollaria käyttäjiltä, jotka ovat tehneet kauppaa peliin kuuluvilla tavaroilla [Thorpe et al., 2008]. Huijaamista virtuaalimaailmoissa tapahtuu samalla lailla kuin oikeassakin. Myyjä voi kieltäytyä suorittamasta palvelusta, josta ostaja on maksanut tai tavaroista voidaan pyytää ylihintaa toivoen, ettei toinen huomaa. Koska harvat virtuaalimaailmat tukevat kirjallisia sopimuksia, pelaajan ainut toivo huijatuksi tulemisen jälkeen on toivoa, että virtuaalimaailman omistajat ratkaisevat ongelman [Thorpe et al., 2008]. Virtuaalimaailmoissa liikkuva raha ja sen muuttaminen oikeaksi rahaksi tarjoaa järjestäytyneellekin rikollisuudelle uusia mahdollisuuksia esimerkiksi rahanpesuun. Toistaiseksi siitä ei kuitenkaan ole mitään merkkejä.

McInnes [2006] huomauttaa vielä, että monet nuoret käyttäjät pitävät virtuaalimaailmojen liiketoimintaa vapaa-ajan toimintana, eli he eivät suhtaudu siihen vakavasti. Tällöin varastamista ja huijaamista ei pidetä samalla lailla vakavana asiana kuin oikeassa elämässä. Ilmiö on sama kuin elokuva- ja musiikki-teollisuutta vaivanneessa piratismassa. Kynnys epärehelliseen toimintaan virtuaalimaailmoissa ei ole siis yhtä korkea kuin reaali maailmassa.

Rahaliikenne kiinnostaa yritysten lisäksi myös lainsäätäjiä. Yhdysvaltojen hallituksessa on pohdittu virtuaalisen omaisuuden verottamista [Barfield, 2009], johon Bray ja Konsynski [2007] suhtautuvat kriittisesti. He kysyvät, missä virtuaalimaailma sijaitsee, sekä verotetaanko asukkaita sen mukaan, missä palvelin sijaitsee, vai sen mukaan, missä avatarien oikean elämän vastineet ovat. Lisäksi Barfield [2009] pohtii, onko käyttäjää verotettava, jos hän vain kerää virtuaalisia esineitä ja valuuttaa muuttamatta niitä oikeaksi rahaksi.

Lainsäädännöllisiä haasteita ja ongelmia lisäävät tekijänoikeuslait ja kysymykset yksityisyydestä. Mahdolliset säännökset muuttaisivat toteutuessaan virtuaalimaailman luonnetta. Käyttäjät ovatkin huolissaan siitä, kuinka yritykset ja lainsäädännöt tekisivät virtuaalimaailmasta liian kaupallisen ja veisivät alkuperäisen virtuaaliympäristön viehättyksen mukanaan. Asiaa vastaan on jopa ryhmittäytytty, yhtenä esimerkkinä SLLA (Second Life Liberation Army),

joka on vandalisoinut virtuaalimaailmassa olevia yrityksiä [Bray and Konsynski, 2007].

Ohjelmoijien kannalta haasteita lisää ohjelmien turvaaminen hakkerointiyrityksiltä, sillä käyttäjät voivat käyttää pelin sisällä olevia ohjelmointivirheitä hyväkseen tehdäkseen esimerkiksi voimakkaita aseita tai nostaakseen arvokkaan esineen useaan kertaan. Myymällä näitä vilpillisesti hankittuja tavaroita he tienavat rahaa.

5.2. Hyödyt kaupankäynnissä

Hyvänä puolena kaupankäynnissä on se, että monet reaali maailman ongelmat eivät voi tapahtua virtuaalimaailmoissa. Thorpe ja muut [2008] mainitsevat muun muassa väärennöksen mahdottomuuden. Pelaaja näkee heti, onko esim. vaate sen näköinen kuin se oikeasti on. Ainoastaan hyvin vaivalloisella erikoiskoodaamisella voidaan tuottaa tällaisia väärennöksiä. Väkivallan käyttö pelissä varastamiseen on myös turhaa. Poikkeuksia kuitenkin löytyy; mm. virtuaalimaailma EVE Onlinessa pelin "villi länsi" -luonteeseen kuuluu se, että toisen hahmon tappamisella saa haltuunsa tämän omaisuuden. Luonnonsuojelun näkökulmasta virtuaaliset hyödykkeet ovat vaarattomia, koska ne eivät kuormita luontoa. Esimerkiksi virtuaalisen auton valmistukseen ei tarvita raaka-aineita.

Säännökset, lait, joita vastaan SLLA taistelee, ovat ristiriitaisia, koska ainakin osa Second Lifen asukkaista haluaa järjestystä virtuaalimaailmaan. Anshe Chung, joka tarjoaa mm. koristeltuja maa-alueita, on huomannut järjestyksen kaipuun, koska virtuaalimaailman asukkaat toivoivat häneltä hieman säännöksiä virtuaaliseen asuinympäristöönsä. Tällöin lait tekisivät virtuaalimaailmasta stabiilimman [Bray and Konsynski, 2007] ja voisivat olla hyödyllisiä reaali maailmallekin tutkimuskohteina. Nykyisellään virtuaalimaailmojen avulla on tutkittu enemmän sosiaalista kanssakäymistä.

Virtuaalisten esineiden tuottaminen on periaatteessa ilmaista, esimerkiksi Second Lifessa käyttäjät voivat luoda asuja käyttämällä pelissä olevaa työkalua. Tällöin voitto on 100-prosenttista ja kuluttajallekaan kustannukset eivät nouse korkealle ainakaan raaka-aineiden takia. Lisäksi, kun tuote on tehty, sen kopiointi on nopeaa ja maksutonta, niin pitkään kuin koodi säilyy. Koska virtuaalisten asioiden ja esineiden luominen on käytännössä maksutonta useimmissa tapauksissa, periaatteessa kuka tahansa voi tehdä niitä. Mahdollisuus liiketoimintaan on siis jokaisella riippumatta iästä tai pääomasta, ja töitä voi tehdä kotoa – ei tarvita tehtaita ja liiketiloja vuokrineen.

6. Liiketoiminnan tulevaisuus virtuaalimaailmoissa

Virtuaalimaailmojen liiketoiminta on vielä niin tuoretta, ettei ole varmaa, onko kyseessä ohimenevä ilmiö vai pysyvä uudistus. Tärkeää olisi kuitenkin tulevaisuudessa ratkaista tekniset ongelmat ja lainsäädännölliset kysymykset. Muutoin virtuaalimaailmoissa tapahtuvan liiketoiminnan luotettavuus kärsii, tai pahimmassa tapauksessa häviää kokonaan, jolloin käyttäjät saattavat lopettaa ostamisen ja myymisen ja siirtyä muihin ympäristöihin. Bray ja Konsynski [2007] pohtivat lisäksi, tulemmeko näkemään virtuaalisia hallituksia ja kuinka reaali maailman hallitukset reagoisivat tällaisiin hallituksiin, mikä hankaloittaisi asioita entisestään. Askeleita tähän suuntaan on jo otettu, sillä Ruotsi on tehnyt oman lähetystön Second Lifeen [Barnes, 2007].

Teknologian kehittyessä voisi olla mahdollista myös maistaa, tuntea ja haisua tuotteita virtuaalisesti [Lui et al., 2007]. Markkinoille kehittyisi tällöin aivan uusia tyylejä mainostaa virtuaalisia tuotteita. Itse asiassa käsite virtuaalisesta tuotteesta muuttuisi kokonaan. Onko tietokoneen kautta tuoksuva hajuvesi kovin erilainen oikeasta hajuvesipullosta? Tulevaisuudenvisioissa markkinointia pystyttäisiin tarkemmin kohdistamaan oikeisiin henkilöihin, mikä voisi tehdä siitä paljon tuotteliaampaa. Toisaalta jo nykyisellään suuri osa käyttäjistä karsastaa mainontaa. Esimerkiksi ystävien mielipiteillä tuotteista on suurempi arvo kuin maksetulla tuotesijoittelulla [Clemons et al., 2007]. Kuitenkin, vaikka mainonnan teho on laantunut, ne ovat silti ottaneet paikkansa reaali maailman katukuvassa. Tällöin niiden ilmaantuminen Second Lifen kaltaiseen virtuaali maailmaan voi tehdä siitä todemmantuntuisemman. On taas eri asia, onko se toivottua vai ei uudessa ympäristössä. Tulevaisuudessa markkinoinnin tulee olla joka tapauksessa paljon harkitumpaa, ja uusia tyylejä kokeilevaa, niin ettei se karkota käyttäjiä.

Toistaiseksi virtuaalimaailmoissa tapahtuva liiketoiminta on kasvussa ja talouden uudet toimintamallit näyttävät toimivan erittäin hyvin.

7. Yhteenveto

Web 2.0 on yhteisöllinen media, johon jokainen voi tuottaa sisältöä. Osasta sisältöä ollaan valmiita maksamaan oikealla rahalla. Virtuaalimaailmoissa, kuten Second Lifessa, liiketoiminta on hyvin vilkasta ja kauppaa käydään oikeiden esineiden lisäksi virtuaalisista asioista ja tavaroista. Käyttäjät ostavat esimerkiksi virtuaalisia kiinteistöjä nostaakseen sosiaalista statustaan virtuaalimaailmassa. Rahankäyttö vaatii kuitenkin luottamusta ja totuttelua.

Uuteen ympäristöön on kehittynyt uudenlaisia liiketoimintamalleja, joista osa luokitellaan osallistumistalouteen kuuluviksi. Uudet liiketoimintamallit ovat olleet suurilta osin menestystarinoita, mutta vanhat, kuten mainostami-

nen, nähdään ristiriitaisesti varsinkin uudessa ympäristössä. On mahdollista, että osallistumistalous voi syrjäyttää perinteisen liiketalouden tulevaisuudessa niin virtuaali- kuin reaali-maailmassakin.

Liiketoimintaan virtuaalimaailmoissa kuuluu tällä hetkellä paljon riskejä, koska sille ei ole vielä säädetty tarkkoja lakeja. Muun muassa eräiden Second Lifen käyttäjien varakkuus, tekijänoikeus- ja yksityisyyskysymykset sekä hakkerointi vaativat huomiota myös hallinnollisella tasolla. Esimerkiksi virtuaalisen omaisuuden verottamista on mietitty. Toisaalta liiketoiminnassa on monia hyötyjä esimerkiksi luonnonsuojelun kannalta. Uusi media tarjoaa paljon uusia mahdollisuuksia yrityksille ja käyttäjille.

Clemons ja muut [2007] toteavat, että virtuaalimaailmojen asukkaat käyttävät Internetiä nähdäkseen ja tehdäkseen mitä vain haluavat ja milloin haluavat. Osallistuminen virtuaalimaailman liiketoimintaan on myös täysin vapaaehtoista, koska avatarit eivät tarvitse esimerkiksi ruokaa selvitäkseen hengissä, eivätkä ne koskaan sairastu [Thorpe et al., 2008]. Koska liiketalous on täysin vapaaehtoista, on sen jatkuvuus täysin käyttäjien kiinnostuksesta kiinni. On kiintoisaa seurata, kuinka virtuaalimaailmojen liiketalous kehittyy ja miltä se näyttää tulevaisuudessa.

Viiteluettelo

- [Barfield, 2009] Woodrow Barfield, On money, taxes and property in virtual reality. *Virtual Reality* 13, 1 (March 2009), 37–39.
- [Barnes, 2007] Stuart Barnes, Virtual worlds as a medium for advertising. *ACM SIGMIS Database* 38, 4 (Nov. 2007), 45–55.
- [Bray and Konsynski, 2007] David A. Bray and Benn R. Konsynski, Virtual worlds: multi-disciplinary research opportunities. *ACM SIGMIS Database* 38, 4 (Nov. 2007), 17–25.
- [Cagnina and Poian, 2008] M. R. Cagnina and M. Poian, Second Life: A Turning Point for Web 2.0 and E-Business? In: *Interdisciplinary Aspects of Information System Studies*. Physica-Verlag HD, 2008, 377–383.
- [Clemons et al., 2007] Erik K. Clemons, Steve Barnett and Arjun Appadurai, The future of advertising and the value of social network websites: some preliminary examinations. In: *Proceedings of the Ninth International Conference on Electronic Commerce, ACM International Conference Proceeding Series* 258 (2007), ACM, 267–276.
- [Greenfield, 2008] Dave Greenfield, Doing business in the virtual world. *eWeek* 25, 8 (Mar. 2008), 37–39, 42–43.
- [Guo and Barnes, 2007] Yue Guo and Stuart Barnes, Why people buy virtual

items in virtual worlds with real money. *ACM SIGMIS Database* **38**, 4 (Nov. 2007), 69–76.

[Hintikka, 2008] Kari A. Hintikka, Johdatus osallistumistalouteen – internetin uusia taloudellisia toimintaympäristöjä. *Tieke ry:n julkaisusarja* **32**, (Maaliskuu 2008).

[Lui et al., 2007] Tsz-Wai Lui, Gabriele Piccoli and Blake Ives, Marketing strategies in virtual worlds. *ACM SIGMIS Database* **38**, 4 (Nov. 2007), 77–80.

[MacInnes, 2006] Ian MacInnes, Property rights, legal issues, and business models in virtual world communities. *Electronic Commerce Research* **6**, 1 (Jan. 2006), 39–56.

[Noam, 2007] Eli M. Noam, The dismal economics of virtual worlds. *ACM SIGMIS Database* **38**, 4 (Nov. 2007), 106–109.

[Shin, 2008] Dong Hee Shin, Understanding purchasing behaviors in a virtual economy: Consumer behavior involving virtual currency in Web 2.0 communities. *Interacting with Computers* **20**, 4–5 (April 2008), 433–446.

[Thorpe et al., 2008] Christopher Thorpe, Jessica Hammer, Jean Camp, Jon Callas and Mike Bond, Virtual economies: threats and risks. In: *Financial Cryptography and Data Security, Lecture Notes in Computer Science* **4886**, Springer Berlin/ Heidelberg, 2008, 239–244.

Ohjaaja: Sari Walldén

Kohti mukautuvia käyttöliittymiä

Pasi Paunu

Tiivistelmä.

Tutkielman tarkoituksena on lähestyä mukautuvien käyttöliittymien ajatusta käsitteellisen mallintamisen näkökulmasta. Esiin otetaan mallintamisen teoriaa sekä siihen perustuvia ohjelmistoja. Näihin asioihin viitaten käydään lopuksi läpi mukautuvien käyttöliittymien toteuttamisajatusta, jonka kivijalaksi käsitteellinen mallintaminen todetaan.

Avainsanat ja -sanonnat: Käsitteellinen mallintaminen, mukautuvat käyttöliittymät, käsitteellistys, Concept D, Enhanced Entity-Relationship Model.

CR-luokat: C.0, D.2.2, H.5.2

1. Johdanto

Pureudun tässä tutkielmassa ajatukseen mukautuvista käyttöliittymistä ja niiden toteuttamisesta käsitteellisen mallintamisen näkökulmasta. Erilaisia käyttäjän toimiin mukautuvia sekä muovattavia käyttöliittymiä on tutkittu jo pitkään ja erilaisia toteutuksia nähty esim. toimintojen käytön mukaan skaalautuvista valikoista [Miah et al., 1997] aina agenttiperustaisiin käyttöliittymien rakentajiin [Taghezout et al., 2008]. Lähestyn tässä aihetta kuitenkin ohjelmien suunnittelun ja hahmottelun puolelta, joissa käsitteellinen mallinnus luo tukevan pohjan tulevaisuuden mukautuville käyttöliittymille. Jotta mukautettavuus olisi ylipäättään mahdollista, on ohjelmien suunnittelussa pystyttävä ottamaan huomioon käsitteiden todellinen merkitys ja hyödynnettävä niitä käyttöliittymien mukautettavuuden peruspilareina.

2. Käsitteellinen mallintaminen

Jotta voimme perehtyä aiheeseen myöhemmin mahdollisimman mukavasti, on ensin käytävä käsiksi aihepiirin käsitteistöön ja määriteltävä, mitä itse käsitteellinen mallintaminen oikeastaan pitää sisällään. Yhtenä hyvänä yleismäärittelyksenä käsitteellinen mallintaminen voidaan ymmärtää eräänlaisena metodina kerätä tietoutta jostakin kohdealueesta (engl. Universe of Discourse, UoD), jossa tietous pitää sisällään erilaisten kohteiden tai objektien ominaisuuksia sekä kuvauksia kohteiden välisistä suhteista [Kangassalo, 1990a, 20]. Käsitteellinen mallintaminen on eräänlainen monitieteellinen hybridi, jossa yhdistyy mm. filosofia, psykologia, kognitiotiede, logiikka, lingvistiikka, tieto-oppi, tiedonhallinta, tekoäly- sekä tietämuskantojen tutkimus. Tästä johtuen käsitteellisellä

mallintamisella on eri tieteenhaaroissa hieman omanlaisensa sovellusperiaatteet sekä teoriakäytäntö jota seurataan. [Kangassalo, 1990a, 21.] Keskityn kuitenkin tässä tutkielmassa lähinnä tietojenkäsittelyn vaikutuspiirissä olevaan käsitteistöön sekä teoriaan, jota sovelletaan ohjelmistojen suunnittelussa ja toteutuksessa käsitteellisen mallintamisen kanssa.

2.1. Keskeiset ajatukset sekä termistö

Tarkemman tarkastelun alla voimme erottaa käsitteellisen mallintamisen ajatukset esittää 1) tarkasteltavan kohdealueen käsitteet sisältöineen, 2) käsitteiden väliset suhteet ja käsitteiden muodostama rakenne sekä 3) edellisten avulla tehty kohdealueen tapauskohtainen tai yleinen kuvaus [Kangassalo 2007, 3 - 7]. Kun tätä jaottelua lähdetään purkamaan osiin, tulee ensimmäisenä vastaan kohdealueen määrittely.

2.1.1. Kohdealue

Kohdealueella tarkoitetaan yleisesti jotakin kiinnostuksen kohteena olevaa fyysistä, sosiaalista tai muuta käsitteellistä avaruuden rajattua osaa. Kohdealueen kuvauksen voidaan sanoa muodostuvan valittujen käsitteiden sekä tietämysprimitiivien ja niiden välisten suhteiden kautta, jotka määrittelevät samalla valitun kohdealueen tai sitä koskevan yleisen teorian. Kuvaus esitetään tällöin käsittekaaviona (engl. conceptual schema), joka kuvaa kaikki käsitteellisen rakenteen yleiset sekä tapauskohtaiset piirteet. Sama kuvaus voidaan myös esittää ns. käsitteellisenä mallina (engl. conceptual model), jossa määrittyy vain tietyn tilanteen tai rajatun tilannejoukon tapauskohtainen kohdealueen kuvaus. [Kangassalo, 2007, 3.] Koska käsitteellisessä mallissa ei oteta välttämättä aivan kaikkia mahdollisia tapauksia huomioon, on se usein käsittekaaviota rajoittuneempi. Seuraavana tärkeänä palasena mallintamisessa on itse käsitteen termi ja termistö.

2.1.2. Käsitteen määritelmä

Käsitteen määritelmä on sen kielellinen rakennusohje, jonka avulla käsitteeseen sisältyvä tietämys voidaan koota tai muodostaa tietämysalkioista tai muista käsitteistä. Määritelmä tyypit voidaan jakaa yleisesti kolmeen osaan:

- 1) Käsitteen intensionaalinen määrittely, jossa käsitteeseen intensionaalisesti sisältyvät toiset käsitteet ja niiden väliset keskinäiset suhteet sekä muut tietämysalkiot (tunnusmerkit) kuvataan jollakin soveltuvalla määrittelytekniikalla.
- 2) Käsitteen määrittely ulkoisten suhteiden avulla, jolloin määriteltävän käsitteen suhteet tarkennetaan niihin saman käsitejärjestelmän käsitteisiin, jotka eivät sisälly määriteltävään käsitteeseen.

- 3) Representaationaalinen eli esityksen määrittely, jossa ilmaistaan määrittely tarkentamalla miten määriteltävää käsitettä edustava arvo tulisi johtaa määritteleviä käsitteitä edustavista arvoista.

[Kangassalo, 1990b, 34, 35.]

2.1.3. Käsite

Varsinaisilla käsitteillä (engl. concept) on mallinnuksessa merkittävin osa. Ne ovat ihmisen ajattelun kognitiivisia työkaluja, joita ilman ei esiintyisi kehittyntä ajattelua eikä varmasti nykyisen kaltaista tietojenkäsittelyä, kuten Kangassalo [2007, 3] esittää. Käsitteet sisältävät suurimman osan siitä informaatiosta, jota me käsittelemme maailmasta sekä sen osasta kohdemaailman rajausten avulla. Kaikki kehittynyt ajattelu ja suunnittelu, mukaan lukien tietojärjestelmien suunnittelu, perustuvat juurikin sille tärkeälle käsitteiden tunnistamisen, kehittämisen ja muodostamisen tarpeelle, jota ilman emme pystyisi luomaan oikeastaan mitään kovin merkittävää Kangassalo [2007, 3].

Yleisesti voimme nähdä, että käsitteellä on rakenteinen sisältö, joka määrittää osan sen roolista tai osuudesta käsitteiden muodostamassa kokonaisuudessa. Sen sisältö voi olla erittäin monimutkainen sisältäen sen käyttöön ja käyttöyhteyksiin liittyvää tietoa. Käsitteen rakenneseosina toimivat ns. tietämysalkiot eli tietämysprimitiivit sekä toiset käsitteet. Käsite voi tällöin koostua sekä pelkästään tietämysprimitiiveistä tai sisältää myös muita käsitteitä. Tietämysprimitiivit eivät voi koskaan esiintyä käsitteissä yksistään vaan niitä tulee aina olla vähintään kaksi, joista toisen on oltava joko käsitteen nimi tai eksistenssi. Seuraavassa lista mahdollisista tietämysprimitiiveistä Kangassalon [1990b, 18 - 20] mukaan:

- *Käsitteen nimi*, jonka tulee olla uniikki ja vain ko. käsitteen tunnistava. Nimi yhdistää tällöin myös käsitteiden sekä kielen tason.
- *Käsitteen eksistenssi*, joka sisältää käsitteelle asetetun totuusarvon. Arvon ollessa tosi käsite on määritelty erillisellä, eksplisiittisesti tai implisiittisesti ilmaistulla määritelmällä, eikä arvo ole syntynyt jonkin satunnaisen tietämystä käsittelevän operaation tuloksena.
- *Käsitteiden väliset intensionaaliset sisältymissuhteet*.
- *Käsitteiden esiintymien väliset ekstensionaaliset suhteet*, joita ovat mm. lukumääräsuhteet ja arvojoukot.
- *Yksilöintiominaisuus*, joka tarkoittaa käsitteen esiintymään liittyvää ominaisuutta, jolla käsite voidaan yksikäsitteisesti tunnistaa muista käsitteistä.
- *Ehto*, joka on totuusarvoinen lause riippuen esim. käyttötilanteen sille antamasta arvosta.

- *Rajoitus* on totuusarvoinen lause, jonka edellytetään olevan tietyssä käyttötilanteessa tosi.
- *Ehdollinen rajoitus* on ehtojen ja rajoitusten muodostama konstruktio, jonka rajoitusosaan liittyvän totuusarvon edellytetään olevan tosi silloin, kun ehto-osaan liittyvä totuusarvo tietyssä käyttötilanteessa on tosi.
- Käsitteiden esiintymisen *esiintymisehto* on totuusarvoinen lause, joka ilmaisee ehdon, jonka ollessa tosi käsitettä vastaava esiintymä voi esiintyä. On huomattava, että esiintymisehdon kuuluminen käsitteen itsensä sisältöön (intensioon) tai käsitettä käyttäen laadittuun kohdealueen kuvaukseen, on hieman tulkinnanvarainen.
- *Aikalause* on joko absoluuttinen viittaus kohdealueen kalenterijärjestelmään tai viittaus tapahtumien järjestystä koskevaan suhdejärjestelmään.
- Käsitettä vastaava *aistitieto*, joka usein muodostaa käsitteen määrittelyn perustan. Aistitiedossa voidaan erottaa kaksi tapausta: käsitteen määrittelyyn tai tunnistuksen perustana oleva *mallihavainto* ja tapahtumiin liittyvät myöhemmät *havainnot*.
- Käsitteen esiintymää edustava *arvo*, joka on yleensä abstraktiin eli määrittelyyn kuuluva käsite. Tiettyä käsitteen esiintymää edustava arvo on ko. käsitteen kohdealueelta rajaamaan ilmiöön kohdistuvan operaation (havainnon, luokituksen, mittauksen tai laskutoimituksen) tulos, jonka ko. operaation suorittava havaintofunktio valitsee mahdollisten arvojen joukosta.
- Käsitteen *tunnistefunktio*, joka aistitietoa (mallihavaintoa ja tietyssä tilanteessa saattaa havaintoa) tutkimalla ratkaisee, sisältyykö käsiteltävään olevaan havaintoon käsitettä vastaavaa ilmiötä, ts. käsitteen esiintymää.

Perinteisessä käsitteellisessä mallintamisessa yllä olevista tietämysprimitiiveistä käytössä on yleensä vain *käsitteen nimi, ekstensionaaliset suhteet, yksilöintiominaisuus, ehto, rajoitus sekä arvo*, mutta poikkeuksina myös *ehdollinen rajoitus, esiintymisehto ja aikalause* [Kangassalo 1990b, 20]. Jos käsite koostuu kahdesta tai useammasta käsitteestä, korostuu tietämysprimitiiveistä *intensionaalinen sisältymissuhde*, johon perehdymme seuraavaksi.

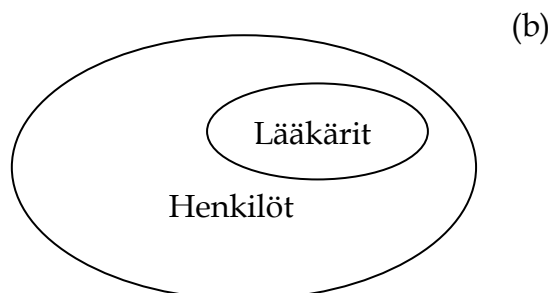
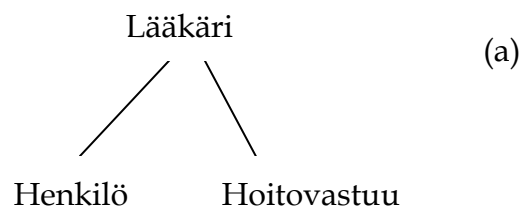
2.1.4. Intensionaalinen sisältymissuhde, intensio ja ekstensio

Intensionaalinen sisältyminen on refleksiivinen ja transitiivinen relaatio kuten Kangassalo [1990b, 20] osoittaa Kaupin [1967, 36] käsitejärjestelmien teoriaan viitaten. Voimme nähdä useita erilaisia suhteita käsitteiden välillä, joista tässä

esittelen vain pääkohdat. Tarkempien käsitteitä koskevien analyysien sekä määritteiden kohdalla tulee alla olevan listan ajatuksia tarkastella Kaupin [1967] teorian valossa:

1. Käsitteet **a** ja **b** voivat olla toisistaan täysin erillisiä, ts. niissä ei ole yhtään yhteistä tunnusmerkkiä.
2. Käsitteillä **a** ja **b** voi olla yhteisiä tunnusmerkkejä, ts. ne ovat tunnusmerkkiyhteisiä.
3. Käsite **b** voi sisältyä käsitteeseen **a**, ts. kaikki käsitteen **b** tunnusmerkit ovat samalla myös käsitteen **a** tunnusmerkkejä. Tässä tapauksessa sanotaan, että käsite **a** **sisältää intensionaalisesti** käsitteen **b**.

Voidaan myös havaita, että käsitteiden ja niiden kuvaaman kohdealueen olioiden välillä on seuraavassa esitetty käänteinen suhde. Kangassalo [1990b, 20] tähdentää, että "jos käsite **b** sisältyy intensionaalisesti käsitteeseen **a**, niin jokaisella käsitettä **a** vastaavalla oliolla on ominaisuus **b**, ts. **b**, on **a**:n tuntomerkki. Tällöin olion **a** esiintyessä esiintyy myös ominaisuus **b**, mutta ominaisuuden **b**:n esiintyessä ei välttämättä esiinny oliota **a**". Transitiivisuuden esiintymisestä intensionaalisessa sisältymisessä sekä käsitteiden ja niiden kuvaaman kohdealueen välisestä käänteisestä suhteesta seuraa se, että käsitteet muodostuvat hierarkioiksi. Tällaisessa hierarkiassa kaikkien yleisimmät käsitteet, jotka sisältyvät kaikkiin muihin käsitteisiin, esiintyvät hierarkian alimmalla tasolla ja kaikkein tarkimmat eli ne erikoistuneet käsitteet, jotka sisältävät useita eri käsitteitä, sijaitsevat hierarkian huipulla. [Kangassalo, 1990b, 20 - 22.] Kuvasta 1 voimme nähdä yksinkertaisen käsitteen ja käsittehierarkian esimerkin, jossa ylimmällä tasolla on **Lääkäri** ja johon intensionaalisesti sisältyy käsitteet **Henkilö** sekä **Hoitovastuu**.



Kuva 1. Käsitteiden tasot a) Intensio ja b) Ekstensio

Kohdan a intensionaalisessa sisällymisessä voimme myös todeta, että käsite **Lääkäri** tuntemiseksi on tunnettava myös käsitteet **Henkilö** sekä **Hoitovastuu**, mutta esimerkiksi käsitteen **Henkilö** tuntemiseksi meidän ei tarvitse tuntea käsitettä **Hoitovastuu**. Käsitteitä voidaan jakaa tästä yhä pidemmälle, kuten käsitettä **Henkilö**, johon olennaisesti sisältyy käsite **Elollinen olento**. Voimme siis havaita käsiterakenteen muodostavan omanlaisensa käsitehierarkian, jossa erikoistunein käsite on ylimpänä sisältäen muita käsitteitä ja alimmalla tasolla yleisimmät käsitteet, jotka sisältävät vähiten muita käsitteitä. Kohdassa b tarkastelemme tilannetta joukko-opin kautta, jolloin havaitsemme, että joukko "henkilöt" sisältävät joukon "lääkärit" eli voidaan sanoa olevan henkilöitä, jotka ovat lääkäreitä, mutta kaikki henkilöt eivät ole lääkäreitä. Mutta kohde, jolla on ominaisuus "lääkäri", voidaan sanoa olevan myös henkilö. [Kangassalo, 1990b, 21 - 22.]

Abstrahoitu tietämys, joka sisältyy johonkin käsitteeseen, voidaan ymmärtää muodostavan käsitteen intensio. Intensio on siis käsitteen ominaisuuksien eli tunnusmerkkien muodostama rakenne, joka määrittelee käsitteeseen sovellettavissa olevat kohdealueen ilmiöt. Näiden tunnusmerkkien ollessa täten käsitteeseen intensionaalisesti sisältyvät tietämysprimitiivit sekä toiset käsitteet ja niiden muodostamat rakenteet [Kangassalo, 1990b, 25 - 28]. Korpela [2009] antaa intensiolle pienehkössä web-sanakirjassaan myös selitteen "käsitteen olennaiset piirteet; sanan tai muun ilmaisuuden kaikkia tarkoitteita kuvaavat piirteet".

Ekstensiolla kuvataan sitä tietämystä, joka kuuluu käsitteen ympäristöön niistä objekteista, joihin käsitteellä on relaatioita [Kangassalo, 1990b, 28 - 32]. Korpela [2009] selventää ekstension termiä vielä kuvauksella "laajentuminen; niiden olioiden joukko, johon jokin nimi tai muu ilmaisuus viittaa".

2.1.5. Käsitekaavio

Käsitekaavion voidaan sanoa olevan kuvaus kohdealuetta kuvaavista käsitteisistä ja niiden välisistä suhteista. Se on siis kohdealueensa käsitteellinen kuvaus ja alusta "- - johon esim. tietosysteemien ja tietokantojen tiedot niiden merkitystä ja tietojen käsittelyssä noudatettavia eheyssääntöjä määriteltäessä kytketään". [Kangassalo, 1990b, 11]. Käsitekaavio luo perustan tietoihin liittyvien sääntöjen oikeellisuuden tarkistamiselle ja varmistaa, että erilaiset eheyssäännöt ovat ylipäätään mahdollisia rakentaa, kun käsitekaavio tunnetaan. Sen rakenneosina toimivat abstrahoidut konstruktiot havaintoihin sisältyneestä informaatiosta sekä käsitteellisestä aineksesta koostuvat tietämysprimitiivit. Myös toiset käsit-

teet tai niiden osista muodostuvat rakenteet kaikkien komponenttien välisten suhteiden ohella ovat käsitekaavion sisältöosia. [Kangassalo, 2007, 7.]

Käsitekaavioista voidaan myös esittää kaksi erilaista jäsenystä: sisäinen sekä ulkoinen käsitekaavio. Edellisestä puhuttaessa, tarkoitetaan ihmisen mielessä olevaa käsitteellistä kaaviota, joka on monin verroin rikkaampi esitys kuin ulkoinen. Voimmekin näin havaita ulkoista käsitekaaviota tehtäessä, että prosessissa tapahtuu ns. informaation häviämistä. Jälkimmäisestä puhuttaessa ei muuta eroa sisäiseen käsitekaavioon juuri ole, mutta on olemassa kuitenkin mahdollisuus, että jokin sisäisen käsitekaavion käsitekuvauksista on toistaiseksi tai pysyvästi mahdotonta esittää ulkoisen käsitekaavion avulla. Myös formaaleja kuvausmenetelmiä käyttämällä voidaan ulkoiseen käsitekaavioon saada sellaisia käsitekuvauksia, joita ei kuvauksen tekijän sisäisessä käsitekaaviossa entuudestaan vielä ollut. Ulkoisen käsitekaavion luontiin liittyy myös rajoite, jonka mukaan siihen ei tulisi koskaan luoda sellaisia ominaisuuksia, joita ei löydy myös sisäisestä käsitekaaviosta. [Kangassalo, 2007, 6.]

Kaavion konkretisointi esim. graafiseen tai sanalliseen muotoon mahdollistaa sen käytön suunnittelutyön eri vaiheissa kaikkien osapuolten välisen kommunikaation pohjana tai tietojärjestelmän mallina. Tällöin kaavion kuvaus tehdään jollakin käsitteidenkuvauskielillä kuten esim. ERMIA, EER tai Concept D. Näihin kieliin sekä muihin ohjelmiin palaan myöhemmin käyttöliittymien ja käsitteellisen mallintamisen yhdistävässä luvussa.

2.1.6. Käsitteellistämisen- ja 100 % -periaate

Käsitteellisen mallintamisen perusteisiin liittyy hyvin oleellisesti niin käsitteellistämisperiaate kuin 100 % -periaatekin. Kangassalo [1990b, 11] viittaa näihin periaatteisiin van Griethuysenin [1982] tutkimuksen kautta esittämällä, että 100 % periaatteessa kohdesysteemin kaikki asiaankuuluvat pysyvät, yleiset tai muuttuvat seikat, jotka määräävät yksittäistapausten käyttäytymistä ja olemassaoloa, eli säännöt, lait ja säännönmukaisuudet tulee kuvata käsitekaaviossa. Kangassalo myös jatkaa, että käsitteellistämisperiaatteen takana on ajatus kuvata käsitekaaviossa ainoastaan kohdesysteemin käsitteellisesti asiaankuuluvia, sekä muuttuvia että pysyviä piirteitä, jolloin jätetään syrjään kaikki se tietojen esittämiseen, fyysiseen organisointiin sekä esille saantiin liittyvät seikat kuten tietorakenteet ja sanomien muodot.

100 % -periaatteen ajatus on siis varmistaa, että kaikki tarvittava tietojen merkityksen säilymiseen tähtäävät säännöt, kuten eheys- ja johdonmukaisuussäännöt ovat yhdessä paikassa, eivätkä joudu ripotelluiksi kymmeniin tai jopa satoihin eri ohjelmistojen osiin tietojärjestelmässä. Käsitteellistämisperiaatteen tavoitteena on taas varmistaa, että tarvittavat tietojen merkitykseen liittyvät

asiat tulevat varmasti kuvatuksi eivätkä tietojärjestelmän tekniseen toteutukseen liittyvät asiat, jotka ovat mm. ohjelmistojen päivittämisen yhteydessä muutosuhan alla.

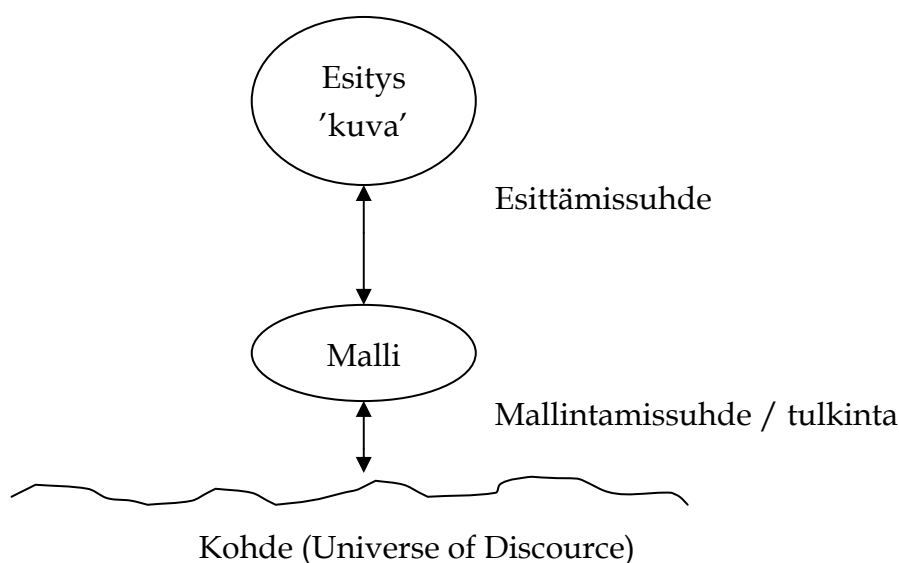
2.2. Mallintamisen prosessi

Nyt kun keskeisimmät termistöt on saatu mallintamisen osalta esiteltyä, on aika katsoa, miten itse mallintamisprosessi toimii. Kun jonkin kohdealueen käsitteistöä lähdetään kuvaamaan, tulee aina varmistua siitä, että käsitteiden asiallinen sisältö tulee niin tarkasti kuvattua, että jokainen asiasta kiinnostunut pysyy niistä ymmärtämään, mikä käsitteiden asiasisältö on ja miten ne liittyvät toisiinsa. Koska käsitteet voivat muodostaa hyvinkin monitasoisia hierarkioita, on kohdealueen mallintajan kyettävä paneutumaan kohdealueen kenttään vaa-ditun tason mukaisesti. Näin käsitehierarkian käyttäjän on mahdollisimman helppo päästä kohdealueen kuvaukseen kiinni.

Kun kohdealueen käsitteet ja niiden muodostama kokonaisuus erotellaan muista asioista, kuten tietojärjestelmien tietokanta- tai ohjelmistoratkaisuista, kyetään saavuttamaan huomattava etu; kohdealueen keskeiset asiat voidaan kuvata halutulla tarkkuustasolla ilman, että niitä joudutaan sekoittamaan alemmilla abstraktiotasoilla oleviin järjestelmän toteutukseen liittyviin yksityiskohtiin [Kangassalo, 2007, 4]. Tämä etu on nähtävissä hyvinkin nopeasti esim. tietojärjestelmän ohjelmistojen tai laitteistojen päivitysten yhteydessä, sillä jos järjestelmän suunnittelu ja toteutus on hoidettu hyvin, on järjestelmän asiasisältö (käsitesisältö) erillään tietojärjestelmän teknisistä ominaisuuksista kuten eri ohjelmistojen ohjelmistoversioista. Järjestelmän käsitesisältö ei tällöin altistu päivitysten ym. aiheuttamalle muutokselle vaan kykenee sopeutumaan uudelle alustalleen ilman ongelmia. Monissa organisaatioissa voidaan nähdä tiedon ja varsinaisen järjestelmän asiasisällön olevan huomattavasti pitkäikäisempää kuin sen alla olevan tietoteknisen toteutuksen [Kangassalo, 2007, 4 - 5].

2.2.1. Vaiheistus

Tietojärjestelmien osalta käsitteellisellä mallintamisella tarkoitetaan ”- - järjestelmän asiasisällön, ts. käsitteellisen sisällön tunnistamiseen, suunnitteluun ja määrittämiseen tarvittavan, systeemin kohdealuetta ja siihen perustuva käsitteellistä sisältöä kuvaavan, toteutuksesta riippumattoman käsitteellisen mallin luomista” [Kangassalo, 2007, 5]. Mallintamisprosessin vaiheet voidaan jakaa kolmeen vaiheeseen, jotka Kangassalo [1990b, 36] esittelee kuvan 2 avulla 1) Kohteeseen, 2) Malliin sekä 3) Esitykseen.



Kuva 2. Mallintamisprosessi [Kangassalo, 1990b, 36]

Kuvassa 2 kuvauksen kohdealue on merkitty termillä *kohde*, joka voi koostua mistä tahansa asioista. Käsitteellistä kuvausta kohteesta esittää kuvan termi *malli*, joka puolestaan voi koostua käsitteistä, tietämysalkioista ja käsitteiden välisistä suhteista muodostuvasta abstraktista rakenteesta. Tämä rakenne taas vastaa *mallintamissuhteen* osoittamalla tavalla kohteessa tunnistettuja ilmiöitä, niiden välisiä suhteita sekä molempien ominaisuuksia, jolloin sitä voidaan oikeastaan sanoa käsittekaavioksi. Mallintamissuhteen käänteinen suhde on *tulkinta*, jossa malliin sisältyvät kohdealueen ilmiöt, suhteet sekä ominaisuudet pyritään tunnistamaan. Malli pyritään tämän jälkeen konkretisoimaan jollakin käsitekuvauskielellä rakenteeksi, joka on kuvassa terminä *esitys*. [Kangassalo, 1990b, 36.]

2.2.2. Työmenetelmän prosessi

Mallinnusprosessi alkaa mallinnuksen tarkoituksen ja tavoitteen määrittämisestä. Niiden tulee olla hyvin tiedostettuja ja oikein valittuja, sillä ne ohjaavat mallinnusta koko työprosessin ajan. Kartoituksen yhteydessä tutkitaan mm. kenelle malli tehdään ja mitä tarkastelukulmaa siinä käytetään sekä millä tarkkuustasolla halutaan edetä. Itse mallintaminen alkaa kohteen objektien tai ilmiöiden ja niiden välisten suhteiden tunnistamisella. Kohdealueen objektit sekä niiden ilmiöiden eri komponentit on rajattava riittävän tarkasti, varsinkin jos mallinnettavat kohteet ovat luonteeltaan abstrakteja [Kangassalo, 1990b, 36 - 40].

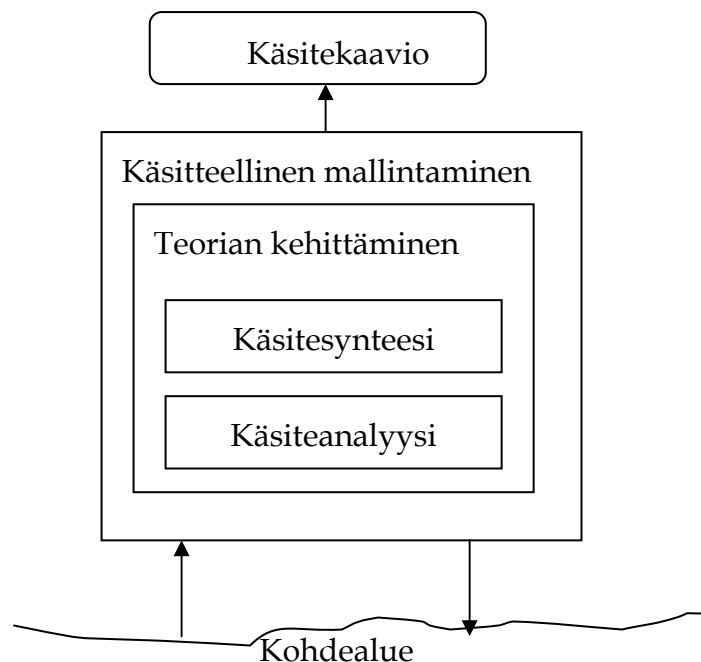
Mallintamisprosessin kolme vaihetta voidaan tarkentaa työprosessin alla *käsitteanalyysiin, käsitesynteesiin* sekä *teorian kehittämiseen*. Ensimmäisessä vaihees-

sa, *käsiteanalyysissä*, pyritään havaitsemaan kohdealueella työskentelevien ihmisten käsitteitä ja osoittaa niille tunnistavat nimet sekä kuvata ne. Tätä kutsutaan myös yleisesti dokumentointivaiheeksi. Tässä vaiheessa yleensä vain ne havaittavissa olevat käsitteet otetaan huomioon, jotka voidaan helposti johtaa kohdealueesta kuten asiakas tai tuote. Tällaista lähestymistapaa käytetään usein suhteellisen pieniin tietojärjestelmiin kuten palkanlaskentaan tai aikataulutusjärjestelmään. [Kangassalo, 1990a, 22 - 25.]

Seuraavassa vaiheessa pyritään löytämään ja määrittämään uusia käsitteitä, jotka voidaan määrittellä edellisen vaiheen pohjalta. *Käsitesynteesin* tarkoitus on yleensä selventää sekä parantaa havaittuja käsitteitä sekä luoda yleistyksen sekä aggregoinnin avulla vanhoista käsiteryhmistä uusia.

Viimeisessä vaiheessa pyritään löytämään käsitteerakenteesta sellaisia säännönmukaisuuksia, jotta niistä voidaan luoda oma kohdealuetta kuvaava teoria. Tällöin koko säännönmukaisen käsitteerakenteen määrittelyt sekä suhteet luodaan tyhjästä uudelleen, jotta voidaan havaita ne kaikkein sopivimmat ja tarkimmat käsitteet, jotka kuvaavat kohdealuetta täsmällisimmin.

Näiden vaiheiden jälkeen syntyvää käsitekaaviota voidaan perustellusti pitää kohdealuetta kuvaavana teoriana. [Kangassalo, 1990b, 36 - 40; 1990a, 22 - 25.] Kuva 3 esittää vielä työprosessin vaiheet sekä sisällyttämisen toisiinsa.



Kuva 3. Mallintamisen tarkempi työprosessi [Kangassalo, 1990b, 39]

3. Käyttöliittymä

Nyt kun olemme saaneet määriteltyä kaikki tarpeelliset käsitteellisen mallintamisen termit ja termistöt, voimme siirtyä tutkimaan käyttöliittymän käsitettä.

Käyn seuraavassa läpi muutamia tärkeitä käyttöliittymän käsitteen määrittelyjä, jonka jälkeen jatkamme tutkimaan miten käyttöliittymät ja niiden suunnittelu sekä käsitteellinen mallintaminen yhdistyvät.

3.1. Määritelmiä

Käyttöliittymällä (engl. user interface) tarkoitetaan kaikkia niitä artefaktin, esineen tai systeemin osia, joiden kanssa järjestelmän käyttäjä voi olla vuorovaikutuksessa. Tähän kuuluu kaikki ne fyysiset elementit kuten kahvat, vivut tai napit sekä havaintoon kytkeytyvät asiat kuten ikonit, värit, kuvat sekä viestit, mutta myös käsitteelliset asiat esim. mitä kaikki järjestelmän ikonit tai viestit oikein tarkoittavat tai mitä me ajattelemme tapahtuvan käyttöliittymän jostakin kahvasta. Hyvin suunniteltua käyttöliittymää on miellyttävä ja helppo käyttää ja se mahdollistaa kaikki ne ihmisten toimet, joita he haluavat suorittaa ko. systeemillä. [Benyon and Green, 1998, 1 - 2] Toinen hyvin samankaltainen, mutta pelkistetty määritys, on Sinkkosen [2009] esitys käyttöliittymästä: "kaikki mitä käyttäjä näkee tuotteesta tai järjestelmästä ja jonka kautta käyttäjä on vuorovaikutuksessa tuotteen kanssa". Määritelmät ovat hyvin lähellä toisiaan, mutta eivät kuitenkaan ole samanlaiset. Benyonin ja Greenin määritys pyrkii suuntaamaan ajatukset enemmän tietoteknisten ratkaisuiden pariin, kun taas Sinkkonen pelkästään mainitsee "tuotteen" sekä "järjestelmän", jossa käyttöliittymän voidaan sanoa esiintyvän. Molemmat määritelmät kuitenkin sopivat käyttööme erinomaisesti tarkasteltaessa hyvinkin abstrakteja tietojärjestelmän käyttöliittymiä, mutta muutamia tarkennuksia voidaan silti kuitenkin tehdä.

Vaikka käyttöliittymiä on mahdollista jaotella monella eri tapaa, yksi tarkempi erottelu on seuraavassa. Teoreettisen jaottelun mukaan käyttöliittymien osa-alueet voidaan jakaa 1) fyysiseen käyttöliittymään, joka käsittelee lähinnä ergonomiaan liittyviä asioita kuten istuma-asentoa, päätetyöskentelyä sekä näyttölaitesäteilyä, 2) rakenteelliseen käyttöliittymään, jolla tarkoitetaan käyttöliittymän noudattamaa rakennetta, sijoittelua sekä toimintaperiaatteita, 3) semanttiseen käyttöliittymään, joka määrittelee käyttöliittymän sisällön eli ihmisen ja tietokoneen välisen dialogin tarkoituksenmukaisuutta käytön kannalta, 4) pragmaattiseen käyttöliittymään, joka käsittelee varsinaisessa käyttötilanteessa ilmeneviä asioita kuten esimerkiksi vasteaikaa, ohjeita sekä virheilmoituksia sekä 5) psykososiaaliseen käyttöliittymään, joka sisältää mm. käyttäjän asenteet ja odotukset sekä uuteen työtapaan liittyvän muutosvastarinnan. [Kallio 1992, 12.] Kallion tekemä käyttöliittymien jaottelu on toimiva, mutta tutkielmassa tarkasteltavien tietojärjestelmien osuuden kannalta vain kohdat 2 ja 3 ovat mielenkiintoisia.

On kuitenkin olemassa myös toinen käyttöliittymien tarkempi erottelu, jonka pääpaino on erilaisissa vuorovaikutuskeinoissa. Tämä jaottelu löytyy Ben Shneidermanilta, johon Kallio viittaa [1992, 13]. Siinä Shneiderman jakaa käyttöliittymät 1) vuorovaikutustyylien mukaan esim. komento-, valikko- ja lomapohjaisiin käyttöliittymiin, 2) tiedon syöttötekniikoiden mukaan, jolloin tarkastellaan näppäimistön ja hiiren käytön ohella esim. tiedon syöttöä puhumalla, 3) näytön organisaation mukaan, jolloin käsitellään esim. värien käyttöä sekä sijoittelua näytöllä, 4) vasteajan mukaan, jolloin tarkastellaan helppokäyttöisyyteenkin liittyvää käyttäjän antaman syötteen ja tietokoneen antaman palautteen välistä aikaa, 5) virheiden käsittelyn mukaan, jolloin pyritään estämään käyttäjän tekemiä virheitä, 6) käyttöliittymien suunnittelun kannalta tärkeiden eri käyttäjien välisten yksilöllisten erojen mukaan sekä 7) käyttäjän toimintaa vuorovaikutuksessa tietokoneen kanssa selittävien ja ennustavien teorioiden mukaan [Kallio 1992, 13]. Shneidermanin jaottelun tärkeimmät luokat ajateltaessa käsitteellisen mallintamisen ja käyttöliittymien suunnittelun yhdistämistä sekä mukautuvien käyttöliittymien toteutusta ovat 6 ja 7.

Astetta karkeampi käyttöliittymien jaottelu ja kategorisointi voidaan vielä esittää jakamalla ne viiteen erilaiseen käyttöliittymämuotoon 1) tekstuaalisiin, esim. kirjat, 2) graafisiin, esim. Linux -järjestelmän graafinen työpöytäympäristö KDE, 3) komentopohjaisiin, kuten Unix -käyttöjärjestelmän Bourne again shell [GNU Project, 2009], 4) auditiivisiin, kuten Nuancen [2009] Dragon NaturallySpeaking 10 ryhmässä sekä 5) haptisiin, ts. kehollisiin kuten esim. Nintendo [2009] Wii -pelikonsolin liikeentunnistuspeli Wii Sports.

4. Käyttöliittymien suunnittelu ja käsitteellinen mallintaminen

Jotta voimme lähteä tutkimaan mukautuvien tai adaptoituvien käyttöliittymien suunnittelua ja toteutusta, on ensin ymmärrettävä, miten perinteinen käyttöliittymien kehitys tietojärjestelmien yhteydessä tapahtuu.

Perinteisessä käyttöliittymien suunnittelussa voidaan nähdä kaksi erillistä isompaa vaihetta: 1) käyttöliittymän toimintojen sekä 2) käyttöliittymän vuorovaikutuksen kartoitus yhdessä esityksellisen (yleensä visuaalisen) puolen kanssa. Toimintojen osalta suunnittelua ohjastaa ohjelmalle tai systeemille asetetut käyttötarkoituseriaa, mutta varsinainen toteutus riippuu aina käytettävistä ohjelmistoarkkitehtuureista sekä järjestelmäympäristöistä. Tästä syystä on erityisen tärkeää, että toimintojen kartoituksen yhteydessä käytettäisiin sellaisia suunnittelu- ja kohdealueen mallinnusmenetelmiä, jotka säilyttävät mahdollisimman paljon kohdealueen käsitteinformaatiota, mutta pystyvät abstrahoimaan riittävällä tarkkuustasolla järjestelmän tietotekniset yksityiskohdat [Kan-

gassalo 2007, 4 - 19]. Näin mahdollistetaan toimintojen mukauttaminen järjestelmästä toiseen minimaalisilla muutostoimilla.

Kun käyttöliittymän alla toimivat toimintaperiaatteet on saatu luotua alkaa varsinainen vuorovaikutuksen suunnittelu systeemin toimintojen sekä käyttöliittymän välille. Tähän osaan kuuluu myös käyttöliittymän visuaalinen, ts. esityksellinen suunnittelu, kuten järjestelmän yleisilmeen ja toimintojen yhdistäminen keskenään. Vaikka käyttöliittymän yleisilme voidaan luoda jo ennen systeemin toimintojen täydellistä kartoittamista, muokkaavat ne kuitenkin siinä määrin käyttöliittymän toiminnallisuutta ja visuaalista puolta, että perinteisesti käyttöliittymäsuunnittelu alkaa vasta varsinaisen systeemin toimintojen tultua pääpiirteittäin valmiiksi. Myös visuaalinen suunnittelu tarvitsee tuekseen vahvaa kohdealueen merkitysten ja toimien mallinnusvälinettä sekä kuvaajaa. Tässä kohtaa onkin hyvä tuoda esille käytettävyyden tutkimus ja sen soveltaminen käyttöliittymien suunnitteluun.

4.1. Käytettävyys

Koska käytettävyyden eri aspektit eivät ole tämän tutkielman polttopisteessä tai sen suurempina mielenkiinnon kohteina, käyn sen ajatuksia läpi vain pintaraapaisun tavoin sekä viitaten suurempaan käytettävyyden tutkimukseen [Usability Professionals' Association, 2009].

Käytettävyydelle ei ole yhtä yleispätevää määritelmää. Yksi useimmin viitattu käytettävyyden käsiterakenteista on kuitenkin Nielsenin [1993, 25] esittämä malli käytettävyyden osatekijöistä:

- *Tehokkuus*, kuinka nopeasti käyttäjä pystyy suorittamaan halutut toimet
- *Tyytyväisyys*, miten tyydyttävää tuotteen käyttö on
- *Muistettavuus*, miten helppoa on palauttaa laitteen käyttö sekä toiminnallisuus uudelleen mieleen
- *Opeteltavuus*, kuinka helppoa on oppia uuden laitteen vuorovaikutuskeinot ja toiminnot
- *Virheet*, miten minimoidaan käyttäjän tekemät virheet toimintoja suorittaessa.

Kun nykyaikaisia käyttöliittymiä suunnitellaan ja toteutetaan, tulisi niiden yhteydessä aina ottaa huomioon myös käytettävyyden näkökulma. Ilman helpposti käytettävää, hyvin toimivaa, tehokasta ja virheetöntä käyttöliittymää voi järjestelmän käyttäjä turhautua ja kokea jopa ahdistusta [Bailey et al., 2001] sitä käyttäessään. Sopivien suunnittelumallien sekä käytettävyydetutkimuksessa esiin tulleiden hyvien periaatteiden avulla on pyrittävä varmistamaan, että käyttäjälle voidaan tarjota sellainen käyttöliittymä, jonka avulla hän pystyy

helposti sekä vaivattomasti käyttämään kaikkia niitä toimintoja, jotka järjestelmä tarjoaa käyttöliittymänsä kautta.

Seuraavaksi lähden tutkimaan, miten käsitteellinen mallintaminen voi toimia käyttöliittymien suunnittelussa.

4.2. Käsitteellinen suunnittelu

Kasvavan informaatiomäärän hallitsemiseksi, muokkaamiseksi sekä hyödyntämiseksi, on yhteiskunnassamme jouduttu toteuttamaan hyvinkin vaihteleva määrä erilaisia ja yleensä suhteellisen kompleksisia tietojärjestelmiä. Mitä monimutkaisempia tietojärjestelmiä joudutaan rakentamaan, sitä isompi tarve on luoda näille järjestelmille toimivat, tehokkaat sekä ennen kaikkea selkeät käyttöliittymät, joiden avulla pystymme kommunikoimaan niiden alla, suurimmaksi osaksi piilossa olevien, informaatiokoneistojen kanssa [Benyon and Green, 1998, 1 - 2]. Ohjelmistokehittäjät käyttävät töissään erittäin kirjavaa joukkoa työkaluja kuten ketterät ohjelmistomenetelmät [Agile Alliance, 2009], EER [Elmasri and Navathe, 2004] tai esim. UML [Booch et al., 1998] suunnitellessaan järjestelmiä sekä niiden käyttöliittymiä. Kaikilla näistä systeemi- tai tietokantasuunnittelukielistä on omanlaisensa periaatteet ja kehysrakenteet, joiden mukaan suunnittelu tällöin tapahtuu. Kun järjestelmää kehittää monta työryhmää, voi käytössä olla useita eri työkaluja, jolloin tarve kommunikoida mallinnettavasta kohdealueesta kasvaa entisestään. Tarvitaan kunnollinen käsitteellistämisen prosessi kohdealueen kohteista, ilmiöistä sekä niiden välisistä suhteista, mutta myös mallintamiskieli, jolla voimme mahdollisimman tehokkaasti välittää ihmisten sisäisten käsittekaavioiden informaatio ulkoiseksi käsittekaavioksi ja tarkoituksenmukaiseksi rakenteeksi. Palaan tällaisiin mallinnus- sekä kuvauskieliin vielä tulevissa luvuissa, mutta seuraavaksi lähdän tarkastelemaan käsitteellisen ja fyysisen mallin yhteyttä suunnitteluun.

4.2.1. Fyysinen ja käsitteellinen malli

Fyysisen ja käsitteellisen mallin erottaminen käyttöliittymien suunnittelussa on hyvin tärkeää. Kun puhutaan käsitteellisestä mallista, viitataan aina asioihin, jotka kuvaavat kohdesysteemin olemusta sellaisena, mikä täyttää ne käsitteelliselle mallille asetetut tarkoitukselliset, rakenteelliset sekä funktionaaliset määrittelyt, jotka sille on asetettu. Kun suunnittelija alkaa miettiä käyttöliittymän visuaalista tai ylipäätään esityksellistä puolta sekä vuorovaikutusta systeemin kanssa, puhutaan fyysisestä mallista. Tähän jaotteluun kuuluu myös tehtäväprosessien määrittely eli nk. *funktionaalinen jaottelu*, joko käsitteellisen tai fyysisen mallin puolelle, jossa toiset asiat suunnitellaan järjestelmän ja toiset käyttäjän tehtäviksi. Vaikka tämä jaottelu tulee tehdä, ei se silti kiellä muiden metodi-

en käyttöä käyttöliittymän mallin suunnittelussa. Jaottelu on luonteeltaan iteraatiivinen, jossa käyttöliittymän käsitteiden analysoijat sekä visuaalisen puolen suunnittelijat voivat fyysisen sekä käsitteellisen mallin avulla tuottaa yhä tarkempia fyysisiä malleja sekä samalla ymmärtää myös käsitteellistä mallia paremmin. [Benyon and Green, 1998, 4.] Jotta käyttäjät voivat kommunikoida käyttöliittymän avulla alla olevan järjestelmän kanssa, täytyy käsitteellinen malli ensin ”upottaa” fyysiseen malliin. Fyysinen malli keskittyy näin käsitteellistä mallia enemmän kysymyksiin, missä ja koska toimet ja päätökset käyttöliittymässä tehdään, mitkä niistä toistetaan, onko vaihtoehtoisia toimintoja sekä missä järjestyksessä kaikki suoritetaan. [Benyon and Green, 1998, 4 - 5.]

4.2.2. Toiminnalliset ja esitykselliset piirteet

Toiminnallisilla piirteillä tarkoitetaan fyysiseen malliin pakattuja käsitteellisen mallin operaatioita, jotka listataan eräänlaisina dialogeina siitä, mitä käyttöliittymä näyttää tai käyttäjän on tehtävä saavuttaakseen tietyn asian käyttöliittymän avulla [Benyon and Green, 1998, 5]. Esimerkkinä tällaisesta dialogista voidaan pitää käyttötapausten [Alhir, 2009] tuottamista jostakin järjestelmän osasta.

Esityksen piirteillä tarkoitetaan yksinkertaisesti kaikkia niitä käyttöliittymän esityksellisiä asioita, jotka suunnittelijan on mietittävä ja luotava, jotta käyttöliittymän toiminnot, dialogit yms. tulevat käyttäjälle selvästi ja ymmärrettävästi esiin [Benyon and Green, 1998, 5 - 6]. Tällaisia ovat esimerkiksi painikkeiden painamisesta johtuvat värimuutokset tai animaatiot.

4.3. Mallien käyttö suunnittelussa

Mallilla tarkoitetaan jotakin sellaista esinettä tai asiaa, joka esittää jotakin kohdealueen kohdetta, ilmiötä tai niiden välistä suhdetta, nostaen esiin vain sen tarkoituksenmukaisimmat piirteet ja piilottaen epäolennaiset. Malli tarjoaa tietynlaisen näkökulman kohdealueeseensa abstrahointimekanismiensa kautta, jotka heijastuvat sen toteutetuissa käsitteiden sisällöissä sekä rakenteissa. [Benyon and Green, 1998, 6 - 7.] Mallit tarjoavat suunnittelijoille keinon nähdä, miten kehitettävänä olevan käyttöliittymän tai järjestelmän osa voisi toimia, kun se on toteutettu tiettyjen periaatteiden avulla. Esimerkkeinä tällaisista malleista voidaan pitää mm. paperiprototyyppejä esiteltäessä käyttöliittymän toimintoja tai vaikkapa animaatioiden avulla tehtyjä simuloituja proteiinien kolmiulotteisten rakenteiden selvittelyä [University of Washington, 2009]. Mallia tehtäessä on aina pidettävä mielessä, kenelle sitä tehdään, mitä tekniikoita sen toteuttamiseksi käytetään ja mille abstraktiotasolle se sijoitetaan. Mallin käsitteelliset hyödyt heijastuvat sen tehokkaasta analyttisestä, kommunikatiivisesta sekä

selittävästä voimasta esittää rakenteita, operaatioita sekä rajoituksia. Myös fyysisen puolen ominaispiirteiden esittäminen sopivan alustan avulla luo mahdollisuuden mm. käytettävyyden tarkastetulle. [Benyon and Green, 1998 6.]

4.3.1. Mallin keinot esityksessä

Malliin sovelletaan erilaisia informaation keräysmenetelmiä, joita käyn yleispiirteisesti seuraavassa läpi Benyon and Greenin esittämien vaihtoehtojen avulla [1998, 8 - 9]:

- *Aggregoinnin* avulla kerätään kohdealueen ilmiöistä yksittäinen kohteen esitys, jossa on otettu huomioon jokin määrä ilmiön eri ominaisuuksia. Käsitteellistämisen avulla liitetään tällöin intensionaalisen sisällyttymissuhteen avulla yksi tai useampia määriteltäviä käsitteitä määreineen määriteltävään käsitteeseen.
- *Luokittelussa* tunnistetaan prosessin avulla erilaisten kohdealueen ilmiöiden jaettuja ominaisuuksia, joiden perusteella ne asetetaan omaan luokkaansa. Käsitteellistämisen avulla määriteltävään käsitteeseen tällöin liitetään joko kaikki määrittelevien käsitteiden yhteiset tunnusmerkit tai joukko tunnusmerkkejä määrittelevien käsitteiden yhteisiä tunnusmerkkejä.

[Kangassalo 1990b, 51, 64.]

Viimeisinä pääkohtina mallin informaation tarkastelun tavoissa ovat kohdealueen yleisen rakenteen tarkastelu, funktionaalinen esitys sekä systeemin dynamiikan esitys [Benyon and Green, 1998, 8 - 9]:

- *Rakenteen tarkastelussa* voidaan luoda katsaus siihen, miten koko haluttu kohdealueen osa tai kokonaisuus rakentuu, mikä kuvautuu käsitteellisen mallintamisen osassa esim. käsittekaavion ulkoistamisessa rakenteeseen, ts. graafiseen muotoon.
- *Funktionaalisisessa kartoituksessa* tutkitaan miten systeemin jokin osa toimii sisällä. Tämä voidaan nähdä käsitteellisessä mallinnuksessa tutkimalla *esitystä* kohdealueesta, jonne kuvautuu kohdealueen kaikki ehdot sekä rajoitukset, mukaan lukien funktionaaliset systeemin osat.
- Dynaamisen esityksen kautta, voidaan mallissa havainnoida miten systeemi vaihtaa eri tiloista toisiin. Tarkastelun kohteena on siis esityksen rakenteen kautta tapahtuvassa toiminnallisen tapahtumaketjun lopputuloksissa.

4.4. Mallinnuskielet sekä ohjelmistot

Tässä kohdassa tutustutaan hieman käyttöliittymien mallinnuksessa käytettävään mallinnuskieliin sekä siihen, miten hyvin ne kykenevät toteuttamaan käsit-

teiden kuvaamisen halutulta kohdealueelta. Tässä kohtaa on hyvä huomioida, että vaikka tutkielmani muutamissa osissa viitataan UML-kuvauskielen toimintoihin, en käy ko. suunnittelukieltä tässä läpi. UML on suurimmaksi osaksi kehitetty pelkästään ohjelmistokehityksen tarpeisiin luokka, aktiviteetti- ja käyttöskenaarioineen, eikä se sisällä sellaisia kiinnostavia piirteitä, joita tulisi tämän tutkielman näkökulman osalta asettaa esiteltäväksi.

Kun olemme päätyneet mallintamaan jotakin reaali maailman osaa, tarvitsemme ulkoisen käsitekaavion kommunikoidaksemme sisäisen käsitekaavion rakenteita, ilmiöitä, kohteita sekä niiden välisiä suhteita. Tällöin astuu esiin mallinnuskielet, joiden avulla voimme pyrkiä enemmän tai vähemmän tehokkaasti kuvaamaan ne informaatio sisällöt, jotka sijaitsevat ihmisen mielen sisäisen käsitekaavion rakenteissa. Vaikka informaatiota häviää siirryttäessä sisäisestä käsitekaaviosta ulkoiseen [Kangassalo, 2007, 5], tulee mallinnuskielen sekä menetelmän valinnassa pyrkiä saavuttamaan mahdollisimman monipuolisesti sääntöjä, ehtoja sekä valintoja toteuttava kieli tai ohjelmisto. Kangassalo [2007, 4 - 6] toteaa, että käytettäessä formaaleja kuvausmenetelmiä on ihmisen mahdollista luoda sellaisia käsitekaavioita, joita hän ei välttämättä vielä omista käsitekaavioissaan tunnista tai hallitse. Kangassalo myös jatkaa, että tällöin mallinnusmenetelmän käyttäjän on huolellisesti perehdyttävä ulkoisen käsitekaavion luomiin käsitekuvauksiin sekä niiden välisiin suhteisiin, jotta hän kykenee omaksumaan ne osaksi omaa sisäistä käsitekaaviotaan. Tällainen prosessi on keskeisten käsitteiden sekä rakenteiden kannalta hyvin merkittävä, kuten Kangassalo [2007, 5] linjaa.

4.4.1. Enhanced Entity-Relationship Model

Enhanced Entity-Relationship Model (EER) on korkean tason käsitteellinen mallinnuskieli, jonka pääasiallinen kohdealue on tietokantojen mallintaminen [Elmasri and Navathe, 2004]. Sitä käytetään kuitenkin myös mm. tietojärjestelmien alueella mallintamaan sen toimintoja sekä rajapintoja. EER-kielen kuvantamismahdollisuudet ovat erityisen hyvät, sillä sen käytössä ovat mm. seuraavat ulkoisen käsitekaavion käytettävissä olevat muodolliset ilmaukset: olio, oliotyyppi, vahva oliotyyppi, heikko oliotyyppi, suhde, suhdetyyppi, attribuutti, attribuutin arvo sekä olioiden, attribuuttien ja suhteiden nimet. Nämä ilmaukset löytyivät jo EER:n edellisestä versiosta, jota laajennettiin vielä 1) hierarkisilla suhdetyypeillä, jolloin entiteeteillä voi olla sekä ylä- että alaluokka, 2) entiteettien perinnällä, 3) entiteettien erikoistumisella, jossa entiteetteihin voidaan liittää entistä tarkemmin vain tiettyjä, haluttuja, ominaisuuksia, 4) yleistyksellä, joka on erikoistumisen vastakohta, 5) luokkarajoituksilla entiteettien

suhteissa ja perinnässä sekä 6) täydellisellä ja osittaisella erikoistumisella [Elmasri and Navathe, 2004].

Edellisen kaltainen lista muodollisista ilmauksista antaa jo erittäin hyvät mahdollisuudet kuvata hyvinkin monimutkaisia tietokantajärjestelmiä, mutta EER:n ongelma puhtaasti käsitteellisenä mallinnuskielenä syntyy sen puutteesta käsitteiden intensionaaliseen sisällyttymiseen. Kun kohdealueena on erittäin monimutkainen ja monista osista koostuva käsitteellinen kokonaisuus, joudutaan mallintamisessa tekemään EER:n kanssa kompromisseja. Periytyminen, rajoitusten sekä erikoistumisen tultua kieleen, voidaan EER:n nähdä lähestyvän ontologisia kuvauskieliä, joissa Kangassalon [2007, 6] mukaan on jo varsin laajat ilmaisumahdollisuudet erilaisten käsittekaavioiden ilmaisuun.

4.4.2. Entity-Relationship Modeling of Information Artefacts

Entity-Relationship Modeling of Information Artefacts (ERMIA) on käyttöliittymien suunnitteluun sekä toteutukseen räätälöity käsitteellinen mallinnuskieli. Se pohjautuu suurimmaksi osaksi Elmasri and Navathen [2004] ER-malliin. Kieleen formalisoitujen ominaisuuksien avulla suunnittelijat voivat esittää sopivalla tarkkuustasolla olevia abstrahointeja kohdealueen järjestelmästä sekä sen käyttöliittymästä, niin käsitteellisellä kuin fyysisellä tasolla. ERMIA kykenee näin esittämään ja hakemaan niitä käsitteellisiä rakenteita, joita siihen on tallennettu. Pääajatuksena on siis yhdistää käyttöliittymän suunnittelijat sekä ohjelmistokehityksen ammattilaiset, jotta yhteinen kommunikaatio- sekä suunnittelualusta olisi heidän välillään mahdollinen.

ERMIA eroaa ER-kielestä kuitenkin siinä, että se on ottanut osaksi mallinusrakennettaan yleensä vain olio-ohjelmoinnissa nähtäviä ominaisuuksia kuten perinnän, kapseloinnin sekä luokkaperiaatteen, joita mm. UML-kieli tukee [Booch et al., 1998]. ERMIA:ssa käytetyt entiteetit ovat myös yhdenmukaisia UML:ssä käytettävien objektien kanssa, mistä seuraa, että ERMIA:ssa voidaan käyttää UML:n tapaisia käyttökenskenarioita mallinnettaessa käyttäjän ja järjestelmän välistä vuorovaikutusta. Merkittävänä erona mm. UML:n käyttämiin objekti- eli luokkarakenteisiin on ERMIA:n entiteeteissä käyttämä toiminnan kuvausten eksplisiittisen sisällyttämisen vapaaehtoisuus. Tästä on huomattavaa etua mallinnettaessa vuorovaikutusta käyttäjän ja järjestelmän välillä, kun pyrkimyksenä on keskittyä koko informaatiostytemin tiedon rakenteeseen ja esitykseen. Kun järjestelmän käyttäjäaktiviteetit on saatu mallinnettua sekä määritettyä, voidaan ERMIA:n entiteetit toteuttaa suoraan UML:n käyttämiksi objekteiksi. [Benyon and Green, 1998, xii.]

ERMIA:ssa ei EER:n tai UML:n tavoin ole toteutettu intensionaalista sisällyttymissuhteen mallintamista käsitteiden suhteiden välille, mutta sen ominai-

suudet luoda silta sekä käsitteellisen aineiston että fyysisen eli havainnollisen materiaalin välille ovat erinomaiset.

4.4.3. Concept D

Kaikista kolmesta nyt esitellystä käsitteellisen kuvauksen mallinnuskielestä Concept D on ainoa, joka hallitsee intensionaalisen sisällyssuhteen kohdealueen käsitteiden välillä. Se on siis graafinen kuvauskieli käsiterakenteiden esittämistä varten.

Concept D voidaan nähdä koostuvan kolmesta tärkeästä osasta: 1) Concept D/D, jonka avulla tehdään kohdealueen tietämyksen keräystä käyttäjiltä ja sen käsiterakenteiden kuvaamista, 2) Concept D/CS, jolla kuvataan koko kohdealuetta koskevia käsitekaavioita ja siitä tehtyjä osanäkemyksiä sekä 3) Concept D/CQL, jolla voidaan tehdä käsitetasolta tapahtuvia sovellustietokannan käsitelyitä [Kangassalo, 1990b, 3]. Perusajatuksena Concept D:ssä on käsitteen ymmärrys keskeisenä inhimillisenä tietämyksen yksikkönä. Koska kieli perustuu käsitteiden intensionaaliseen sisällyssuhteeseen, pyritään mallinnuskielen avulla "kuvausta laadittaessa kuvaamaan mallin tekijällä kohdealueesta olevaa tietämystä eikä yritetä suoraan laatia kohdealueen pelkistettyä kuvaa" [Kangassalo, 1990b, 3 - 4]. Aiemmin esitellyt ER-malliin perustuvat mallinnuskielet toteuttavat ainoastaan ekstensionaalista lähestymistapaa.

Työ on kielen avulla asteittain tarkentuvaa, jossa määrittelemällä kohdealueella esiintyvät ja käytetyt käsitteet voidaan kuvaus laatia mallinnuskielen graafista formalismia hyväksi käyttäen aina sopivalle tarkkuustasolle asti. Tämän jälkeen kuvataan kohdealueen rakenne sekä käyttäytyminen edellisessä vaiheessa saadun käsitteistön avulla. Tällainen Concept D -kielen avulla tehty kohdealueen kuvaus on yleensä monesta eri hierarkiatasosta koostuva rakenne. Näin on mahdollista rajoittaa työskentely tarpeen mukaan halutulle tasolle ilman, että jouduttaisiin hallitsemaan samanaikaisesti kaikille tasoille kuuluvia yksityiskohtia. [Kangassalo, 1990b, 2 - 4.]

Concept D eroaa myös muiden ominaisuuksiensa suhteen niin EER- kuin ERMIA-kielestä. Se sisältää monia yhteiseksi laskettavia periaatteita, mutta myös paljon muuta, joista seuraavassa esitellään pääkohdat [Kangassalo, 1990b, 11 - 170]:

- Concept D sisältää yleistyksen sekä aggregoinnin lisäksi myös tiedon *transformoinnin* eli ominaisuuden muunnoksen käsitekaaviossa.
- Kielessä on useita eri tapoja tehdä *ehdollisuus*- ja että *rajoitus* sääntöjä käsitteiden ja niiden suhteiden välille.
- Käsitteiden tunnisteilla on tarvittaessa mm. *kootun tunnisteiden* sekä *kantaman tai voimassaolon* määritteet.

- Myös *temporaaliset* eli ajalliset määrittelyt ovat mahdollisia.
- *Tapahtumien* sekä *prosessien* kuvaamista tuetaan vahvasti.

Concept D -kielen voi hyvin nähdä toimivan käyttöliittymien mallintamisessa, vaikka siinä ei varsinaisesti ole fyysisen puolen esitysmuotoja ohjelmistokehityksen tarpeisiin. Voisin kuitenkin nähdä, että tällaiset ominaisuudet olisivat siihen mahdollisia toteuttaa, sillä kielen laaja käsittekaavion kuvantamis- ja esittämismahdollisuudet luovat erittäin vankan pohjan monipuolisten ratkaisujen kehittämiseksi.

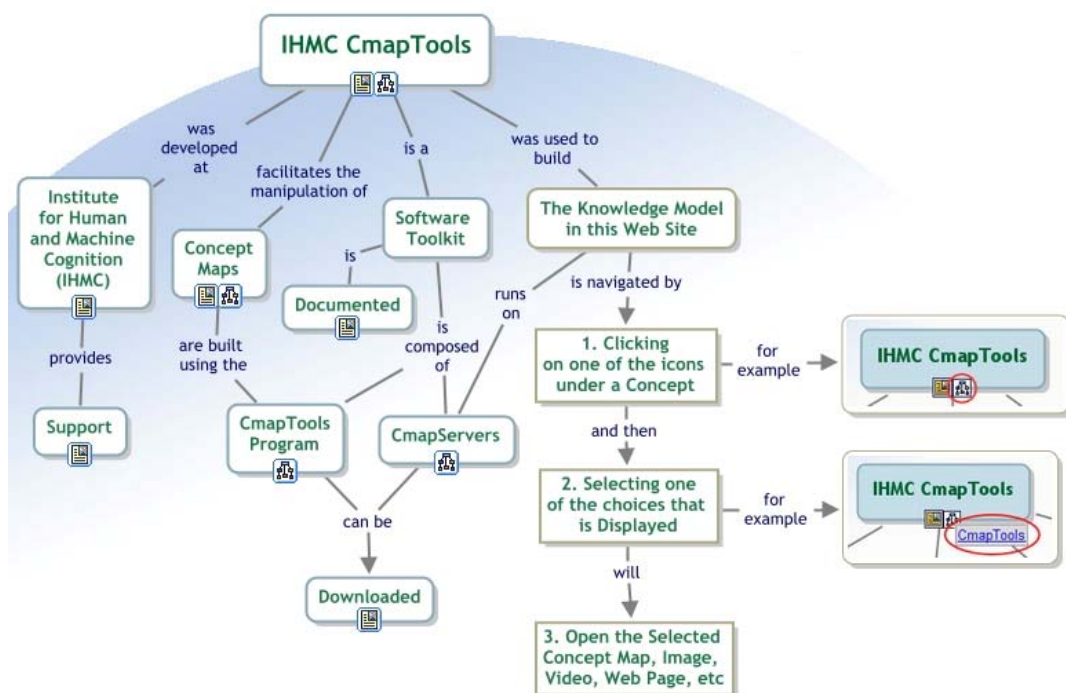
5. Kohti mukautuvia käyttöliittymiä

Mukautuviksi tai adaptoituviksi käyttöliittymiksi voidaan ymmärtää kaikki sellaiset tuotteet, joiden käyttöliittymät sopeutuvat tai muutoin mukautuvat käyttäjänsä tapaan käyttää ko. tuotetta [Sinkkonen, 2009]. Sopeutumisella voidaan ymmärtää esimerkiksi työntekijän työtapojen mukaan tapahtuvaa toimintojen ryhmittelyä tai kokonaisen ohjelman toimintojen määrittelyä agenttien tai käyttäjiensä halujen sekä tarpeiden mukaan [Taghezout et al., 2008].

Mukautettavuuden voidaan siis yleisesti nähdä olevan sidonnainen järjestelmän kanssa vuorovaikutuksessa olevaan käyttäjään. Tällaisten järjestelmien kehittämisessä käsitteellisellä mallintamisella on iso osa. Jos puhutaan esimerkiksi suurista tieto- ja informaatiojärjestelmistä, joissa saattaa olla satoja tai jopa useita tuhansia erilaisia käsitteitä, joita käyttäjän tulisi tavalla tai toisella systeemin käyttöliittymän kautta käyttää, voi tehokkaan, virheettömän sekä helpon käyttöliittymän toteuttaminen olla hyvinkin haasteellista. Kun tähän liitetään vielä ajatus käyttäjän toimiin, tarpeisiin sekä haluihin mukautuvasta käyttöliittymästä liikutaan jo lähes mahdottomalta tuntuvalla suunnittelun sekä toteutuksen alueella. Menemättä tarkemmin tällaisen käyttöliittymän fyysisen tai toiminnallisen toteutuksen syövereihin, voidaan järjestelmän suunnittelun osalta kuitenkin todeta, että ilman perusteellista systeemin kohdealueen käsitteellistä mallintamista ei halutun tapaista käyttöliittymää ja järjestelmää perusteellisesti voida toteuttaa. Käsitteellinen mallintaminen on suurilta osin abstraktia ajattelua vaativaa työtä, ja sitä voidaan erittäin hyvin verrata esimerkiksi matematiikkaan sillä erotuksella, että matematiikassa käsitellään aina ”puhtaita” rakenteita vailla mitään merkityksiä, kun taas käsitteellisessä mallintamisessa käsitellään ainakin muodoltaan vastaavan tyyppisiä rakenteita. Mallintamisessa rakenteilla on lisäksi sovelluksen antama asiasisältö, joka tulee mallinnuksessa ottaa huomioon ja jota ei ilman perusteltua syytä voi edes muuttaa, koska se kuvaa käyttäjän käsitystä asianomaisesta käsitteestä tai käsitteistä. [Kangasalo, 2007, 18 – 19.] Voidaan siis perustellusti kysyä, onko järjestelmän nykyisiä

tai tulevia käyttäjä sitten mahdollista hyödyntää tällaisen systeemin mallinnuksen eri vaiheissa?

Yksi tällainen vaihtoehto voitaisiin nähdä käyttäjien kouluttamisessa mallinnuskielen käyttöön, jolloin käyttäjillä olisi mahdollisuus tuoda omat käsittekaavionsa kohdealueesta esille, kuten Concept D -kielen avulla tehdyssä yrityksen tietosisällön mallintamisessa [Kangassalo, 1984]. Käyttäjiltä voitaisiin myös pyytää kohdealueesta tehtyjä käsittekarttoja, joita voitaisiin myös hyödyntää mallintamisen apuna. Esimerkkinä tällaisesta ohjelmasta voisi olla Novakin ja Cañasin [2006] CmapTools.



Kuva 4. CmapTools:lla luotu ohjelman oman verkkosivun käsittekartta

Kuvassa 4 on yksi esimerkki CmapToolsin tuottamasta käsittekartasta. Kuvan voidaan todeta näyttävän erehdyttävästi miellekartalta, mutta ero niihin löytyy kuitenkin CmapToolsin tavasta pyrkiä ohjaamaan käyttäjää luomaan käsitteitä, joilla voidaan nähdä selkeitä sisällyksiä toisiinsa sekä hierarkkisia tasoja. Vaikka käsittekarttajärjestelmät eivät varsinaisesti lukeudu perinteisiin mallintamismenetelmiin, voisi niiden käyttöä silti hyödyntää suunnittelun yhtenä osana.

Käsitteellisellä mallintamisella voidaan käyttöliittymien sekä niiden alla toimivien järjestelmien osana nähdä olevan suuri vastuuosuus siitä, että systeemin kohdealueen kaikki tarpeelliseksi havaitut kohteet, ilmiöt sekä niiden väliset

suhteet tulevat sekä oikein että tehokkaasti jäsenneyiksi suunniteltavan systeemin toteutuksessa. Tähän vastuukysymykseen nojaten voimme päätellä, että tulevien mukautuvien käyttöliittymien sekä järjestelmien suunnittelussa sekä toteutuksessa on otettava käsitteellisessä mallinnuksessa saavutetut kohdealueen esitykset tarpeellisella tarkkuustasolla huomioon. Jos tällainen käsitteisiin nojautuva mukautuvuus pystytään järjestelmissä toteuttamaan, astutaan silloin aimo askel kohti mukautuvien käyttöliittymien suuria mahdollisuuksia.

6. Yhteenveto

Käsitteellinen mallintaminen voidaan nähdä yhtenä tärkeimmistä osa-alueista järjestelmän sekä käyttöliittymän suunnittelussa. Sen avulla saavutetut edut kohdealueen informaation keräämisessä sekä mallintamisessa ovat kiistattomat. Vaikka tutkielmassa esitetyistä mallinnuskielistä vain yksi oli sorvattu käyttöliittymien suunnittelua silmällä pitäen, voidaan silti Concept D -kielen ominaisuuksia pitää niistä parhaimpina sekä monipuolisimpana sen mahdollistaessa jopa erittäin kompleksisten systeemien ja niiden kohdealueiden informaation keräämisen mallintamisen avulla. Tulevaisuuden monipuolisimmat ja käytettävyydeltään sulavimmat mukautuvat käyttöliittymät voidaan nähdä toteutettavan käsitteitä painokkaimmin korostavien tietojärjestelmien osana.

Mukautuvien käyttöliittymien ongelmaa voidaan lähestyä yllättävänkin monesta näkökulmasta. Käsitteellisen mallintamisen valitseminen tutkielman polttopisteeseen käyttöliittymien suunnitteluratkaisuissa toi kuitenkin niitä tarvittavia teorioita tutkimuksen jatkoon kannalta esiin, että näkökulman valinta tuntui loppuun asti perustellulta. Teorioiden esittäminen vei tutkielmasta yllättävän paljon aikaa sekä vaati keskittymistä, joka rajatun aikahaarukan puitteissa karsi mm. tutkittavien ohjelmien määrän vain olennaisimpiin sekä rajasi varsinaisen mukautuvista käyttöliittymistä kirjoitetun osuuden ajateltua pienemmäksi. Aiheen jatkokäsittelyä onkin ehkä juuri tästä syystä täysin mahdollista ajatella toteutettavan. Näkökulma tällaisessa tutkimuksessa voisi olla esimerkiksi varsinaisen toteutuksen havainnoinnissa tai jopa itse toteuttamisessa.

Viiteluettelo

Agile Alliance 2009, Public resources. Checked 17.12.2009.

<http://www.agilealliance.org/resources>.

Alhir S. S. 2009. Understanding Use case modeling. Checked 17.12.2009.

<http://www.methodsandtools.com/archive/archive.php?id=24>.

- [Bailey et al., 2001] Bailey B. P., Carlis J. V. and Konstan J. A., The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In: *Human-Computer Interaction - INTERACT'01*, 593 - 601.
- [Benyon and Green, 1998] Benyon D. and Green T., *Conceptual Modeling for User Interface Development*. Springer-Verlag, New York, 1998.
- [Booch et al., 1998] Booch G., Jacobson, Rumbaugh J., *Unified Modeling Language User Guide*. Addison-Wesley, Cambridge, MA, 1998.
- [Elmasri and Navathe, 2004] Elmasri R., Navathe S. B., *Fundamentals of Database Systems*. 4. ed. Addison-Wesley, Boston, MA, 2004.
- GNU Project 2009. Bourne again shell. Checked 17.12.2009.
<http://tiswww.case.edu/php/chet/bash/bashtop.html>.
- [Kallio, 1992] Kallio T., *Käyttöliittymät ja Niiden Suunnittelu*. Anson Oy, Myllykoski, 1992.
- [Kangassalo, 1990a] Kangassalo H., Foundations of conceptual modelling: a theory construction view. In: Kangassalo, H., Ohsuga, S., Jaakkola, H., (eds.), *Information Modelling and Knowledge Bases*. IOS Press, Amsterdam, (1990), 19-35.
- [Kangassalo, 1990b] Kangassalo H., Kuvauskieli käsitteellistä mallintamista varten. Concept D. Luonnos, Tammikuu, 1990.
- [Kangassalo, 2007] Kangassalo H., *Käsitteellinen Mallintaminen*. Luonnos, Syyskuu, 2007.
- [Kangassalo and Aalto, 1984] Kangassalo H. and Aalto P., Experiences on user participation in the development of a conceptual schema by using a concept structure interface. In: Shackel, Brian (ed.) *INTERACT 84 - 1st IFIP International Conference on Human-Computer Interaction (Sep. 4-7 1984)*, London, 597-601.
- [Kauppi, 1967] Kauppi, R., *Einführung in die Theorie der Begriffssysteme*. University of Tampere, Ser. A, Vol. 15, 1967.
- Korpela, J. 2009. Pienehkö sivistyssanakirja. Tarkistettu 17.12.2009.
<http://www.cs.tut.fi/~jkorpela/siv/>.
- [Nielsen, 1993] Nielsen J., *Usability Engineering*. Academic Press, Inc., San Diego, 1993.
- Nintendo 2009, Wii sports. Checked 17.12.2009.
<http://www.nintendo.com/games/detail/1OTtO06SP7M52gi5m8pD6CnahbW8CzxE>.
- [Novak and Cañas, 2006] Novak J. D. and Cañas A. J., The theory underlying concept maps and how to construct them. Technical Report IHMC CmapTools, Jan 2006. Also available as
<http://cmap.ihmc.us/Publications/ResearchPapers/TheoryUnderlyingConceptMapsHQ.pdf>.

Nuance 2009, Dragon naturallyspeaking 10. Checked 17.12.2009.

<http://www.nuance.com/talk/>.

Sinkkonen, I. 2009. Käytettävyysanasto. Tarkistettu 17.12.2009.

<http://www.adage.fi/blogi/2002/kaytettavyysanasto>.

University of Washington, 2009. Rosetta@home. Checked 17.12.2009.

<http://boinc.bakerlab.org/rosetta/>.

Usability Professionals' Association 2009. Usability resources. Checked 17.12.2009.

http://www.upassoc.org/usability_resources/index.html.

[van Griethuysen, 1982] van Griethuysen J. J., Concepts and terminology for the conceptual schema and the information base. In ISO/TC97/SC5-N695, van Griethuysen J. J. (ed), ANSI, 1982.

Ohjaaja: Hannu Kangassalo

Mediakasvatus peruskoulussa

Jenni Päckilä

Tiivistelmä.

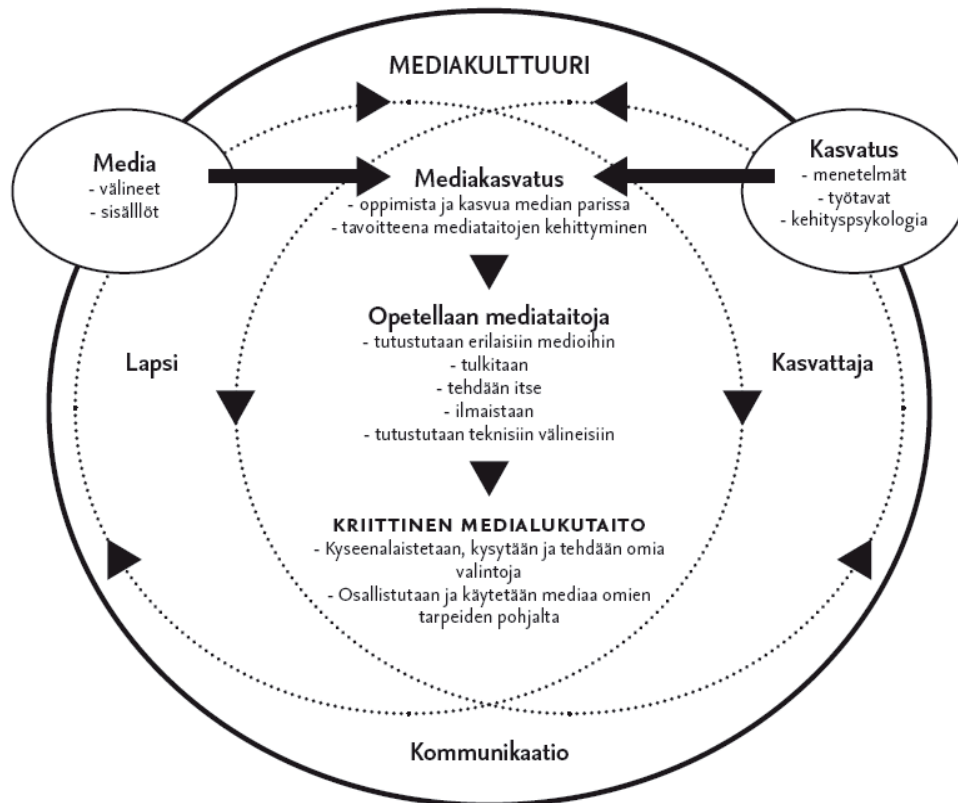
Tämä tutkielma on tehty osaksi taustakartoitusta peruskoulun yläasteen yhteisöllisyyden kehittämishankkeelle, Kohtaamisen areenat koulussa. Hankkeen yhtenä tarkastelun kohteena, ja mahdollisena välineenä kehittämistyössä, on mediakulttuuri, koska se vaikuttaa yhä enemmän myös koulutyöskentelyyn. Tutkielmassa käsittelemäni aiheet olen valinnut mediakulttuurin alueelta hankkeen näkökulmasta. Käsittelen tutkielmassa mediakasvatuksen näkökulmia ja määrittelyjä. Medialukutaitoa käsittelen erityisesti kulttuurisen ja visuaalisen lukutaidon osalta. Selvitän tutkielmassa myös, miten lapset ja nuoret käyttävät mediaa ja mediakasvatusta perusopetuksessa. Esittelen lisäksi muutamia mediakasvatuksen verkkomateriaaleja.

Avainsanat ja -sanonnat: Mediakasvatus, medialukutaito

CR-luokat: K.3.2

1. Mediakasvatus ja mediapedagogiikka

Mediakulttuurin alle sijoittuu mediakasvatuksen käsite (kuva 1). Mediakasvatukselle on olemassa monenlaisia määrittelyjä. Wikipedian mukaan mediakasvatus on kasvatusta ja viestintätieteen poikkiteollinen tutkimuksen sekä opetuksen alue, jossa opitaan havaitsemaan ja tarkastelemaan median vaikutuksia ja merkityksiä yksilöille ja yhteiskunnille ja sen tavoitteena on medialukutaidon kehitys. (Wikipedia, 2007.) Käytän itse termejä tämän määritelmän mukaisesti. Kupiainen ja muiden (2007, 16) mukaan mediakasvatuksen käsite on vakiintunut suomalaisen käytäntöön 2000-luvulla. Mediakasvatusta ei löydy peruskoulun opetussuunnitelmista, mutta sieltä löytyy kyllä medialukutaito. Tämä sekavuus käsitteissä johtuu osaksi kansainvälisestä keskustelusta ja eri kulttuureista. (Kupiainen, Sintonen ja Suoranta 2007, 16.) Mediakasvatuksen taustalla on kuitenkin aina pedagoginen ote ja pyrkimys taitojen, tietojen ja osaamisen harjaannuttamiseen. Samasta asiasta käytetään monia termejä. Muita käytettyjä termejä ovat esimerkiksi viestintäkasvatus, mediapedagogiikka, medialukutaito, mediantaju ja digitaalinen lukutaito (Mediakasvatusseura 2009). Kuvassa 1 avataan mediakasvatuksen ja medialukutaidon käsitteitä sekä niiden suhdetta toisiinsa.



Kuva 1. Mediakulttuuri. (Pentikäinen, Ruhala & Niinistö 2007, 101)

Kotilaisen ja Kivikurun (1999, 13) mukaan mediakasvatuksen lähtökohdasta ollaan montaa mieltä, ja koska mediakasvatuksessa on vastakkain monia yhteiskunnan kannalta tärkeitä ja ristiriitaisia asioita, on määrittely vaikeaa. Yleisesti koetaan, että lapsia tulee suojella medialta. Sintosen (2002, 111) mielestä ainoana tehtävänä ei ole pelastaa lapsia mediasisältöjen haitallisilta vaikutuksilta, kuten väkivallalta ja pornografialta. Mediapsykologisen ajattelun mukaan ei yksi väärä sisältö tee mediatuotetta vaaralliseksi, vaan tärkeää onkin lasten mediakokemuksista keskustelu ja niiden hankkiminen yhdessä aikuisten kanssa (emt., 111). Mediakasvatusta voidaan katsoa neljästä näkökulmasta, kun tavoitteena on lasten hyvinvoinnin edistäminen. Ensimmäisenä näkökulmana on taidekasvatuksen ilmaisullinen näkökulma, joka keskittyy median sisältöihin ja omaan tuottamiseen. Toinen näkökulma on teknologiakasvatuksellinen, eli teknisten välineiden käyttö joko opetuksen välineenä tai kohteena ja tietotekniikan hyödyntämistä kasvatustyössä. Kolmas on yhteiskunta- ja kulttuurikriittisesti painottuva näkökulma, joka käsittelee median taustoja ja rakenteita sekä analysoi sisältöjen ja pohtii vaikuttavuutta. Neljäs suojelullinen näkökulma keskittyy lasten suojeluun median haitallisilta vaikutuksilta. (Mediakasvatus varhaiskasvatuksessa 2008, 7.)

Mediapedagoginen tutkimus ja kehitystyö pitävät sisällään lasten ja nuorten ja median välisen suhteen tutkimusta sekä median vaikutusta lasten ja

nuorten sosiaalistumiseen sekä mediaopetuksen arviointia mediadidaktiikan avulla. Mediapedagoginen alue tieteellisenä alueena on edelleen lastenkengissä sekä tutkimuksen että opettamisen osalta. Tavoitteena mediakasvatuksen kehityksessä tulisi olla oppilaan demokraattisen ja autonomisen mediankäytön parantaminen sekä ilmaisun että analyysin tasolla, koska tulevaisuudessa entistä tärkeämpiä taitoja tulevat olemaan itsenäisten mielipiteiden muodostus ja kriittinen valikoivuus, ja mediakasvatus parhaimmillaan kasvattaa juuri näitä taitoja. Oppilaiden valmisteleminen tulevaisuuden yhteiskunnan vaatimuksien mukaisiksi tulisi pitää sisällään elektronisen tiedon käsittelyn ja sen ymmärtämisen lisäksi heidän aikakautensa elektronisen audiovisuaalisen median sekä yleisen median kielen ymmärtämistä. (Tuftte 2003, 42-44.)

2. Lasten ja nuorten mediankäyttö

Lasten mediankäyttö huolestuttaa usein aikuisia, ja varsinkin lapset, jotka ovat yksinäisiä tai joilta puuttuu sosiaalisia kontakteja, saattavat olla vaarassa uppoutua tietokonemaailmaan. Toiset taas uskovat, että lapset oppivat ja saavat iloa median parissa. Sosiaalisesta näkökulmasta medially on mullistava vaikutus. Media voi koota lapsia yhteen, mutta se voi myös erottaa. (Matikkala & Lahikainen 2005, 92.) Sosiaalinen mediankäyttö on medioiden käyttämistä yhdessä ja käyttökokemuksista puhumista myös mediankäyttötilanteen ulkopuolella. Epäsosiaalisesti käytettynä median avulla paetaan mielikuvitusmaailmaan. (von Feilitsen 1976, Suonisen 2004, 60 mukaan.) Lahikaisen ja Matikkalan (2005, 93-95) mukaan 8–10 -vuotiaat hyödyntävät jo aktiivisesti medioita, mutta eivät vielä monipuolisesti hyödynnä mediatekniikkaa (matkapuhelimia, tietokoneita tai pelikoneita) sosiaalisessa viestinnässä, joka lisääntyy vasta teini-ikäisillä. Nuoremmilla lapsilla mediaan liittyy leikkejä ja media motivoi myös keskusteluja. Mediasisältöjen tunteminen vaikuttaa tätä kautta myös kaverialueeseen sosiaaliseen hierarkiaan sijoittumiseen. Media esiintyy lasten leikeissä sosiaalisena resurssina. (Noppari ym. 2008, 57.) Sosiaaliseen hierarkiaan sijoittumiseen tätä kautta vaikuttaa tietysti myös lapsien mahdollisuudet käyttää eri medioita ja tähän onkin vaikeampi vaikuttaa. Medialeikkejä on erilaisia, ja ne vaihtelevat poikien rajummista taisteluleikeistä tyttöjen rauhallisempiin ihmishuhteleikkeihin. Nämä mediasta omaksutut leikinaiheet voivat kuitenkin johdattaa myös tiedon hankkimiseen ja loogisen ajattelun kehittämiseen. (Noppari ym. 2008, 57-69.)

Media siis liittyy myös pienten lasten elämään vahvasti, ja kun otetaan huomioon lasten sosiaalinen ympäristö kodin ulkopuolella, voidaan rajoittamalla mediankäyttöä myös hankaloittaa lapsen sopeutumista päiväkodissa tai koulussa. Lahikainen ja Matikkala (2005, 92) pohtivatkin tutkimuksessaan,

jonka kohderyhmänä ovat tois- ja kolmasluokkalaiset, esimerkiksi paljonko ja kenen kanssa lapset käyttävät mediaa, mikä on yhteys lasten eristyneisyyden tai sosiaalisuuden ja mediankäytön välillä ja miten mediankäyttö ryhmittelee ja erottelee lapsia.

Nopparin ja muiden (2005, 62) mukaan vanhempien lasten, n. 11–14 -vuotiaiden, mediankäyttö liittyi perinteisten leikkien sijasta lasten pelaamiin tietokone- ja konsolipeleihin, joissa toimintaleikkien lisäksi ihmissuhdeleikit ovat hakeneet uuden toteutumistavan. Vanhemmat suhtautuvat huolestuneesti medialeikkeihin sisältöihin liittyen, mutta median käytön rajoittaminen vähenee kuitenkin lasten ikävuosien karttuessa. Nuoret eivät myöskään juuri puhu vanhemmilleen heitä askarruttavista mediasisällöistä ja tapahtumista, koska yhteisiä käyttökokemuksia ei ole. (Noppari ym. 2005, 62-64 .)

3. Mediakasvatus perusopetuksessa

Mediakasvatusseura yhteisöjäsenineen ja Viestinnän keskusliitto ovat ehdottaneet peruskoulun opetussuunnitelmatyöryhmälle mediakasvatuksen sisällyttämistä opetussuunnitelmaan siten, että opetussuunnitelma sisältäisi mediakasvatusta yhden viikkotunnin koko peruskoulun ajan eri oppiaineisiin integroituna, koska laajuudelliset erot mediakasvatuksen opetuksessa lisäävät alueellista epätasa-arvoa ja vähentävät kansalaisten tasavertaisuutta. Ehdotuksen tekijöiden mukaan medialukutaito kuuluu nykypäivän kansalaistaitoihin, joka tulisi taata kaikille. (Mediakasvatusseura yhteisöjäsenineen & Viestinnän keskusliitto 2009, 1-2.)

Mielipiteitä siitä, mitä mediakasvatus opetusohjelmassa sitten pitäisi sisältää, löytyy monenlaisia. Sintosen (2002, 112) mukaan, tietokoneen ja muiden mediankäyttövälineiden käyttötaitoa opettamalla lapsista ja nuorista kasvatettaisiin medialukutaitoisia yhteiskunnan kansalaisia. Mediankäyttövälineisiin voidaan lukea esimerkiksi pelikonsolit, internet, kamera, videokamera, televisio, tietokone, radio, matkapuhelin, opiskeluromput ja -pelit, virtuaaliset oppimisympäristöt ja medialelut (Mediakasvatus varhaiskasvatuksessa 2008, 15). Sintosen (2002, 112) mielestä välineorientoitunut ajattelutapa on kuitenkin riittämätön lähtökohta mediakasvatukselle vaan siihen tulisi sisältyä välineiden osalta teknisen kehityksen tuntemusta sekä erilaisten teknisten toimintaratkaisujen ymmärtämistä. Lisäksi myös mediasisältöjen kriittistä havaitsemista, tulkinnan ja analyysin taitoa, mediakielen, ilmaisun ja lajityyppien tuntemusta, kommunikaatiotaitoa, kykyä tosiasioiden erottamiseen fiktiosta sekä mediakulttuurin osalta välineiden toimintakulttuurin tuntemista (emt, 112).

Kotilaisen ja Hankalan (1999, 52) mielestä mediakasvatuksen huomioiminen on erityisen tärkeää myös opettajankoulutuksessa, koska korkeakoulujen

aineenopettajien ja luokanopettajien koulutuksessa kannetaan päävastuu tulevista mediakasvattajista. Valtaosalla opettajista ei ennen 1990-luvun puoltaväliä ole ollut mahdollisuutta tutustua mediakasvatukseen, mutta mahdollisuudet ovat lisääntyneet ja nyt mediakasvatusta on mahdollista opiskella erilaisia kokonaisuuksia (emt., 52). Tieto- ja viestintäteknikkaa hyödyntävän opetuksen oppimisprosessissa oppijoilta ja opettajilta vaaditaan ainakin kykyjä teknisten laitteiden käyttöön, tiedon tuottamiseen ja oppimateriaalien analysoimiseen. Lisäksi opettajalta vaaditaan myös verkkopedagogiikan tuntemista. (E-oppiminen 2002–2003.)

3.1. Medialukutaito ja televisio

Mediakasvatuksen sisältöjä mietittäessä on mielestäni edelleen tärkeää huomioida myös television merkitys, muiden uusien mediankäyttövälineiden lisäksi. Luke (2003, 105) toteaaakin, että vaikka 2000-luvun lapset ja nuoret ovat eläneet lapsuudestaan asti multimediaohjelmistojen ja pelikonsolien parissa, on televisio edelleen vakaa osa heidän elämäänsä. Television luetaan sosiaalisesti kasvattajaksi sen parissa vietetyn ajan takia. Television käyttöä on tutkittu lähes 50 vuotta, ja vaikka sen vaikutukset jakavatkin mielipiteitä, pidetään televisiota silti yleisesti sosiaalisen oppimisen lähteenä. Se muokkaa asenteita, sosiaalisia ja kulutustottumuksia sekä maailmankatsomusta, ja siksi televisuaalisten tekstien tutkimus on edelleen tärkeä osa medialukutaito tutkimusta, vaikka medioiden yhdistyessä multimediasovellukset tulevat entistä tärkeämmiksi mediakasvatuksessa. Narratiivisen ja teksti-kuva -analyysin periaatteet voidaankin siirtää myös uusien, samanlaista kriittisyyttä vaativien, medioiden tutkimiseen. (Luke 2003, 105-106.) Käsittelen narratiivisuutta tarkemmin alakohdassa 4.1.1.

Kurkela (2002, 93) käsittelee television soveltuvuutta mediakasvatuksen opetuskäyttöön Uutisjuttu-ohjelman kautta. Kirjoittaja käsittelee Uutisjutun syntyä, sen saamaa palautetta ja tulevaisuuden näkymiä ja tarkoittaa mediakasvatuksella medioiden yleistä tuntemusta ja niiden sisältöjen ja pelisääntöjen ymmärtämistä teknisen osaamisen sijaan. Vaikka useat ovatkin sitä mieltä, että televisiota katsotaan tarpeeksi jo kotona, koulu voisi kuitenkin antaa eväitä ja harjaannuttaa oppilaat mediakasvatuksen avulla valikoiviksi ja aktiivisiksi tv-ohjelmien käyttäjiksi. Vuorovaikutuksen mahdollisuus on tärkeää ja näkemästään pitää päästä esittämään kysymyksiä ja keskustelemaan. (emt., 93.) Uutisjutun tavoitteena oli kasvattaa lapsia monipuolisiksi mediakokijoiksi ja tavoitteena oli median viestien lukutaidon kehittämisen ja faktan ja fiktion erottamisen lisäksi myös nostaa esille median merkitystä elämänhallinnassa ja vaikuttajana valintoihin. (Kurkela 2002, 97.)

Myös koulu-tv:n osalta on kokemuksia oppimisen tukemisessa television

avulla 40 vuoden ajalta. Televisio mahdollistaa havainnollistamisen ja konkreti-soimisen, mutta lisäksi sen avulla voidaan myös avata myös uusia näkökulmia, ja osa medialukutaitoa tai mediaosaamista onkin nähdä, miten monimutkaisiin asioihin voi liittyä monia näkökulmia. Muita televisioon ja medialukutaitoon liittyviä asioita ovat omien mielikuvien jäsentäminen sekä kuvan ja sanan yhteiskäytön eli audiovisuaalisuuden ymmärtäminen. (Kurkela 2002, 95.)

3.2. Lapset ja nuoret median tekijöinä

Sosiaalinen media sekoittaa median käyttäjän ja tekijän rooleja, ja tämä tulee huomioida myös mediakasvatuksessa. Internetin kautta nuorikin voi saada mielipiteensä julki, ja esimerkiksi Uutisjutun osalta tämä mahdollistaa uusia vuorovaikutusmalleja, joissa oppilaat ja koululuokat pääsevät vaikuttamaan esimerkiksi ohjelman sisältöön (Kurkela 2002, 103).

Lapset ja nuoret perehtyvät moneen uuteen asiaan toiminnallisesti, ja luomalla omaa mediantuotantoa ja omia mediaesityksiä lapset ja nuoret voivat tutustua mediaan vuorovaikutuksellisesti. Aistihavainnot lisäävät tarvetta toimia sekä luovat pohjan oivallukselle, joka johtaa vastaavasti ymmärrykseen. Tekniikkaa vierastavia opettajia mediaesitysten sisällyttäminen opetukseen voi pelottaa, ja välineisiin tulisikin perehtyä sekä katsojan/käyttäjän että tekijän näkökulmasta. Pilkkomalla mediakasvatuksen sisällöt osakokonaisuuksiksi voidaan helpottaa kummankin osapuolen ymmärrystä. Taulukossa 1 on esitetty toiminnallisen mediaopetuksen malli. (Opetusministeriö 2007, 25-26.) Mallissa toiminnallinen mediaopetus on pilkottu kuuteen osakokonaisuuteen.

1	havainto, elämys, aistit	audiovisuaalinen elämys, mm. elokuvat elokuvateatterissa
2	kokeilu	idea työskentelystä ja ilmaisusta hahmottuu oppilaille, "ideat liikkeelle"
3	harjoittelu	kokeilussa havaitut elementit tutuiksi, mm. tekniikka, työpaja, ilmaisu
4	ilmaisu	työtapa ja ilmaisu käyttöön (dokumentointi)
5	arviointi	työskentely, prosessi, työt
6	soveltaminen	mediataitojen soveltaminen eri aineissa

Taulukko 1. Toiminnallisen Mediaopetuksen malli. (Opetusministeriö 2007, 25-26)

4. Medialukutaito

Medialukutaidolla ymmärretään yleisesti luku- ja kirjoitustaitoa eli vastaanottajan kykyä ymmärtää ja tulkita mediaviestejä ja omien mediaviestien tuotta-

mista. Nykykulttuurin multimodaalisuus, eli eri esittämisen muotojen käyttäminen rinnakkain, tekee varsinaisesta lukemisestakin monisyisempää toimintaa, tuottamista, eikä siksi yksittäistä esittämismuotoa tulisi tarkastella toisista mediamuodoista riippumattomana. (Sintonen 2002, 113-114). Välineiden käytön opettelu ja hallinta, analysoinnin taito, media-esitysten kriittinen arvioiminen ja omien viestien ja mediaesitysten tuottamisen taidot ovat mediakasvatuksen osa-alueita (Mediakasvatusseura 2009). Medialukutaito voidaan luokitella myös sen syvyyden mukaan suppeaan, toiminnalliseen, luovaan ja kriittiseen. Alimmalla eli suppealla tasolla on kysymys vain taidosta käyttää mediavälineitä. Toiminnallinen medialukutaito sisältää kyvyn ilmaisuun eli lukemiseen ja kirjoittamiseen sekä tekstien tulkintaan ja tiedonhankintaan. Median käyttäminen, soveltaminen ja arvioiminen tarpeen mukaisesti on olennaista luovalla tasolla ja kriittisen medialukutaidon omaava käyttäjä osaa erottaa tärkeän tiedon vähemmän tärkeästä, löytämään oleellisimman tiedon tietomassasta ja muodostamaan oman näkemyksensä asioista. (E-oppiminen 2002-2003.)

Lukutaidon kenttää laajentavat uudet monikanavaiset tekniikat, jotka tulevat perinteisen kirjoittamisen rinnalle, joten myös koulujen tarjoama opetus on parhaimmillaan visuaalis-auditivis-kinesteettistä monet kanavat yhdistävässä kokemuksellisesta ja elämyksellisestä kokonaisopetusta (Heinonen 2002, 109). Medialukutaito on kokonaisvaltainen lukutaito, joka voidaan jakaa digitaaliseen, teknologiseen, kulttuuriseen ja visuaaliseen lukutaitoon (Varis 2002, 13-14). Käsitellen tässä tarkemmin kulttuurista ja visuaalista lukutaitoa, koska ne ovat tärkeitä kehittämishankkeen kannalta.

4.1. Kulttuurinen lukutaito

Kulttuurinen lukutaito on kokonaisuus johon liittyvät perinne, omakohtaiset ja sukupolvien kokemukset, ympäristö sekä yhteisön sisäiset säännöt, normit ja tavat. Siihen sisältyy kyky lukea tekstejä, tulkita kuvia, ääniä ja ympäristön monia merkkejä sekä median tuntemus. (Heinonen 2002, 104). Perspektiivin eli näkökulman josta maailmaa katsomme ja tulkitsemme, määrää moni asia kuten ikä, sukupuoli, kansallisuus, kieli, perhetausta, aika, kokemukset tai koulutus (Heinonen 2002, 105). Saavuttaakseen kirjallisen lukutaidon lapsen tulee ensin oppia, kuinka tehdään kieltä koskevia havaintoja ja puhutaan kielestä. Kulttuuriseen lukutaitoon liittyy narratiivisuuden käsite ja narratiivisen perinteen siirtämisessä ja luomisessa on keskeistä kyky luoda mielikuvia. Näiden mielikuvien työstäminen tapahtuu parhaiten oman tekemisen kautta. Kasvattajan tehtävä on antaa lapselle ajattelun ja toiminnan apuvälineitä, ja näitä voidaan tarjota median avulla. (Heinonen 2002, 107.) Digitaalinen kulttuuri mahdollistaa erilaisen vuorovaikutteisen tarinankerronnan, ja moniaistinen tarinankerronta

taas avaa uusia näkökulmia ja laajentaa kokonaisuutta (Heinonen 2002, 108).

4.1.1 Narratiivisuus

Narratiivi tarkoittaa kertomusta, ja narratiivisen analyysin avulla hahmotetaan tutkittavasta kohteesta kerrottavia kertomuksia tai siitä miten tutkittavan ilmiön olemus voidaan hahmottaa kertomuksenkaltaisena rakenteena. Narratiivisen analyysin kohteena voi olla myös mediateksti, kuva, elokuva tai ympäristö, joka voidaan ymmärtää kertomuksena tai kertomuksenkaltaisena rakenteena. Analyysissa kohdetta eritellään kerrontaan ja kertomuksen teorioihin liittyvien käsitteiden, termien ja näkökulmien avulla. Analyysissa pyritään kertomuksien avulla osoittamaan ajattelu- ja toimintatapoja, merkityksiä ja asenteita, joita kertomuksen kohteeseen liittyy. (Jyväskylän yliopisto kurssi- ja oppimateriaali Koppa.)

Mediaesitys pitää usein sisällään viestin, ja median kautta viestitään kielen avulla. Median kieli on tekstiä, kuvaa, ääntä, symboleja ja näiden moninaisia yhdistelmiä. Lapsen kieli kehittyy jatkuvasti, ja jopa pienet lapset omaksuvat nopeasti erilaisia kielellisiä asioita ja näin median käyttämä kieli, kielikuvat ja käsitteet leviävät helposti ja tarttuvat myös muuhun kielenkäyttöön. Lapsille median sisällöt ovat aluksi koosteita yksittäisistä pienistä tilanteista, ja kyky seurata tarinoita kasvaa vähitellen. Esimerkiksi mainoksia lapsi ei pysty aluksi erottamaan muista ohjelmasisällöistä. Vähitellen lapsi oppii erottamaan ne erillisiksi kokonaisuuksiksi, kuitenkin lapsi ei ymmärrä piilomainontaa eikä mainosten tarkoitusta. Ajattelun kypsymisen myötä lapsi oppii erottamaan keksityt lastenohjelmien sadut sekä uutisten todellisuuden, mikä vaikuttaa mediatulkintojen lisäksi myös lapsen pelkokehitykseen. Varhaisemmassa vaiheessa pelot perustuvat usein havaittuun ja kuultuun, mutta kehityksen edetessä pelkoja ruokkii mielikuviutus ja ajattelu sekä parempi ymmärrys median sisällöistä.

Mediakasvatuksen ja mediatulkintojen näkökulmasta keskeistä on faktan ja fiktion erottaminen. Mediakasvatuksessa huomioon tulisi ottaa myös lasten ajattelun, tunne-elämän sekä persoonallisuuden kehitystehtävät ja niiden merkitys lapsen mediasuhteeseen. Mediakasvatuksessa tulisi myös varmistaa, ettei lapsen arvo- ja todellisuuskäsitys rakennu yksinomaan median pohjalta. Mediakasvatuksessa voidaan hyödyntää lasten kehitysvaiheita etsimällä niiden avulla lapsille kiinnostavia ja kehitystä tukevia mediasisältöjä. Lapsi tuntee erillisyyttä ja eroahdistusta vanhemmista, esimerkiksi näiden tunteiden hallinta on keskeistä lapsen kehityksessä. Lapsia puhuttelevia sisältöjä ovat siis hyvästelyt, yksin jääminen tai eksyminen vanhemmista. Myöhemmässä kehitysvaiheessa keskeistä on tunteiden hallinta, mediassa esiintyvät samastumiskohteet ja roolimallit. Tärkeää empatia- ja roolinottotaitojen kehittyminen on myös myöhem-

mälle moraalikehitykselle. (Mediakasvatus varhaiskasvatuksessa 2008, 12-16.)

4.2. Visuaalinen lukutaito

Kommunikointi kuvien avulla, kyky tulkita kuvallisia esityksiä sekä tuottaa niitä ovat määritelmiä visuaaliselle lukutaidolle, joka voidaan jakaa visuaaliseen ajatteluun, kommunikointiin ja oppimiseen. Visuaalisessa lukutaidossa olennaista on, että kuvan merkitys syntyy vasta aivoissa tapahtuvan prosessin tuloksena. (Kuvanlukutaito.) Visuaaliseen lukutaitoon liittyy myös kyky tulla tietoisiksi medioiden välittämistä viesteistä ja merkityksistä ja miten kuvat ja tieto muokkaavat kokonaisuuden (Kurkela 2002, 101). Kuviin liittyy esittämisen tapoja, jotka ovat muodostuneet vuosisatojen aikana, ja näiden kulttuuriin sidottujen esitystapojen ymmärtäminen ja tulkinta eivät ole itsestään selviä. (Kuvanlukutaito.) Kuvien vaikutus voi olla lisäarvoa antava tai manipuloiva. Jälkimmäistä käytetään varsinkin mainostamisessa ja nykytekniikan mukanaan tuomat mahdollisuudet kuvien käsittelyyn korostavat entisestään kriittisen kuvanlukutaidon tärkeyttä.

5. Yhteenveto

Mediakasvatuksen hyväksyminen opetussuunnitelmaan mahdollistaa tasa-arvoisen medialukutaidon kaikille. Tärkeää onkin, minkälaisena kokonaisuutena mediakasvatus opetussuunnitelmaan sisällytetään. Monipuolisuus on tärkeää sekä mediakasvatuksen näkökulmien että sisällön osalta. Mielestäni olennaista on myös mediakasvatuksen sisällyttäminen opettajakoulutukseen, sekä täydennyskoulutuksen järjestäminen jo työtään tekeville opettajille. Koululaitoksen kehittämistarpeisiin liittyvässä yhteiskunnallisessa keskustelussa on nostettu esille, että nykykoulusta puuttuu yhteisöllisyys. On siirrytty kilpailun kouluun jossa korostetaan yksilöllisyyttä, avoimuutta, yrittäjähenkisyyttä ja opetusmenetelmien moninaistamista. Koulun elämään vaikuttaa nykyisin suuresti ympäröivä kulttuuri erityisesti mediakulttuuri. (Suoranta 2003, 122.) Koska mediakulttuuri vaikuttaa yhä enemmän koulutyöskentelyyn, voidaan mediakasvatuksen tietoisemmalla haltuunotolla vaikuttaa laajemminkin koululaitoksen kehittämiseen ja lasten ja nuorten hyvinvoinnin mahdollisuuksien lisääntymiseen.

Viiteluettelo

- E-oppiminen. <http://www.yksityinenkielitoimisto.net/eoppiminen/fi6.html>. (14.12.2009)
- Heinonen, S-L. (2002). Kulttuurinen lukutaito ja narratiivinen perinne. Teok-

- nessa Sintonen, S. (toim.). *Median sylissä: Kirjoituksia lasten mediakasvatuksesta*, 104-110. Tampere: Tammer-Paino Oy.
- Jyväskylän yliopisto, Kurssi- ja oppimateriaali Koppa.
<https://webapps.jyu.fi/koppa/avoimet/hum/menetelmapolkuja/menetelmapolku/aineiston-analyysimenetelmat/narratiivinen-analyysi>. (14.12.2009)
- Jyväskylän yliopisto, Kurssi- ja oppimateriaali Koppa.
<https://webapps.jyu.fi/koppa/avoimet/hum/menetelmapolkuja/menetelmapolku/aineiston-analyysimenetelmat/narratiivinen-analyysi>. (14.12.2009)
- Kotilainen, S. & Hankala, M. (1999). Mediassa on aktiivisen oppimisen mahdollisuus. Teoksessa Kotilainen, S., Hankala, M. & Kivikuru, U. (toim.). *Mediakasvatus*, 43-70. Helsinki: Oy Edita Ab.
- Kupiainen, R., Sintonen, S. & Suoranta, J. (2007). Suomalaisen mediakasvatuksen vuosikymmenet. Teoksessa Kynäslahti, H., Kupiainen, R. & Lehtonen, M. (toim.). *Näkökulmia mediakasvatukseen*, 3-26. Helsinki: Mediakasvatusseura. <http://www.mediakasvatus.fi/publications/ISBN978-952-99964-1-4.pdf>. (14.12.2009)
- Kurkela, K. (2002). Uutisjuttu – mediakasvatusta perusopetukseen. Teoksessa Sintonen, S. (toim.). *Median sylissä: Kirjoituksia lasten mediakasvatuksesta*, 93-103. Tampere: Tammer-Paino Oy.
- Kuvanlukutaito, Viestintätieteiden yliopistoverkosto.
<http://www.uta.fi/viesverk/kuvanluku/index.php>. (10.12.2009)
- Luke, C. (2003). Critical Media and Cultural Studies in New Times. In Lavender, T., Tufte, B. & Lemish, D. (ed.). *Global Trends in Media Education – Policies and Practises*, 105-118. Creskill, NJ: Hampton Press Inc.
- Mannerheimin Lastensuojeluliitto.
<http://www.mll.fi/kasvattajille/mediakasvatus/>. (9.12.2009)
- Matikkala, U. & Lahikainen, A.R. (2005). Pelit, tietokone ja kännykkä lasten sosiaalisissa suhteissa. Teoksessa Lahikainen, A.R., Hietala, P., Inkinen, T., Kangassalo, M., Kivimäki, R. & Mäyrä F. (toim.). *Lapsuus mediamaailmassa: Näkökulmia lasten tietoyhteiskuntaan*, 92-109. Helsinki: Gaudeamus.
- Mediakasvatusseura (2009). <http://www.mediakasvatus.fi/>. (10.12.2009)
- Mediakasvatusseura yhteisöjäsenineen (2009). Medialukutaito lasten ja nuorten kansalaistaidoksi – Kannanotto peruskoulun tuntijako työryhmälle. http://mediakasvatus.fi/files/u4/ukutaito_lasten_ja_nuorten_kansalaistaidoksi.pdf. (8.12.2009).
- Mediakasvatus varhaiskasvatuksessa (2008).
http://www.kerhokeskus.fi/easydata/customers/kerhokeskus/files/liitetiedot/mediakasvatus_varhaiskasvatuksessa_verkkoversio.pdf. (13.12.2009)
- Metka Ry. <http://www.mediametka.fi/>. (8.12.2009)
- Noppiari, E., Uusitalo, N., Kupiainen, R. & Luostarinen, H. (2008). *“Mä oon nyt*

Online!” Lasten mediaympäristö muutoksessa. Tampere: Tampereen yliopisto, Tiedotusopin laitos.

- Opetusministeriö, Koulutus- ja tiedepolitiikan osasto (2007). *Ehdotus toimenpideohjelmaksi mediataitojen ja -osaamisen kehittämiseksi osana kansalais- ja tietoyhteiskuntataitojen edistämistä.* Valtioneuvosto: Yliopistopaino.
- Sintonen, S. (2002). Mediakasvatusta järjellä ja tunteella. Teoksessa Sintonen, S. (toim.). *Median sylissä: Kirjoituksia lasten mediakasvatuksesta*, 111-118. Tampere: Tammer-Paino Oy.
- Suoninen, A. (2004). *Mediakielitaidon jäljillä. Lapset ja nuoret valikoivina mediankäyttäjinä.* Helsinki: Gummerus.
- Suoranta, J. (2003). *Kasvatus mediakulttuurissa. Mitä kasvattajan tulee tietää.* Tampere: Vastapaino.
- Tufte, B. (2003). Media: The Alternative School. In Lavender, T., Tufte, B. & Lemish, D. (ed.). *Global Trends in Media Education – Policies and Practises*, 37-44. Creskill, NJ: Hampton Press Inc.
- Varis, T. (2002). Medialukutaito - johdatusta verkottuvan oppimisen medialukutaitoihin.http://www.uta.fi/~tikame/medialukutaito_varis.pdf (10.12.2009).

Ajax ja sen vaikutukset web-sovelluksen käytettävyyteen

Tommi Ritola

Tiivistelmä.

Tämä tutkielma kertoo moderneihin web-tekniikoihin kuuluvan Ajaxin peruseriaatteista sekä web-sovellusten käytettävyydestä. Ajaxin mahdollisuudet käytettävyyden alueella ovat todella merkittävät, mutta osataanko niitä hyödyntää oikealla tavalla? Web-sovellusten käytettävyys on tärkeä aihe juuri nyt, sillä teknologian kehitys on internetin puolella niin nopeaa, että käyttäjät jäävät usein unohduksiin uusia, hienoja sivustoja suunniteltaessa. Tässä tutkielmassa käydään läpi Ajaxin etuja ja haittoja käyttäen apuna aikaisemmin julkaistuja tutkimuksia ja muita materiaaleja.

Avainsanat ja -sanonnat: Ajax, web 2.0, käytettävyys

CR-luokat: C.2.1, D.2.11

1. Johdanto

Viime vuosien aikana internetin sovellukset ovat muuttaneet muotoaan hurjasti. Jo olemassa olevia tekniikoita on opittu hyödyntämään paremmin ja uusia tekniikoita web-sovellusten tekoon tulee koko ajan lisää. Web 2.0 on nykyisin muotisana ja yritykset panostavat web-sivuihinsa entistä enemmän rahaa ja aikaa.

Web 2.0:aan hyvin läheisesti liittyvä Ajax-tekniikka antaa sovellusten kehittäjille avaimet uudenlaisiin toiminnallisuuksiin sekä käytettävyyden parantamiseen. Koska web-sivuista tulee entistä monimuotoisempia, liikkuu asiakkaan ja palvelimen välillä myös enemmän dataa. Tähän on osaltaan auttanut internet-liittymien nopeuksien räjähdysmäinen kasvu, mutta toisaalta Ajax-tekniikan avulla sivuston vaatiman dataliikenteen määrä saadaan pysymään aisoissa. Tällöin saatavilla olevaa kaistaa voidaan hyödyntää muissa asioissa kuin itse sivun lataamisessa.

Uusien teknologioiden tulo ei kuitenkaan ole tarkoittanut sitä, että sivustojen käytettävyys paranee. Monet panostavat mieluummin siihen, että sivustot näyttävät hyviltä ja niissä on mahdollisimman paljon hienoja toiminnallisuuksia. Hyvän käytettävyyden myötä asiakkaat saadaan kuitenkin pysymään sivustolla ja asiakkaat pysyvät tyytyväisempinä. Onko Ajaxista siis enemmän hyötyä vai haittaa?

Tässä tutkielmassa esitellään ensin Ajax tekniikkana, sitten perehdytään tarkemmin web-sovellusten käytettävyyteen. Lopuksi käsitellään Ajaxin vaikutuksia käytettävyyteen. Tutkielman ei ole tarkoitus pureutua Ajaxin tekniikkaan kovin syvällisesti, vaan Ajax esitellään helposti ymmärrettävällä tasolla ja mietitään, miten Ajaxin käyttö vaikuttaa web-sovelluksiin.

2. Ajax

2.1 Mitä se on?

Ajax on akronyymi sanoista Asynchronous JavaScript And XML. Termin kehitti Jesse James Garrett vuonna 2005. Garretin (2005) mukaan Ajax ei ole uusi tekniikka, vaan se on jo olemassa olevien tekniikoiden hyödyntämistä uudelleen tehokkaalla tavalla. Ajaxin avulla voidaan manipuloida web-sivun sisältöä ilman, että koko sivua pitää ladata uudelleen (Petreley, 2007). Tätä taustalla tapahtuvaa ja muun toiminnan samaan aikaan mahdollistavaa tiedonhakua kutsutaan asynkroniseksi.

Asynkronisessa tiedonhaussa selain ei jää odottamaan vastausta palvelimelta, vaan jatkaa toimintaansa normaalisti mahdollistaen myös usean samanaikaisen haun tekemisen. Thau (2006, 263) kuvaa Ajaxin toimintaa kolmella kohdalla:

- käyttäjän aikaansaama tapahtuma (esimerkiksi hiiren liikuttaminen) teettää yhden tai useita kyselyitä palvelimelle saadakseen haluamaansa tietoa
- palvelin suorittaa kyselyn taustalla ja web-selain jatkaa normaalisti toimintaansa, jolloin käyttäjä voi jatkaa sivuston käyttämistä
- kyselyn tulos tulee näkyviin heti, kun palvelin on käsitellyt kyselyn (huolimatta siitä, ovatko muut kyselyt vielä valmiita), ja kyselyn tulosta käytetään yhdessä DHTML-tekniikan kanssa web-sivun sisällön päivittämiseen.

Ajax-ohjelmoinnin taustalla toimii JavaScript-komentosarjakieli, jonka kautta käytetään Document Object Model (DOM) -ohjelmointirajapintaa web-sivun sisällön muokkaamiseen. DOM on alustasta ja ohjelmointikielestä riippumaton rajapinta, joka antaa ohjelmille ja skripteille mahdollisuuden hakea ja päivittää dokumentin sisältöä, rakennetta ja tyyliä (W3C, 2005). Tekniikka itsessään on ollut käytössä jo pitkään, mutta Ajax tuo sen käyttöön lisäarvoa. Ajaxin avulla tiedon hakeminen ja laskennat voidaan suorittaa palvelimen päässä, eikä asiakaskoneella (client) (Petreley, 2007).

2.2 Mihin sitä käytetään?

Yksi Ajaxin uranuurtajista on jo vuosia ollut Google, joka on käyttänyt tekniikkaa useissa sovelluksissaan. Googlen sovellukset ovatkin mitä parhaimpia esimerkkejä kuvaamaan Ajaxin monipuolisia mahdollisuuksia web-kehityksessä.



Kuva 1. Google-hakukoneen Ajax-ominaisuus. (Google, 2009a)

Kuvassa näkyy, kuinka Googlen hakukone hyödyntää Ajaxia niin, että siitä on hyötyä myös käyttäjälle. Kun hakukenttään kirjoitetaan osa hakusanasta, hakukone tekee taustalla Ajax-kyselyitä ja ehdottaa kyselyn tulosten perusteella käyttäjälle eri hakuvaihtoehtoja. Sivu ei kuitenkaan hakujen aikana lataudu joka kerta uudelleen, vaan päivittäminen tapahtuu vain kenttään, jossa ehdotetut tulokset ovat. Tämä on hyvä esimerkki siitä, miten yksinkertaisilla asioilla sivuston käytettävyyttä voidaan parantaa Ajaxin avulla.

Toinen – hieman monipuolisempi ja kehittyneempi – esimerkki Ajaxin käytöstä on Googlen kalenteri (Google Calendar), joka on täysin web-selaimessa toimiva kalenterisovellus. Sovelluksen voisi sanoa muistuttavan ulkonäöltään ja toiminnallisuuksiltaan hyvin paljon työpöytäsovellusta, mutta mitään tietokoneelle asennettavaa ohjelmaa ei web-selaimen lisäksi tarvita. Nämä toiminnallisuudet on saatu tehtyä nimenomaan Ajaxia hyväksi käyttäen, ja kun

sovelluksen toimintaa tarkastelee lähemmin, huomaa, että Ajax-kyselyitä tehdään todella paljon käyttäjän navigoidessa sivustolla.

Nämä kaksi esimerkkiä näyttävät, että Ajaxin avulla voidaan tehdä todella monipuolisia web-sovelluksia. Tämä aiheuttaa myös sen, että sovelluksista tulee monimutkaisempia myös ohjelmoijan näkökulmasta. Toisaalta, kuten Google-hakukoneen esimerkistä huomaa, voi Ajaxia hyödyntää myös yksinkertaisemmissä konteksteissa.

2.3 Tekninen toteutus

Ajaxin käytössä toteutuksellisesti ehkä tärkeimpänä asiana pidetään request-oliota, jonka avulla kyselyiden tekeminen Ajax-tyylillä on mahdollista. Thau (2006, 265) jakaa olion käytön neljään kohtaan:

- olio luodaan
- oliolle kerrotaan, mihin kysely lähetetään
- oliolle kerrotaan, mitä tuloksen saamisen jälkeen tulisi tehdä
- suoritetaan kysely.

W3C:n XMLHttpRequest-dokumentaatioissa (2009) olion käyttämisestä on kuvattu muun muassa koodiesimerkillä 1.

```
function fetchStatus(address) {
    var client = new XMLHttpRequest();
    client.onreadystatechange = function() {
        // in case of network errors this might not give reliable results
        if(this.readyState == 4)
            returnStatus(this.status);
    }
    client.open("HEAD", address);
    client.send();
}
```

Koodiesimerkki 1. Request-olion luominen yksinkertaisimmillaan. (W3C, 2009)

Koodiesimerkin 1 fetchStatus()-funktio tarkistaa sille parametrina annetun dokumentin tilan ja palauttaa sen. Yllä oleva esimerkki on kaikessa yksinkertaisuudessaan hyvä, jos halutaan demonstroida olion käyttöä yleisellä tasolla, mutta siinä ei ole otettu huomioon selaimien erilaista request-olion käsittelyä. Koodiesimerkissä 1 olio on luotu client-muuttujaan kutsumalla

XMLHttpRequest-luokan rakenninta. Kuitenkaan esimerkiksi kaikki Internet Explorer -selaimet eivät kyseistä rakenninta löydä. Thau (2006, 266) esimerkissä olion luominen on kuvattu koodiesimerkissä 2 kuvatulla tavalla.

```
var request = null;
if (window.XMLHttpRequest) {
    request = new XMLHttpRequest();
} else if (window.ActiveXObject) {
    request = new ActiveXObject("Microsoft.XMLHTTP");
}
```

Koodiesimerkki 2. Request-olion luominen myös IE:lle. (Thau, 2006)

Koodiesimerkissä 2 otetaan huomioon myös Internet Explorer -selaimet, jotka käyttävät request-oliosta nimitystä ActiveXObject. Tällä tavalla saadaan mukaan tuki useammalle selaimelle. Ajax-ohjelmoinnin tueksi on olemassa myös useita valmiiksi luotuja kirjastoja, joihin olion luominen eri selaimille on toteutettu niin, ettei ohjelmoijan tarvitse siitä huolehtia erikseen.

2.4 Tekniset edut ja haitat

Ajaxin yksi suurista eduista verrattuna esimerkiksi Flashiin ja Javaan on, ettei se tarvitse erillisiä selaimen ladattavia liitännäisiä (Mäki, 2008). Tämä taas vähentää käyttäjältä vaadittavia toimia sivuston toimimaan saattamiseksi. Nykyään uusimmat selaimet tukevat Ajaxissa käytettäviä tekniikoita lähes poikkeuksetta, kunhan otetaan huomioon selaimien erilaiset toteutustavat.

Ajaxin avulla web-sivustoille saadaan käyttäjäkokemusta rikastavia sekä käytettävyyttä parantavia ominaisuuksia ilman, että tarvitsee turvautua tekniikoihin, joita selain ei välttämättä oletuksena tue. Ajaxin avulla sivustoille saadaan suhteellisen helposti myös "ammattimaista otetta" ja sivusto saadaan erottumaan paremmin massasta. Voikin sanoa, että Ajaxin avulla web-sovelukseen saadaan mukaan samankaltaisia ominaisuuksia, joita työpöytäsovelluksissa on totuttu näkemään (Garrett, 2005).

Ajaxin käyttö ei siltikään ole täysin ongelmaton, kuten request-olion luomisessakin huomattiin. Mäen (2008) mukaan Ajaxin riippuvuus JavaScriptistä voidaan nähdä myös ongelmana, koska selain ei välttämättä salli JavaScriptin käyttöä. Tämä saattaa johtua esimerkiksi siitä, että käyttäjä on itse kytkenyt JavaScriptin pois päältä.

3. Käytettävyys web-sovelluksessa

3.1 Yleistä

Whiteheadin (2006) mukaan "käytettävyys on käyttäjä- ja tehtäväriippuvaista ja merkittävää on, miten hyvin käyttäjä suoriutuu tekemistään tehtävistä, miten tehokkaasti käyttäjä tämän tekee ja kuinka tyytyväinen käyttäjä on tehtävän aikana ja sen jälkeen". Hän viittaa myös Nielsenin käytettävyyskriteereihin, joita ovat

- opittavuus
- muistettavuus
- tehokkuus
- luotettavuus
- käyttäjien tyytyväisyys.

Web-sivun tulisi siis olla helposti opittava, eikä käyttäjällä saisi mennä kauaa toiminnallisuuden muistamiseen pienen tauonkaan jälkeen. Käytön tulisi olla myös tehokasta niin, että käyttäjän tekemän työn määrä olisi mahdollisimman pieni. Lisäksi sovelluksen tulisi toimia oikein ja sen tulisi opastaa käyttäjää tekemään tehtävät oikealla tavalla. Yhtenä mittarina voidaan pitää myös käyttäjän tyytyväisyyttä sivustoon. Nämä eivät kuitenkaan ole ainoita käytettävyyden mittareita, vaan tässä on esiteltyä vain keskeisimmät ja tunnetuimmat käytettävyystekijät.

Jos asiakas ei löydä sivuilta etsimäänsä tuotetta, hän ei voi ostaa sitä. Jos tärkeät tiedot on piilotettu, ostamispäätöksen mahdollisuus heikentyy. Esimerkiksi luottokortin tietoja annettaessa tehdyt virheet voivat vaikuttaa myyntiin negatiivisesti. Huonosti toteutetut sivustot myös ruuhkauttavat teknisen tuen ja näin ollen aikaansaavat suuria lisäkustannuksia. (Constantine and Lockwood, 2002).

3.2 Web 2.0

Internetin sisältö on viime vuosina muuttunut huimasti. Web-sivut tulevat koko ajan monipuolisemmiksi ja yhä useampi sovellus toimii ainakin jossain määrin verkossa. Web 2.0 on suurta bisnestä ja yritykset myös käyttävät siihen paljon rahaa.

Web 2.0:lla tarkoitetaan internetin uutta suuntausta, ei niinkään uusia tekniikoita. Esimerkkejä web 2.0 -sovelluksista ovat muun muassa blogit, wikit, Google Maps, Facebook ja Myspace (Pilgrim, 2008). Nielsenin (2007) mukaan web 2.0:n tuomien mahdollisuuksien mukana on tullut kuitenkin myös ongelmia, sillä käytettävyyteen ja käyttäjiin ei enää välttämättä kiinnitetä niin paljon

huomiota. Nielsenin mielestä ennen yhtäkään koodiriviä tulisi suunnittelu tehdä huolella; hyvään sivustoon kuuluu muun muassa helppokäyttöisyys, hyvät hakutoiminnot, selkokieliyys sekä käytettävyytestaus (Nielsen, 2007).

3.3 Ajaxin hyötyjä ja haittoja

Ajax liittyy hyvin läheisesti web 2.0:aan ja sen käyttö on yleistynyt huimasti viimeisten vuosien aikana. Ajaxin käytöllä tiedetään olevan paljon positiivisia vaikutuksia käytettävyyteen, mutta myös sen haittapuolista kannattaa olla tietoinen web-sivustoa suunnitellessa.

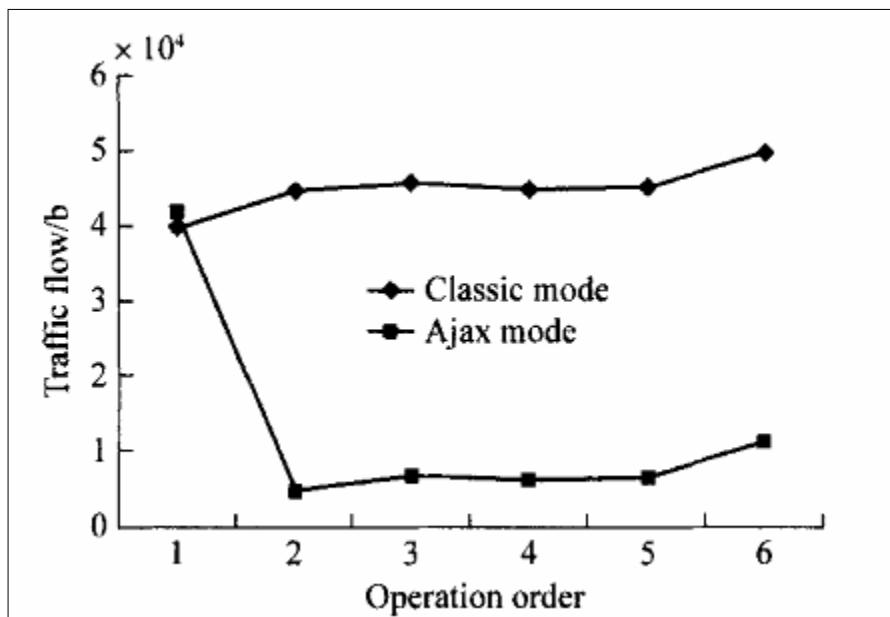
Suurin Ajaxin tuoma etu web-sivun käytettävyyteen lienee se, että koko sivua ei tarvitse ladata uudelleen, jos halutaan päivittää tiettyä kohtaa sivusta. Tämä vaikuttaa käytettävyyteen positiivisesti useammallakin tavalla. Sivustolla tapahtuvan tiedonsiirron määrä pienenee huomattavasti ja varsinkin hitailla yhteyksillä tapahtuvaa ruudun "välkähtämistä" sivun latautuessa ei tarvitse katsella. Lisäksi sivuilla navigointi on jouhevampaa ilman turhaa odottelua.

Liao ja kumppanit (2007) ovat tutkineet tarkemmin Ajaxin vaikutusta web-sovelluksen suorituskykyyn. Heidän mukaansa web-sivun suorituskykyyn lasketaan mukaan muun muassa tietovirran määrä, vastauksessa ilmenevä viive sekä latausajat. Tutkimuksessaan he käyttivät erillistä ohjelmaa, jolla suorituskykyä mitattiin.

Mode	Classic/b	Ajax/b
Average size of last five pages	45 879	6 936
Total size of six pages	674 602	480 646
Average size of six pages	112 433	80 107
Total savings		28.7%

Kuva 2. Ajaxin ja perinteisen tavan väliset erot tiedonsiirtomäärissä. (Liao *et al.*, 2007)

Kuvan 2 taulukossa nähdään tutkimuksessa ilmenneet erot tiedonsiirtomäärissä perinteisemmän "web 1.0" -tyylin ja Ajax-tekniikan välillä. Tutkimuksessa testattiin kuutta eri sivua ja verrattiin niiden tuloksia keskenään. Huomattavaa on, että keskimääräinen sivukoko (average size of six pages) oli Ajax-tekniikkaa käytettäessä reilusti pienempi. Tämä tulos selittyy ainoastaan sillä, että perinteisellä tyyllillä koko sivu ladataan uudelleen, jos halutaan muuttaa sivun sisältöä.



Kuva 3. Yhteenlaskettu tiedon määrä perinteisellä- ja Ajax-tekniikalla. (Liao *et al.*, 2007)

Kuva 3 osoittaa selvästi Ajax-tekniikan paremmuuden, kun vertailukohteenä on siirrettävän datan määrä. Kuvan perusteella Ajax-tekniikan (Ajax mode) avulla saatetaan päästä jopa neljäsosaan siitä datan määrästä, jonka perinteinen tekniikka (Classic mode) vaatii. Tutkimuksessa todettiin, että Ajaxin avulla verkon resurssit tulevat paremmin käyttöön ja datan liikuttelu on sulavampaa ja vakaampaa.

Vaikka Ajaxin hyödyt ovat selkeät, on myös useita Ajaxin käyttöä vastustavia asiantuntijoita. Yksi heistä – ehkä kuuluisin – on Jakob Nielsen. Käytettävyyden pioneeri Nielsen on tuonut esiin useita ongelmia, joita Ajaxin käyttöön liittyy. Nielsenin (2005) mukaan Ajaxin myötä webin kehittäjän, Tim Berners-Leen, alkuperäinen ajatus webistä murtuu.

Alkuperäisen ajatuksen mukaan sivun näkymä vastaa sivun navigaatiota. Toisin sanoen sivun näkymä on aina sama. Ajaxissa näkymää voidaan manipuloida ilman, että mennään sivulta pois. Jos käyttäjä esimerkiksi jättää Ajax-pohjaiselle sivulle kirjanmerkin ja yrittää palata sen avulla takaisin näkymään, joka oli kirjanmerkkiä jätettäessä, se ei välttämättä onnistu. Tämä johtuu siitä, että kirjanmerkkiin tallentuu vain sivun osoite, eikä tallentamishetkellä olevaa sivun näkymää. Näkymä sivun ”perustilasta” saattaa olla muuttunut, mikäli on tehty Ajax-kyselyitä. Tämä johtaa myös siihen, että sivujen suorat osoitteet (URL) eivät aina toimi. Sivun osoite viittaa aina tiettyyn sivuun, ei sivun näkymään. (Nielsen, 2005)

Nielsenin (2005) mukaan Ajaxin kanssa on ongelmia myös esimerkiksi tulostettaessa; kaikki selaimet eivät osaa tulostaa Ajax-sivuja oikein. Myöskään

hakukoneet eivät osaa aina käsitellä Ajax-sivuja, koska ne eivät ota huomioon sivun tilaa. Nielsen kertoo myös, että jos sivustolla on tarjolla sekä Ajax-tyylinen että perinteinen sivu, käyttäjät yleisemmin valitsevat perinteisen toteutustavan sivun.

4. Yhteenveto

Tutkielmassa keskityttiin web 2.0:n yhteen tärkeään tekniikkaan, Ajaxiin. Ajaxin hyödyt ovat melko laajasti tiedossa, mutta on myös muistettava, että asioilla on aina kääntöpuolensa. Tietotekniikassa asiat tapahtuvat nopeasti ja uusia tekniikoita tulee jatkuvasti lisää. Tämä saattaa usein johtaa siihen, että tekniikoita käytetään vain sen takia, että ne ovat olemassa.

Tutkielmassa tuli esille Ajaxin kiistämättä hyvät vaikutukset käytettävyyteen. On tärkeää, että kasvavat web-sovellukset saadaan toimimaan järkevästi niin, etteivät siirrettävät datamäärät räjähtäisi käsiin. Toisaalta mikäli tekniikkaa käytetään väärin, voi siitä olla myös paljon haittaa. Olennaista Ajaxin käytössä on, että sovelluksen kehittäjä tietää, mitä on tekemässä. Ajaxia ei tule käyttää sen takia, että se on yksi vaihtoehto, tai siksi, että se tekisi sivustosta teknisesti hienon. Ajaxia tulee käyttää vain silloin, kun siitä oikeasti on käyttäjälle enemmän hyötyä kuin haittaa.

Viiteluettelo

- Constantine, L.L. & Lockwood, L.A.D. (2002). Usage-centered engineering for web applications, *IEEE Software* 19, 2, 42 – 50.
- Garrett, J. J. (2005). *Ajax: A New Approach to Web Applications*, from <http://www.adaptivepath.com/ideas/essays/archives/000385.php>.
- Google (2009), <http://www.google.com>.
- Google Calendar (2009), <http://www.google.com/calendar>.
- Liao, W.L., Liu, F. & Yang, J. (2007). The impact of Ajax on network performance, *The Journal of China Universities of Posts and Telecommunications*, 14, 32 – 34.
- Mäki, S. (2008). *Web 2.0- verkkosovellusten piirteet ja Ajax niiden toteutuksessa*. Pro gradu -tutkielma. Tampereen yliopisto, Tietojenkäsittelytieteiden laitos.
- Nielsen, J. (2005). *Why Ajax sucks (most of the time)*, from <http://www.usabilityviews.com/ajaxsucks.html>.
- Nielsen J. (2007). *Web 2.0 "neglecting good design"*, from <http://news.bbc.co.uk/2/hi/technology/6653119.stm>.
- Petreley, N. (2007). Ajax simplified, *Linux Journal*, 157, 1.

- Pilgrim, C. J. (2008). Improving the usability of web 2.0 applications, *Proceedings of the 18th International Conference on World Wide Web*, 821 – 830. New York: ACM Press.
- Thau, D. (2006). *Book of JavaScript: A Practical Guide to Interactive Web Pages (2nd Edition)*, San Fransisco: No Starch Press.
- Whitehead, C. C. (2006). Evaluating web page and web site usability, *ACM Southeast Regional Conference: Proceedings of the 44th Annual Southeast Regional Conference*, 788 – 789.
- W3C Document Object Model, W3.org (2005), from <http://www.w3.org/dom>.
- W3C XMLHttpRequest, W3.org (2009), from <http://www.w3.org/TR/2009/WD-XMLHttpRequest-20091119/>.

Massiivisesti hajautetut avain-arvo -tietokannat

Timo Sirainen

Tiivistelmä.

Hajautetut avain-arvo -tietokannat jotka skaalautuvat tuhansiin palvelimiin pohjautuvat usein tekniikoihin, joita Amazon Dynamo -tietokanta käyttää. Tässä tutkielmassa kuvataan Dynamon toiminnan tärkeimmät perustekniikat.

Avainsanat ja -sanonnat: Hajautetut tietokannat, Amazon Dynamo.

CR-luokat: H.2.4

1. Johdanto

Yhä useammat palvelut tarvitsevat massiivisia tietokantoja, jotka skaalautuvat helposti palvelun suosion kasvaessa. Perinteisesti tietokantoja on skaalattu vertikaalisesti, eli ostamalla nopeampi ja kalliimpi palvelin. Jos tietokannalla on paljon lukijoita, voidaan sitä skaalata myös replikoimalla tietokanta usealle palvelimelle, jotka hoitavat lukija-kyselyitä. Tämä ei kuitenkaan auta loputtomasti, jos pullonkaulana on kirjoitusnopeus.

Tietokantoja voidaan myös skaalata horisontaalisesti siruttamalla (sharding), eli hajuttamalla tietokannan taulun rivit usealle eri palvelimille. Tällöin kirjoitusnopeuttakin voidaan kasvattaa lisäämällä palvelimia. Siruttaminen on kuitenkin hankalaa toteuttaa perinteisillä tietokannoilla, koska asiakasohjelmien täytyy sisäisesti toteuttaa logiikka, jolla rivit jaetaan palvelimille.

Eric Brewerin [2002] CAP-teoreeman mukaan hajautetut tietokannat voivat toteuttaa korkeintaan kaksi ominaisuutta seuraavasta kolmesta: yhdenmukaisuus (consistency), tavoitettavuus (availability) ja verkon partitioiden suvaitsevaisuus (tolerance to network partitions). Jos halutaan, että tietokanta on hajautettu eri verkkoihin ja kaikki partitiot ovat aina tavoitettavissa, se ei voi olla myös yhdenmukainen. Perinteiset ACID-tietokannat ovat keskittyneet enemmän pitämään tietokannan yhdenmukaisena kuin pitämään sen tavoitettavana.

BASE (basically available, soft-state, eventual consistency) -tietokannat eivät täytä yhdenmukaisuuden vaatimusta, mutta ovat aina tavoitettavissa [Pritchett, 2008]. Tietokannasta saatu tieto ei välttämättä ole aina täysin ajantasainen, mutta se tieto, mikä on saatavilla, on aina luettavissa, ja tietoa voi aina päivittää. Lopullinen yhdenmukaisuus (eventual consistency) kuitenkin tarkoittaa, että tietokanta yhdenmukaistuu tietyn ajan kuluessa, kunhan kaikki verkkoyhteydet ovat kunnossa [Vogels, 2008]. BASE-tietokannat skaalautuvat yleensä helposti sadoille tai tuhansille palvelimille.

2. Algoritmit

2.1. Vektorikellot

Hajautetuissa järjestelmissä on usein tärkeää tietää tapahtumien järjestys. Yksi ratkaisu olisi käyttää keskitettyä aikapalvelinta, joka antaisi jokaiselle tapahtumalle yksilöllisen aikaleiman. Tämä on kuitenkin yleensä epäkäytännöllinen ratkaisu, koska se lisää ylimääräistä viestintää ja vaatii, että aikapalvelin on aina kaikkien tavoitettavissa.

Lamport [1978] kehitti menetelmän, jolla tapahtumat voidaan järjestää käyttäen loogisia kelloja ja ilman ylimääräistä viestintää. Jokaisella prosessilla on oma looginen kello, jota päivitetään seuraavasti:

- Jokainen sisäinen tapahtuma kasvattaa prosessin kellon arvoa.
- Kellon arvo lähetetään jokaisen ulospäin menevän viestin mukana.
- Kun sisäänpäin tuleva viesti vastaanotetaan, päivitetään kellon arvo suuremmaksi kuin sen nykyisen arvon ja viestissä olevan arvon maksimi.

Lamportin kellolla voidaan siis päätellä, että jos tapahtuma a tapahtui ennen tapahtumaa b , niin $C(a) < C(b)$, jossa $C(x)$ on tapahtuman x looginen kello. Päinvastoin ei kuitenkaan voida päätellä, eli jos $C(a) < C(b)$, saattoi a tapahtua ennen b :tä, mutta se saattoi tapahtua myös samanaikaisesti. Vektorikellojen avulla voidaan erottaa nämä tapaukset.

Fidge [1988] ja Mattern [1988] kehittivät vektorikellot samanaikaisesti. Vektorikello on lista loogisia kelloja, yksi jokaiselle järjestelmän prosessille. Vektorikelloa päivitetään seuraavasti:

- Aluksi jokaisen loogisen kellon arvo on nolla.
- Jokainen sisäinen tapahtuma kasvattaa prosessin oman loogisen kellon arvoa.
- Kaikkien loogisten kellojen arvot lähetetään jokaisen ulospäin menevän viestin mukana.
- Kun sisäänpäin tuleva viesti vastaanotetaan, päivitetään loogiset kellot. Jokaisen loogisen kellon uusi arvo on sen nykyisen arvon ja viestissä olevan arvon maksimi.

Vektorikelloja u ja v voidaan vertailla:

- $u \leq v$, jos ja vain jos kellon u jokaisen komponentin arvo on korkeintaan kellon v vastaavan komponentin arvon suuruinen.
- $u < v$, jos ja vain jos $u \leq v$ ja $u \neq v$.
- $u \parallel v$, jos ja vain jos $\neg(u < v)$ ja $\neg(v < u)$, jossa \parallel tarkoittaa että vektorikellot ovat toisistaan riippumattomat.

$VC(x)$ on tapahtuman x vektorikello. Jos tapahtuma a tapahtui ennen tapahtumaa b , niin $VC(a) < VC(b)$. Voidaan myös päätellä päinvastoin, eli jos $VC(a) < VC(b)$, tapahtui a ennen b :tä. Jos $VC(a) \parallel VC(b)$, tapahtumat a ja b tapahtuivat samanaikaisesti.

2.2. Merkle-puut

2.2.1. Lamport-allekirjoitus

Lamport [1979] kehitti digitaalisen allekirjoitusmenetelmän, jonka turvallisuus perustuu kryptografisesti turvalliseen yksisuuntaiseen funktioon. Funktion voi toteuttaa esimerkiksi symmetrisellä salausalgoritmillä, kuten AES tai DES, käyttämällä aina samaa salattavaa tekstiä. Tällöin jos salausalgoritmi on muotoa $S_{avain}(\text{selkoteesti}) \rightarrow \text{salattuteksti}$, voidaan yksisuuntainen funktio $F(x) \rightarrow y$ toteuttaa esimerkiksi $S_x(0) \rightarrow y$.

Viestin allekirjoitus toteutetaan yksi bitti kerrallaan. Jokaista bittiä varten generoidaan kaksi salaista x :n arvoa: $x[0]$ ja $x[1]$. Näistä generoidaan julkiset y :n arvot: $y[0] = F(x[0])$ ja $y[1] = F(x[1])$. Jos allekirjoitettavan bitin arvo on 0, tehdään $x[0]$ julkiseksi, jos bitin arvo taas on 1, julkaistaan $x[1]$. Allekirjoitus voidaan varmistaa tarkastamalla, että $F(x)$ palauttaa oikean aiemmin julkaistun y :n. Jos halutaan allekirjoittaa n bittiä pitkä viesti, tarvitaan siis $2n$ julkista y :tä. Julkisten y :iden määrän pienentämiseen kehitettiin myös useita menetelmiä.

Alla on yksinkertainen esimerkki, kuinka henkilö A allekirjoittaa bittijonon, ja henkilö B varmistaa sen oikeellisuuden. Käytännössä kuitenkin käytettäisiin suurempia x :n ja y :n arvoja.

1. A generoi satunnaisia $x[0]$ ja $x[1]$ arvoja: (132, 28), (123, 20), (203, 17), (3, 232), ...
2. Generoidut x :n arvot ajetaan salausalgoritmin läpi, jolloin saadaan julkiset $y[0]$ ja $y[1]$ arvot: (37, 135), (51, 82), (17, 95), (218, 48), ...
3. A julkaisee kaikki y :n arvot ja B vastaanottaa ne.
4. A allekirjoittaa bittijonon 0110 julkaisemalla tarvittavat x :n arvot: 132, 20, 17, 3. Näitä x :n arvoja ei enää koskaan käytetä allekirjoittamaan muita bittijonoja.
5. B vastaanottaa bittijonon 0110 ja julkaistut x :n arvot.
6. B tarkistaa, että ajamalla julkaistut x :n arvot salausalgoritmin saadaan odotetut y :n arvot.

2.2.2. Merkle-puut

Lamport-allekirjoitukset ovat epäkäytännöllisiä, koska jokaista allekirjoitettavaa viestiä varten tarvitaan kokonaan uudet julkaistut y :t. Merkle [1998] kehitti

menetelmän, jolla Lamport-allekirjoituksia voidaan käyttää ainoastaan yhdellä julkaistulla y:tä vastaavalla arvolla.

Merkle-puu on ääretön puu kertakäyttöisiä allekirjoituksia. Puu voi olla monihaarainen, mutta yksinkertaisimmillaan se on binääripuu. Binääripuussa jokaisessa solmussa on kolme allekirjoitusta: vasen, oikea ja viesti. Vasen allekirjoittaa vasemman lapsisolmun, oikea allekirjoittaa oikean lapsisolmun, ja viesti on varattu varsinaisen viestin allekirjoittamiseen.

Puun solmut numeroidaan siten, että puun juuri on 1, solmun i vasen lapsi on $2i$ ja oikea lapsi on $2i+1$. Allekirjoitusten x arvoihin viitataan kolmiulotteisella taulukolla $x[\langle \text{solmun numero} \rangle, \langle \text{vasen, oikea tai viesti} \rangle, \langle \text{indeksi kertakäyttöallekirjoituksessa} \rangle]$. Solmun i allekirjoituksen j kaikkiin indekseihin viitataan käyttämällä merkintää $x[i,j,*]$, ja vastaavasti y :lle merkintää $y[i,j,*]$. Kaikki y :n arvot luodaan funktion F avulla: $y[i,j,k] = F(x[i,j,k])$. Määritellään myös hajautusfunktio $\text{HASH}(i) = (F(y[i,\text{vasen},*]), F(y[i,\text{oikea},*]), F(y[i,\text{viesti},*]))$. Ainoa julkaistava arvo, jota tarvitaan allekirjoitusten varmistamiseen, on $\text{HASH}(1)$.

Puun koko on ääretön. Ei kuitenkaan ole käytännöllistä tallentaa ääretöntä x :n taulukkoa, joten sen arvot täytyy pystyä generoimaan tarpeen mukaan. Yksi tapa toteuttaa tämä on käyttää salausfunktiota S : $x[i,j,k] = S_{\text{salainen avain}}(\langle i,j,k \rangle)$.

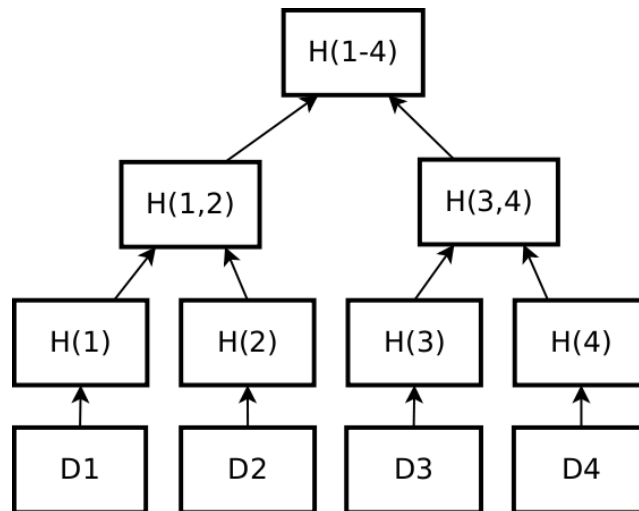
Kun tarvitaan uusi allekirjoitus, allekirjoittaja valitsee seuraavan käyttämättömän solmun i puusta, ja allekirjoittaa viestin julkaisemalla i :n, $y[i,\text{viesti},*]$ arvot sekä tarvittavan x :n osajoukon taulukosta $x[i,\text{viesti},*]$. Allekirjoittaja julkaisee myös arvot $F(y[i,\text{vasen},*])$ ja $F(y[i,\text{oikea},*])$, jonka jälkeen vastaanottaja voi laskea arvon $\text{HASH}(i)$:n. Jos i on 1, varmistetaan että julkinen $\text{HASH}(1)$ täsmää laskettuun arvoon. Jos i ei ole 1, jatketaan algoritmia allekirjoittamalla $\text{HASH}(i)$ sen vanhemmalla solmulla $i/2$, mutta sen sijaan, että käytettäisiin viesti-allekirjoitusta, käytetään vasen- tai oikea -allekirjoitusta, riippuen siitä, onko i parillinen vai pariton. Tätä jatketaan, kunnes i on 1. Kokonainen allekirjoitus vaatii siis $\log_2 i$ kertakäyttöallekirjoituksen lähettämistä.

2.2.3. Tiivistepuut

Merkle-puita on myöhemmin käytetty kuvaamaan yleisesti puurakennetta, jossa puun solmu allekirjoittaa tai tiivistää sen lapsisolmut. Tällaisia Merkle-puita kutsutaan myös tiivistepuiksi (hash tree).

Kuvassa 1 on esimerkki tiivistepuusta. Puun lehdet tiivistävät tietolohkojen sisällöt, ja muut solmut tiivistävät lapsisolmujen tiivisteet. Oletetaan, että samat tietolohkot halutaan säilyttää toisella palvelimella, mutta molemmat palvelimet voivat muuttaa niitä. Jos molemmat palvelimet säilyttävät tiivistepuuta tietolohkoista, on muuttuneiden tietolohkojen etsintä nopeaa: jos puun juuren tiiviste on sama molemmissa puissa, mitään ei ole muuttunut. Muussa tapauksessa edetään lapsisolmuun, kunnes löydetään sama tiiviste tai päädytään lehteen,

jolloin tiedetään että tietolohko on muuttunut. Tällä tavoin käydään koko puu läpi.



Kuva 1. Esimerkki tiivistepuusta, jossa tietolohkoista D1..D4 tallennetaan tiivisteet puun lehtiin. Puun muiden solmujen arvona on tiiviste lapsisolmujen tiivisteistä.

3. Amazon Dynamo

Amazon Dynamo [DeCandia et al., 2007] on yksi ensimmäisiä BASE-tietokantoja. Amazon tarvitsi tietokannan, joka skaalautuisi tuhansille palvelimille, jotka sijaitisivat useissa konesaleissa ympäri maailman. Tietokannan pitäisi pystyä käsittelemään virhetilanteet automaattisesti ja ilman tiedon häviämistä, jotta asiakkaat eivät turhautuisi ja ostaisi ostoksiaan muualta.

Dynamo suunniteltiin olettamaan, että virhetilanteita tapahtuu jatkuvasti, ja että ne pitää pystyä korjaamaan nopeasti ja helposti. Palvelimia voidaan lisätä ja poistaa Dynamosta ilman monimutkaisia toimenpiteitä, Dynamo hoitaa kaiken automaattisesti. Dynamo vaatii todella vähän varsinaista ylläpitoa. Tärkeimpinä perusajatuksina Dynamon suunnittelussa olivat:

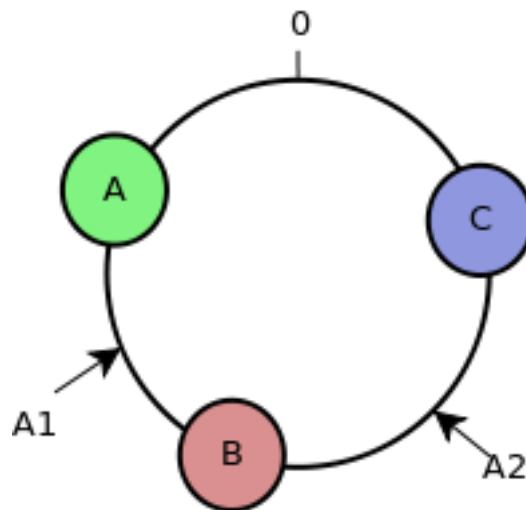
- Skaalautuvuus: Palvelimien lukumäärä pitää voida nopeasti kasvattaa moninkertaiseksi.
- Symmetrisyys: Kaikki palvelimet ajavat samaa ohjelmistoa ja hoitavat samoja asioita. Ei siis ole erikoisia palvelimia, jotka tietävät tietokannan rakenteesta enemmän kuin toiset.
- Hajautuvuus: Palvelimet voivat sijaita ympäri maailman eri konesaleissa.
- Heterogeenisyys: Palvelimet voivat olla erilaisia, joten niillä voi olla suuriakin eroja muistin ja levytilan määrässä. Dynamon täytyy osata antaa palvelimille sopiva määrä töitä niiden kapasiteetin mukaan.

Dynamo on avain-arvo -tietokanta, eli se tukee tiedon hakemista, tallentamista ja poistamista ainoastaan primääriavaimen perusteella. Avaimelle voi asettaa vain yhden arvon, eli tietokanta ei tue sarakkeita. Tämä voi vaikuttaa rajoittuneelle, mutta se riittää moniin eri käyttötarkoituksiin.

Dynamosta on tehty kaksi suosittua kloonia, jotka ovat julkaistu avoimella lähdekoodilla: Dynamite ja Project Voldemort. Myös Cassandra-tietokanta mainostaa yhdistäneensä parhaat osat Dynamosta sekä Googlen BigTable-tietokannasta.

3.1. Tiedon hajauttaminen

Dynamo hajauttaa alkiot eri palvelimille käyttäen yhtenäistä hajautusta (consistent hashing). Yhtenäinen hajautus -menetelmällä voidaan toteuttaa hajautustaulu, jonka kokoa voi muuttaa ilman että kaikki alkiot täytyy hajauttaa uudelleen [Karger et al., 1997]. Tämä on tärkeää, kun toteutetaan usean palvelimen hajautettu hajautustaulukko (distributed hash table). Normaalilla hajautusfunktiolla palvelimien lisäys tai poisto vaatisi lähes kaikkien alkioiden siirtämistä toiselle palvelimelle, joka olisi erittäin raskas operaatio.



Kuva 2. Avain A2 kuuluu ensisijaisesti palvelimelle B, ja avain A1 palvelimelle A.

Yhtenäisessä hajautuksessa hajautusfunktion antamat arvot voidaan ajatella sijoittuvan ympyrän kehälle kuvan 2 mukaisesti. Jokainen palvelin sijoitetaan satunnaiselle pisteelle kehällä (kuvassa A, B ja C). Kun halutaan löytää palvelin, joka on vastuussa avaimesta, ajetaan hajautusfunktio avaimelle, jolloin löydetään sen aloituspiste kehällä. Tästä pisteestä lähdetään kulkemaan myötäpäivään, kunnes löydetään palvelin. Jos palvelin on kaatunut, jatketaan matkaa, kunnes löydetään seuraava toimiva palvelin.

Palvelimien lisäys ja poisto vaikuttaa vain osaan kehää, joten suurin osa alkiosta pysyy paikoillaan. Myös mitä enemmän palvelimia lisätään, sitä vähemmän alkiota täytyy siirtää.

Koska palvelimet sijoitetaan satunnaisesti kehälle ja koska hajautusfunktiot eivät välttämättä sijoita alkiota tasaisesti koko kehälle, voi jotkut palvelimet joutua käsittelemään enemmän alkiota kuin toiset. Tämän takia kehälle ei oi-

keasti sijoiteta vain yhtä palvelinta, vaan jokaisella palvelimella on useita virtuaalipalvelimia kehällä. Tämä myös ratkaisee kuinka tehokkaammille palvelimille saadaan enemmän alkioita: ne luovat enemmän virtuaalipalvelimia kuin muut.

3.2. Tiedon replikointi

Yhtenäisellä hajautuksella saadaan selville alkion ensisijainen palvelin. Yleensä alkiot halutaan kuitenkin replikoida useammalle palvelimelle, jottei yksittäisten palvelimien hajoaminen kadota tietoa. Dynamossa voidaan määritellä, kuinka monelle eri palvelimelle alkiot replikoidaan. Alkioiden replikat löytyvät yksinkertaisesti kulkemalla palvelinlistan kehää myötöpäivään alkion ensisijaiselta palvelimelta.

Mikäli yksi listan palvelimista on sillä hetkellä nurin, hypätään sen yli aivan kuin sitä ei olisi ollut olemassa. Alkiot voivat siis väliaikaisesti tallentua palvelimille, joille alkiot eivät varsinaisesti kuulu. Nämä palvelimet siirtävät alkiot lopulta takaisin, kun alkioiden oma palvelin nousee pystyyn. Kaatunut palvelin voidaan myös poistaa pysyvästi, jolloin sen kaikki alkiot kopioidaan listan seuraaville palvelimille.

On kuitenkin mahdollista, että palvelimet, joilla on väliaikaisia alkioita, kaatuvat itse. Nämä puuttuvat alkiot pitää silti saada lähetettyä niiden omalle palvelimelle, kun se nousee pystyyn. Dynamo käyttää Merkle-puita puuttuvien alkioiden etsimiseen.

Dynamon Merkle-puun lehtisolmuissa ovat kaikkien avainten tiivisteet. Vanhempien solmujen arvo on tiiviste sen lapsista, ja tätä jatketaan sen juurisolmuun asti. Kahta Merkle-puuta vertaillaessa on siis nopea löytää erot niiden avaimissa: puuta kuljetaan juuresta alaspäin, kunnes solmujen tiivisteet ovat samat tai päädytään lehtisolmuihin.

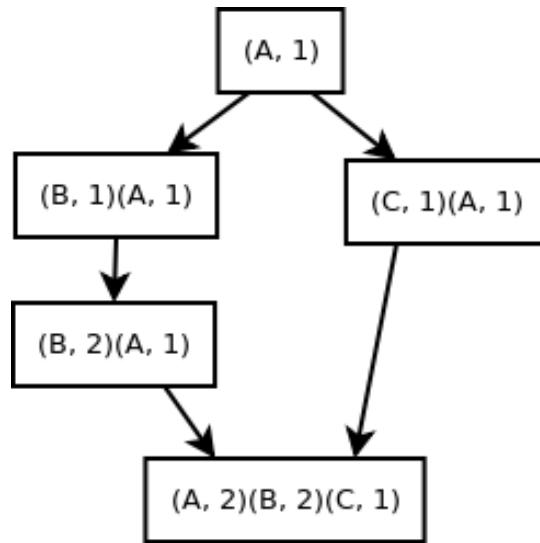
Dynamossa jokaisella virtuaalipalvelimella on erillinen Merkle-puu sen avaimille. Kun palvelin nousee pystyyn, se etsii palvelimen, joka omistaa saman avainvälin, ja etsii muutokset vertailemalla oman ja toisen palvelimen Merkle-puita.

3.3. Tiedon versiointi

Dynamo ei estä konfliktoivia muutoksia, vaan palauttaa kaikki konfliktoivat versiot hakutuloksissa. Asiakasohjelma päättää, kuinka versiot yhdistetään ja tallentaessa uuden arvon merkitsee samalla eri versiot yhdistetyiksi.

Versiointi ja konfliktien käsittely perustuu alkiokohtaisiin vektorikelloihin, jotka palautetaan hakutulosten yhteydessä. Koska Dynamossa palvelimien lukumäärä vaihtelee jatkuvasti, tallennetaan vektorikello listana (palvelin, laskuri) -pareja. Päivitetessä alkiota nostetaan oman palvelimen vektorikellon las-

kuria yhdellä ja lähetetään koko tunnettu (palvelin, laskuri) lista Dynamolle. Kuvassa 3 on esimerkki vektorikellolistan päivittämisestä.



Kuva 3. Palvelin A luo alkion, jonka jälkeen palvelimet B ja C päivittävät sitä samanaikaisesti. Lopuksi A huomaa konfliktin ja yhdistää muutokset.

Konfliktien tunnistus tapahtuu vertailemalla kahden muutoksen vektorikelloja. Muutos 2 ei ole konfliktissa muutos 1:n kanssa, jos kaikki laskurit muutos 1:n vektorikellossa ovat pienempiä tai yhtä suuria kuin muutos 2:n vektorikellossa, eikä muutos 2:lta ei puutu palvelimia, jotka ovat muutos 1:llä. Muussa tapauksessa muutokset ovat konfliktissa toistensa kanssa.

Koska alkiota voi päivittää suuri määrä palvelimia, Dynamo rajoittaa vektorikellon kokoa. Vektorikellon (palvelin, laskuri) pareissa säilytetään myös aikaleimaa, ja kun vektorikellon koko täyttyy, pudotetaan listalta pois pari, jolla on vanhin aikaleima.

3.4. Dynamon käyttötarkoitukset

Dynamoa voi käyttää moniin tarkoituksiin, jotka eivät vaadi, että luettu tieto on aina täysin ajan tasalla. DeCandia ja muut [2007] kuvaavat useampia eri käyttötarkoituksia, joihin Amazon käyttää sitä:

- Ohjelmakohtainen konfliktinkäsittely: Ostoskori on yksi esimerkki tästä. Käyttäjä voi lisätä useita tavaroita ostoskoriin. Jos lisäykset menevät väliaikaisen ongelman takia toisistaan tietämättä tietokantaan, näkee lukija kaksi erilaista ostoskoria. Lukija yhdistää ostoskorien sisällöt, jotta tietoa ei katoa. Toisaalta, jos käyttäjä oli poistanut tavaran ostoskorista, saattoi ostoskorien yhdistys lisätä sen nyt takaisin.
- Aikaleimapohjainen konfliktinkäsittely: Käyttäjän tilatiedot (session information) sisältää väliaikaista tietoa, kuten mitä sivua käyttäjä on se-

laamassa sillä hetkellä. Tilatietojen yhdistämisestä ei ole juurikaan hyötyä, joten konfliktinkäsittely säilyttää vain uusimman muutoksen ja unohtaa vanhat.

- Nopea lukukanta: Dynamon asetuksissa voi määritellä kuinka monelle palvelimelle eri tietueet replikoituvat. Tämän ääripäänä on replikoida tieto kaikille palvelimille, joka mahdollistaa harvoin muuttuvan tiedon lukemisen nopeasti miltä tahansa palvelimelta, kuten Amazonin tuoteuuttelo.

Amazonin ulkopuolella Dynamoja ja sen klooneja on käytetty moniin eri tarkoituksiin. Facebook kehitti Cassandra-kloonin nopeuttamaan sähköpostien etsimistä. IBM:n tutkijat loivat sähköpostipalvelimen, joka käyttää ainoastaan Cassandraa tietokantana. LinkedIn loi Project Voldemort -kloonin, jota LinkedIn WWW-sivusto käyttää.

4. Yhteenveto

Dynamo on avain-arvo -tietokanta, jonka kokoa voi skaalata helposti suuremmaksi tai pienemmäksi tarpeen mukaan. Dynamo käsittelee virhetilanteet automaattisesti, joten se on myös erittäin vikasietoinen. Dynamo on Amazonin WWW-sivujen pohjana, sekä sen avoimen lähdekoodin klooniversioita käytetään laajenevasti moniin eri tarkoituksiin.

Viiteluettelo

- [Brewer, 2002] Eric A. Brewer, Towards robust distributed systems. In: *Proc. of the 19th Annual ACM Symposium on Principles of Distributed Computing*, 2000, 7.
- [DeCandia et al., 2007] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lekshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Voshall and Werner Vogels, Dynamo: Amazon's highly available key-value store. *ACM SIGOPS Operating Systems Review* **41**, 6 (Dec. 2007), 205-220.
- [Fidge, 1988] Colin J. Fidge, Timestamps in message-passing systems that preserve the partial ordering. In: *Proc. of the 11th Australian Computer Science Conference*, 1988, 56-66.
- [Karger et al., 1997] David Karger, Eric Lehman, Tom Leighton, Matthew Levine, Daniel Lewin and Rina Panigrahy, Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. In: *Proc. of the 29th Annual ACM Symposium on Theory of Computing*, 1997, 654-663.

- [Lamport, 1978] Leslie Lamport, Time, clocks and the ordering of events in a distributed system. *Comm. ACM* **21**, 7 (Jul. 1978), 558-565.
- [Lamport, 1979] Leslie Lamport, Constructing digital signatures from a one-way function. Technical report CSL-98, SRI International, 1979.
- [Mattern, 1988] Friedemann Mattern, Virtual time and global states of distributed systems. In: *Proc. of the Workshop on Parallel and Distributed Algorithms*, 1988, 215-226.
- [Merkle, 1988] Ralph C. Merkle, A digital signature based on a conventional encryption function. In: *Proc. of a Conference on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, 1988, 369-378.
- [Pritchett, 2008] Dan Pritchett, BASE: an ACID alternative. *ACM Queue* **6**, 3 (May/Jun. 2008), 48-55.
- [Vogels, 2008] Werner Vogels, Eventually consistent. *ACM Queue* **6**, 6 (Oct. 2008), 14-19.

Päätöspuut hahmontunnistuksessa

Nina Tyni

Tiivistelmä

Tässä tutkielmassa käsitellään päätöspuuluokittelijoiden käyttöä hahmontunnistuksessa. Aluksi esittelen sekä hahmontunnistuksen että päätöspuumenetelmän peruseriaatteet, minkä jälkeen käyn läpi, miten päätöspuita sovelletaan hahmontunnistuksessa. Esittelen kirjallisuuden avulla, miten puita on käytetty kaukokartoitusdatan analysointiin sekä yhdistetty muihin luokittelumenetelmiin.

Avainsanat: Hahmontunnistus, päätöspuut, luokittelu

CR-luokat: I.5.2, E.1

1. Johdanto

Tietokonepohjaisen hahmontunnistuksen (engl. pattern recognition) käyttö on lisääntynyt jatkuvasti pyrittäessä järjestelmien entistä suurempaan automaattisuuteen. Hahmontunnistuksessa tehdään päätöksiä, jotka perustuvat eri tavoilla mitattujen aineistojen yksittäisten objektien, eli hahmojen, luokkaan. Päätöksentekoa varten dataa täytyy käsitellä tietyllä tavalla ennen kuin voidaan soveltaa jotain luokittelumenetelmää, jonka antaman tulosten perusteella voidaan tehdä jatkotoimia.

Päätöspuut ovat yksinkertainen luokittelumenetelmä, jota on etenkin viime aikoina käytetty ja tutkittu paljon hahmontunnistussovelluksissa. Päätöspuita käytetään määrittelemään hahmon luokka esittämällä sarja yksinkertaisia hahmon ominaisuuksia koskevia kysymyksiä. Kysymyssarja muotoillaan puuksi, jossa eteneminen riippuu kysytyn ominaisuuden arvosta.

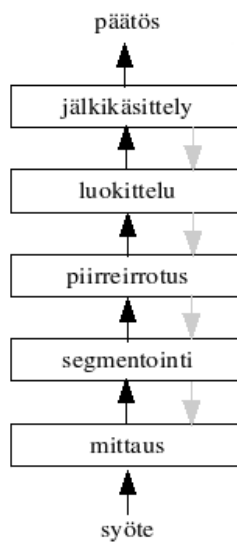
Tässä työssä esittelen aluksi hahmontunnistusta yleisesti: mitä se on, ja millainen prosessi yksittäinen hahmontunnistustehtävä on. Sen jälkeen käyn läpi päätöspuumenetelmän yleisideaa, puidenmuodostamista ja käyttöä. Lopuksi esittelen päätöspuiden käyttöä hahmontunnistuksessa muutamien esimerkkitapausten avulla.

2. Hahmontunnistus

2.1. Hahmontunnistusprosessi

Hahmontunnistuksen tavoitteena on vastaanottaa ja käsitellä raakadataa, tehdä datan ominaisuuksista päätelmiä ja toimia näiden päätelmien mukaan [Duda et al., 2001]. Esimerkiksi kasvojen tunnistaminen näköaistin avulla ja avainten löytäminen taskusta tunnustelemalla ovat ihmisen jatkuvasti kohtaamia hahmontunnistustehtäviä. Tietokonepohjaisessa hahmontunnistuksessa data voi olla esimerkiksi kuvaa tai ääntä, joka on tallennettu kameran tai mikrofoniin avulla. Tyypillisiä alan sovelluksia ovat käsinkirjoitettujen osoitteiden tunnistaminen postinlajittelussa, erilaiset tekstintunnistustehtävät (sähköpostien lajittelu) ja puheen automaattinen kirjoitetuksi tekstiksi muuntaminen. Periaatteena on tunnistaa jokin hahmo ja päätellä, millaista luokkaa kyseinen hahmo edustaa. Hahmon luokka määrää, mitä jatkotoimenpiteitä tehdään.

Jain and Duni [1997] esittää hahmontunnistusjärjestelmän suunnittelun neliosaisena prosessina. Aluksi data kerätään ja esikäsitellään: esimerkiksi otetaan valokuva kohteesta ja poistetaan kuvasta epärelevantti taustatieto. Esikäsitelystä datasta poimitaan tehtävässä kiinnostavan hahmon tunnistamisen kannalta olennaiset piirteet, joita voivat esimerkiksi kuvantunnistuksessa olla hahmon koko, muoto, väri ja tekstuuri. Valittuja piirteitä käytetään, kun datasta muodostetaan opetusdata, jonka avulla opetetaan luokittelija tunnistamaan, mihin kategoriaan yksittäinen hahmo kuuluu. Lopulta voidaan luokittelijan avulla tunnistaa uusia objekteja. Kuvassa 1 on yksi tapa jakaa hahmontunnistusjärjestelmän vaiheet osiin. Seuraavaksi edellä esitetyn listauksen vaiheita käsitellään yksityiskohtaisemmin vaihe kerrallaan.



Kuva 1. Esimerkki hahmontunnistusjärjestelmän osista. Kuvaus painottaa informaation yksisuuntaista liikkumista, mutta kuten harmaat nuolet osoittavat, ylemmiltä tasoilta voidaan lähettää alemmas palautetta, jota voidaan hyödyntää toiminnassa. [Duda et al., 2001]

2.2 Esikäsitteily ja segmentointi

Alkuperäinen data täytyy aluksi järjestää johonkin järkevään muotoon. Hahmontunnistustehtävässä täytyy ensinnäkin pystyä erottamaan havaitusta informaatiiovirrasta yksittäiset hahmot toisistaan. Tämä tapahtuu segmentointivaiheessa, jossa siis erotellaan yksittäiset hahmot toisistaan sekä hahmot tehtävän kannalta turhasta taustasta. Segmentointi olisi luonnollisesti helppoa, jos tiedettäisiin jo, miten yksittäiset kohteet tunnustetaan. Koska kohteita kuitenkin ollaan erottelemassa juuri niiden tunnistamista ja luokittelua varten, ei vielä tiedetä, koska siirrytään hahmosta toiseen tai missä edes menee kiinnostavan hahmon ja pelkän taustan raja. Duda ja muut [2001] käyttävät segmentoinnista esimerkkinä liukuhihnalla kuvattuja kaloja, joista jaetaan kahteen luokkaan lajin mukaan. Ihanteellisessa tapauksessa kalat olisivat hihnalla peräkkäin, selvästi toisistaan erillään. Käytännössä ne kuitenkin ovat sekaisin, osittain toistensa päällä ja vierekkäin, joten segmentointivaiheessa täytyy päättää, missä yksi kala loppuu ja toinen alkaa.

Segmentoinnin lisäksi esikäsitteilyyn liittyy yleensä datan yleistä siistimistä ja koherenttiuden tarkistamista. Dataa saattaa olla useassa paikassa ja useassa muodossa (esimerkiksi eri tietokannoissa), joista täytyy koota yhtenäinen kokonaisuus. Lisäksi mittausvaiheessa tapahtuneita virheitä ja puuttuvia tapauksia joudutaan ehkä tasoittamaan ja poistamaan. Attribuuttien mitta-asteikot tai

arvoalueet saattavat myös kaivata käsittelyä: numeerisia arvoja voidaan haluta jaotella ryhmiä tai arvot täytyy luokittelualgoritmia varten muuntaa tietyille arvovälille. Hahmontunnistuksessa esikäsittely jatkuu tavallaan vielä seuraavassakin vaiheessa, piirreirrotuksessa, jossa datan määrää pyritään vähentämään ja valitsemaan jatkokäsittelyyn vain hahmojen olennaisimpia piirteitä.

2.3 Piirreirrotus

Useimmiten mittausvaiheessa kertynyttä dataa on niin paljon, ettei sitä voida suoraan alkaa käsitellä. Hahmoja saatetaan kuvata redundanteilla attribuuteilla tai attribuuteilla, joista ei välttämättä ole hyötyä hahmon luokittelussa. Redundanteilla attribuuteilla tarkoitetaan attribuutteja, jotka korreloivat toistensa kanssa voimakkaasti. Kaikkia tällaisia attribuutteja ei kannata pitää mukana käsittelyssä, koska ne eivät anna luokittelutilanteessa mitään ylimääräistä informaatiota.

Luokitteluvaihetta silmälläpitäen esikäsittelyvaiheessa muodostettu data muunnetaan *piirreirrotuksessa* kompaktimpaan muotoon, jossa attribuutteja on vähemmän kuin alkuperäisessä datassa [Webb, 1999]. Datan konkreettisen määrän vähentämisen lisäksi attribuuteista tutkitaan, kuinka niiden arvot eroavat eri kategorioihin kuuluvilla objekteilla. Tavoitteena on valita kaikista muuttujista ne, jotka sisältävät eniten erotteluvaa informaatiota. Tällöin objekteista etsitään ominaisuuksia, joiden arvot ovat mahdollisimmat samanlaisia keskenään samaan luokkaan kuuluvilla objekteilla ja erilaisia eri luokkiin kuuluvilla [Duda et al., 2001]. Näin luokittelijalle voidaan antaa relevantti piirrejoukko, mikä parantaa etenkin yksinkertaisten luokittelijoiden suoritusta.

Piirreirrotuksessa käytetään kaikkia saatavilla olevia muuttujia. Redundantit tai tehtävän kannalta turhaa informaatiota sisältävät attribuutit kannattaa poistaa ajan ja laskentakustannusten säästämiseksi [Webb, 1999]. Prosessin keventämisen lisäksi muuttujien vähentäminen voi vähentää luokittelussa tapahtuvia virheitä, sillä kun huomioon otettavien attribuuttien määrä vähenee, luokittelijan kompleksisuus pienenee. Vaikka luokittelijan saama informaatio lisääntyy, kun sille käsiteltäväksi annettavien attribuuttien määrä lisääntyy, voi sen yleistyskyky kärsiä, koska samalla luokitteluprosessi monimutkaistuu.

Jos hahmontunnistustehtävässä käsitellään kuvaa tai ääntä, on myös tärkeää löytää hahmoista piirteitä, jotka pysyvät samoina riippumatta syötteeseen liittyvistä epärelevanteista muutoksista [Duda et al., 2001]. Esimerkiksi kuvantunnistuksessa piirteiden pitäisi olla mittakaavasta ja kuvakulmasta riippumattomia, jotta eri etäisyyksiltä kuvatut hahmot voidaan tunnistaa samaan kategoriin kuuluviksi. Edellä mainitussa kalaesimerkissä tällaisia piirteitä ovat kalan

muoto, väri ja pintarakenne. Piirreirrotuksen lopputuloksena on (reaali)lukuvektori, johon hahmon tärkeiksi katsotut piirteet on tiivistetty luokittelua varten.

2.4 Luokittelu

Piirreirrotuksessa muodostettua piirrevektoria hyödynnetään, kun päätetään, mihin kategoriaan, eli luokkaan, yksittäinen objekti kuuluu. Koska täydellisesti toimiva luokittelija on lähes aina mahdoton toteuttaa, usein luokittelussa lähinnä määritetään, millä todennäköisyydellä tietynlainen objekti kuuluu eri luokkiin [Duda et al., 2001]. Luokittelijan suunnittelu on siis usein käytännössä tilastollinen tehtävä.

Tan ja muut [2005] määrittelevät *luokittelijan* (classifier) tai *luokittelutekniikan* systemaattiseksi lähestymistavaksi luokittelumallin muodostamiseksi syötteestä. Tässä työssä käsiteltävien päätöspuiden lisäksi muita luokittelutekniikoita ovat muun muassa sääntöpohjaiset luokittelijat, neuroverkot ja naiivi Bayesin luokitin. Kukin tekniikka soveltaa omaa oppimisalgoritmiaan, jonka avulla pyritään löytämään datan attribuuttien ja luokkaleimojen väliseen suhteeseen parhaiten sopiva malli.

3. Päätöspuut

3.1 Päätöspuumenetelmän peruseriaate

Hahmontunnistusprosessissa käsiteltävä data voi olla numeerista tai *nominaalista*. Numeerista dataa kuvataan aiemmin esitettyyn tapaan luvuvektoreilla, eli käytännössä joukkona lukuarvona, jotka ilmaisevat hahmon eri attribuuttien arvoja. Numeeriselle datalle on olemassa yksiselitteinen tulkinta arvojärjestyksestä ja yksittäisten arvojen samankaltaisuudesta. Voidaan helposti laskea, että vaikkapa pituuden arvo 160 on lähempänä 165:tä kuin 150:ntä.

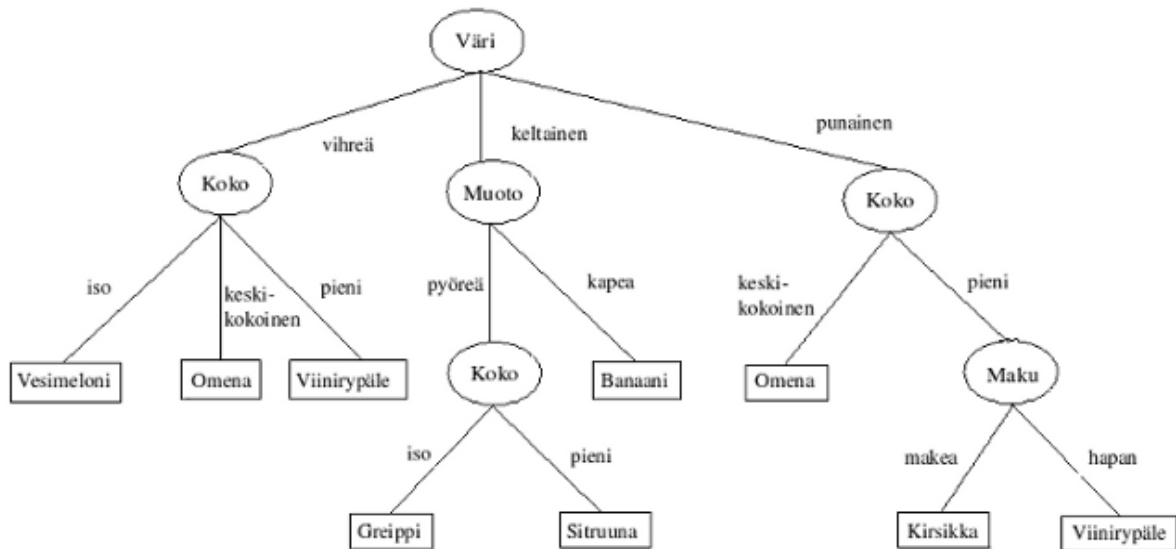
Nominaalista dataa käsiteltäessä hahmoja kuvaillaan erilaisilla attribuuteilla, ominaisuuksilla tai muuttujilla, joilla ei ole minkäänlaista luonnollista järjestystä tai samankaltaisuuden mitta [Duda et al., 2001]. On mahdotonta sanoa, miten vaikkapa erilaisia muotoja voidaan järjestää tai ryhmitellä samankaltaisuuden mukaan: onko soikea muoto lähempänä nelikulmiota kuin kolmiota? Tällaisissa tapauksissa hahmojen ominaisuuksia kuvataan aiemmin esiteltyyn reaali-luvuvektorien sijaan *d-monikoilla* (d-tuple). D-monikoissa valittuihin attribuutteihin liitetään vastaavat ominaisuudet. Esimerkiksi joitain hedelmiä voitaisiin kuvata monikolla {punainen, kiiltävä, makea, pieni}, joka on lyhyempi ver-

sio listasta *väri=punainen, tekstuuri=kiiltävä, maku=makea ja koko=pieni* [Duda et al., 2001].

Helppo lähestymistapa hahmon luokitteluun on esittää sarja sen ominaisuuksia koskevia kysymyksiä siten, että seuraava kysymys riippuu aina edellisen vastauksesta [Duda et al., 2001]. Menettely sopii hyvin ei-metriselle datalle, koska kaikki kysymykset voidaan esittää niin, että niihin vastaus on joko ”kylä/ei” tai ”arvo(ominaisuus) \in arvojoukko”. Tällainen kysymysten ja vastausten sarja voidaan järjestää *päätöspuuksi*, joka on hierarkkinen, solmuista ja suunnatuista kaarista koostuva rakenne. Päätöspuissa on kolmenlaisia solmuja:

- *Juurisolmu*, johon ei tule yhtään kaarta, ja josta lähtee nolla tai enemmän kaarta ulospäin.
- *Sisäsolmuja*, joista jokaiseen tulee tasan yksi kaari, ja lähtee yksi tai useampi kaari.
- *Lehti-* eli *päätösolmuja*, joilla on kullakin vain yksi tuleva kaari, eikä lainkaan lähteviä.

Puussa jokaiseen lehtisolmuun liittyy luokkatunniste, joka kertoo lehtisolmuun päätyvien tapausten todennäköisen luokan. Esimerkiksi kuvassa 2 mahdollisia luokkatunnisteita ovat *vesimeloni, omena, viinirypäle, greippi, sitruuna, banaani, omena* ja *kirsikka*. Juureen ja sisäsolmuihin tallennetaan objektin attribuuttien arvoja testaavia ehtoja, joiden perusteella toisistaan poikkeavat objektit erotellaan [Tan et al., 2005]. Jakamiseen voidaan käyttää yhtä tai useampaa attribuuttia, mutta yleensä on suosittu yhteen attribuuttiin perustuvia jakoja [Murthy, 1998]. Siksi tässä työssä käsitellään jatkossa tilanteita, joissa datan jakamiseen käytetään vain yhtä attribuuttia kerrallaan. Sisäsolmuilla voi myös olla useampia lapsisolmuja, mutta kaikki monitiepuut voidaan supistaa vastaavasti toimiviksi binääripuiksi, joissa jokaisella solmulla on kaksi lasta.



Kuva 2. Esimerkki päätöspuusta [Duda et al., 2001]

Objektin luokittelu aloitetaan puun juuresta. Sen jälkeen edetään puussa alaspäin aina siihen haaraan, johon sisäsolmujen testien tulokset osoittavat. Näin jatketaan, kunnes päädytään johonkin lehtisolmuun, josta tapauksen todennäköinen luokka selviää. Oletetaan esimerkiksi, että jotain hahmoa kuvataan demonikolla $\{punainen, kiiltävä, makea, pieni\}$. Kuvan 2 päätöspuulla luokittelu etenisi ensin punaisen värin perusteella oikeanpuolimmaiseen haaraan. Koska koko-attribuutin arvo on pieni, siirrytään koko-solmun oikeanpuoleiseen haaraan, jossa testataan maku-attribuuttia. Maku on makea, joten päädytään lehtisolmuun, jonka luokkaleima on *kirsikka*. Täten kyseinen hahmo luokitellaan kirsikaksi.

3.2 Päätöspuun muodostaminen

Useimmat päätöspuumenetelmät suosivat puun muodostuksessa ahnetta, osoittavaa tapaa [Murthy, 1998], jossa puu muodostetaan juuresta lähtien. Tällöin suositaan myös rekursiivista kasvatusta, jossa jokaisessa solmussa tehdään samat tarkistukset ja toimenpiteet. Eräs tällainen algoritmi on *Huntin algoritmi* (Hunt's algorithm), johon useat päätöspuuinduktiomenetelmät perustuvat. Huntin algoritmista puu muodostetaan rekursiivisesti jakamalla opetusdataa jatkuvasti puhtaampiin osajoukkoihin [Tan et al., 2005]. Alussa kaikki tapaukset ovat juuressa. Jos D_t on solmuun t liittyvien opetustapausten joukko, ja

$y = \{ y_1, y_2, \dots, y_c \}$ luokkaleimat, Huntin algoritmi määritellään rekursiivisesti seuraavasti:

1. askel: Jos kaikki solmun D_i tapaukset kuuluvat samaan luokkaan y_i , tehdään t :stä lehtisolmu, johon liittyvä luokkaleima on y_i .

2. askel: Jos D_i :ssa on useaan eri luokkaan kuuluvia tapauksia, valitaan attribuuttien arvoja testaava ehto, jolla tapaukset jaetaan alijoukkoihin. Testin jokaiselle tulokselle luodaan lapsisolmu, ja D_i :n tapaukset jaetaan lapsille testituloksen perusteella. Tämän jälkeen algoritmia sovelletaan kaikkiin D_i :n lapsisolmuihin. [Tan et al., 2005]

Suuri osa päätöspuun suunnitteluun käytettävästä työstä kuluu solmuissa tehtävien testien tai kyselyiden kehittämiseen [Duda et al., 2001]. Jos attribuutit ovat riittäviä, toisin sanoen ne kuvaavat dataa riittävän hyvin tapausten erottamiseksi, yhdestä opetusjoukosta pystytään yleensä muodostamaan useita erilaisia kaikki opetusjoukon tapaukset oikein luokittelevia päätöspuita [Quinlan, 1986]. Useista erilaisista puista suositaan usein yksinkertaisinta, koska sen ajatellaan parhaiten kuvaavan käsiteltävän ongelman rakennetta ja siten luokittelevan oikein tapauksia myös opetusjoukon ulkopuolelta [Quinlan, 1986]. Siksi peruseriaate puun rakentamisessa on yksinkertaisuus.

Yksinkertaisuusvaatimuksesta seuraa, että solmuissa pyritään jakamaan tapaukset niin, että lopputuloksena on mahdollisimman kompakti ja vähäsolmuinen puu [Duda et al., 2001]. Etsitään siis ominaisuutta, jonka arvoja vertailemalla saataisiin välittömille jälkeläissolmuille lähetettävä data mahdollisimman puhtaaksi. Solmun sanotaan olevan *puhdas* (pure), jos kaikki sen sisältämät opetustapaukset kuuluvat samaan luokkaan [Murthy, 1998]. Puhtauden käsitettä formalisoitaessa puhutaan kuitenkin solmun *epäpuhtaudesta* (impurity). Tällöin solmun N epäpuhtautta merkitsee $i(N)$. Jos kaikki solmun N tapaukset kuuluvat samaan luokkaan, halutaan $i(N)$:n olevan 0. Jos taas eri luokkien edustajia on yhtä paljon, halutaan $i(N)$:n olevan mahdollisimman suuri. Epäpuhtaudelle on kehitetty useita matemaattisia mittoja, joiden kaikkien perusidea on suunnilleen samanlainen.

Tavallisin solmun epäpuhtautta mittaava suure on *entropiaepäpuhtaus* (entropy impurity)

$$-\sum_j P[\omega_j] \log_2 P[\omega_j],$$

jossa $P[\omega_j]$ on luokkaan ω_j kuuluvien tapausten osuus solmussa N . Jos kaikki tapaukset kuuluvat samaan luokkaan, on epäpuhtaus 0. Muussa tapauksessa epäpuhtausarvo on positiivinen saaden suurimman arvonsa silloin, kun eri luokkien osuus solmun tapauksista on yhtä suuri [Duda et al., 2001]. Epäpuh-

tausmittareita käytetään solmuissa, kun halutaan tietää, kuinka hyvin jokin attribuutti erottelisi tapauksia lapsisolmuihin.

3.3 Puun koon hallinta

Huntin algoritmissa solmuissa tehdään jakoja niin kauan, että kaikki lehtisolmujen sisältämät tapaukset kuuluvat keskenään samaan luokkaan. Jos jakoja tehdään niin kauan, että jokaisessa päätesolmussa on vain yhden luokan tapauksia, puu on todennäköisesti *ylisovitettu* (overfitted) käytettyyn opetusdataan. Tällöin lehtisolmut saattavat heijastaa datassa olleita yksittäisiä poikkeavia tai virheellisiä tapauksia.

Ylisovitettu puu luokittelee hyvin opetusdatan tapaukset, mutta suoriutuu huonosti uusista tapauksista. Koska puun muodostamisessa olennaista on olla takertumatta liikaa opetusaineistoon, täytyy jotenkin löytää keino pitää puu tarpeeksi yleisenä, mutta kuitenkin riittävän tarkkana opetusaineiston suhteen. Puun täytyy siis kuvastaa jotain olennaista suhdetta tapauksen luokan ja sen attribuuttien arvojen välillä, eikä olla pelkkä pintapuolinen yhden aineiston kuvaus [Quinlan, 1986]. Idea toteutuu vain, jos löydetään oikeat kriteerit, jotka erottelevat eri luokkiin kuuluvat tapaukset riittävän luotettavasti toisistaan.

Yksi keino ehkäistä puun liikakasvua on asettaa jokin lopetuskriteeri jaoille. Jos solmussa havaitaan, että lopetuskriteerin ehdot täytetään, lopetetaan jakojen tekeminen, vaikka kaikilla solmussa olevilla tapauksissa ei olisikaan sama luokkaleima. Lopetusehto voi olla esimerkiksi epäpuhtauden vähenemiselle ennalta asetettu kynnyсарvo: jos solmun paras jakovaihtoehto vähentäisi epäpuhtautta vähemmän kuin kyseinen kynnyсарvo, jakaminen lopetetaan. Jaot voidaan lopettaa myös, jos solmussa on jäljellä vain pieni määrä opetusaineiston tapauksista, tai jos jollain tilastollisella menetelmällä testaamalla todetaan, ettei laskennallisesti kannattavin jako eroa merkittävästi satunnaisesti tehtävästä jaosta [Duda et al., 2001]. Sopivan jakokriteerin löydyttyä jokaisessa solmussa täytyy siis päättää, kannattaako jakamista ylipäänsä jatkaa, ja kynnyсарvot ovat keino arvioida jakamisen kannattavuutta.

Puun kasvun rajoittaminen on hyödyllistä, koska se estää monimutkaisten alipuiden muodostumista puun alimmille tasoille. Tällöin vältetään myös ylisovittamisongelmalta. Oikean kynnyсарvon valinta on kuitenkin vaikeaa. Jos arvo on liian korkea, eli kasvattaminen lopetetaan liian aikaisin, voi puu olla *alisovitettu* (underfitted) malliin nähden. Alisovittaminen on ylisovittamisen päinvastainen versio: alisovitettu puu ei ole kunnolla oppinut opetusdatan rakennetta, eikä osaa luokitella tapauksia oikein. Alisovittamista vältettäessä kynnyсарvoa pienennetään. Jos kynnyсарvo kuitenkin on liian pieni, päädytään jälleen ylisov-

vitettuun puuhun, koska puun kasvua ei ole onnistuttu pysäyttämään tarpeeksi ajoissa [Tan et al., 2005]. Kynnysarvon päättämisen lisäksi kasvun ennenaikaisen lopettamisen ongelmana on se, että vaikka jossain vaiheessa todetaan, että tehtävä jako on hyödytön, voisivat kyseisen solmun jälkeiset jaot olla hyviä kokonaisuuden kannalta.

Hyvä vaihtoehto lopetuskriteerille puun koon hallintaan on *karsia* puuta, jonka on ensin annettu vapaasti kasvaa lopulliseen kokoonsa. Karsimismetodissa (pruning) rakennetaan ensin kokonainen puu, jossa minkään lehtisolmun edelleen jakaminen ei enää paranna luokittelutarkkuutta opetusdatalla, toisin sanoen solmujen epäpuhtaus on minimissään. Tämän jälkeen valmis puu karsitaan lehtisolmuista lähtien. Tämä voidaan tehdä eri tavoin. Alipuu voidaan korvata lehtisolmulla, jonka luokaksi määrätään enemmistöluokka niistä tapauksista, jotka kyseiseen alipuuhun kuuluvat. Toinen vaihtoehto on korvata kokonainen alipuu eniten käytetyllä alipuun haaralla [Tan et al., 2005]. Karsiminen lopetetaan, kun korvausten ei enää havaita parantavan puuta.

Päätöspuulle jälkikäteen tehtävä karsinta on yleensä kasvattamisen lopettamista parempi vaihtoehto, koska supistamispäätökset tehdään täyden puun perusteella [Tan et al., 2005]. Laskennallisesti karsiminen tosin käyttää turhaa energiaa, koska puuta kasvatettaessa tehdään työtä, jonka tulokset tavallaan menevät hukkaan, kun muodostettuja alipuita ei käytetäkään lopputuloksessa.

Karsimisen jälkeen päätöspuun lehtisolmujen luokkaleimat eivät enää ole yksiselitteisiä, koska nyt solmuihin voi päätyä useidenkin eri luokkien edustajia. Tällöin solmun leimaksi valitaan se luokka, jonka edustajia solmussa on eniten [Tan et al., 2005].

3.4 Päätöspuiden arviointi

Kun päätöspuu on rakennettu, täytyy se testata. Suosittu tapa on jakaa data jo ennen puun muodostamisen aloittamista kahteen erilliseen osaan, opetus- ja testidataan, jossain sopivassa suhteessa. Yleistä on varata kaksi kolmasosaa aineistosta puun muodostamiseen ja kolmasosa testaamiseen. Opetusaineistosta (training data) muodostetaan puu, jonka luokittelukykyä sitten testataan testidatalla. Käyttämällä erillistä testidataa voidaan paremmin arvioida, kuinka puu luokittelee sille ennestään vieraan tapaukset. Huonona puolena tosin on se, että opetusta varten tapauksia jää käytettäväksi vähemmän [Tan et al., 2005].

Kiinteälle joukkojaolle vaihtoehtoinen lähestymistapa saada testiaineistoa on ristiinvalidointi (cross validation). Ristiinvalidoinnissa opetusaineisto jaetaan satunnaisesti m osajoukkoon, joista jokaisen koko on n/m , missä n on ope-

tusaineistossa olevien tapausten lukumäärä [Duda et al., 2001]. Sen jälkeen puu muodostetaan m kertaa niin, että joka kerralla käytetään eri osajoukkoa testiaineistona. Usein joukkoja muodostetaan kymmenen, jolloin puhutaan 10-kertaisesta ristiinvalidoinnista. Tällöin testit tehdään yhteensä 10 kertaa, jotta jokaista joukkoa päästään käyttämään testaamiseen tasan kerran.

Puuta testatessa testiaineiston tapaukset luokitellaan puun avulla, ja verrataan, kuinka hyvin puun tapauksille määräämät luokkaleimat vastaavat tapausten todellisia luokkia. Oikein ja väärin ennustettujen tapausten määrät kirjataan ylös *sekaannusmatriisiksi* (confusion matrix) kutsuttuun taulukkoon. Taulukossa 1 on malli sekaannusmatriisista kahden luokan luokittelutehtävälle. Jokainen taulun merkintä f_{ij} tarkoittaa niiden luokkaan i kuuluvien tapausten määrää, joiden on ennustettu kuuluvan luokkaan j . Esimerkiksi f_{01} on luokkaan 0 kuuluvien tapausten määrä, jotka on virheellisesti luokiteltu luokan 1 jäseniksi. Taulukosta laskettuna oikein luokiteltujen tapausten määrä on $(f_{11} + f_{00})$ ja väärin ennustusten määrä $(f_{10} + f_{01})$. [Tan et al., 2005]

		Ennustettu luokka	
		<i>Luokka = 1</i>	<i>Luokka = 0</i>
Todellinen luokka	<i>Luokka = 1</i>	f_{11}	f_{10}
	<i>Luokka = 0</i>	f_{01}	f_{00}

Taulukko 1. Sekaannusmatriisi kahden luokan luokittelutehtävälle [Tan et al., 2005]

Sekaannusmatriisin tiedoista lasketaan puun suoritusta kuvaavia tunnuslukuja. Tärkeä mittari on *tarkkuus* (accuracy), joka kuvaa oikein luokiteltujen tapausten suhdetta koko aineistoon. Tarkkuus määritellään seuraavasti:

$$\text{Tarkkuus} = \frac{\text{Oikeinluokiteltujen tapausten lukumäärä}}{\text{Kaikkien luokittelujen lukumäärä}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

Vastaavasti lasketaan *virheaste* (error rate), joka ilmaisee väärin menneiden luokittelujen osuuden kaikista tapauksista. Yleensä luokittelualgoritmeissa pyritään muodostamaan malleja, jotka saavuttavat testiaineistolla mitatessa korkeimman mahdollisen tarkkuuden tai pienimmän virheasteen.

Sekä tarkkuus että virheaste ovat yleisesti käytettyjä luokittelijan suoritusta kuvaavia mittareita, joiden avulla voidaan arvioida luokittelijan hyvyttä sekä vertailla eri luokittelijoita keskenään. Jos esimerkiksi valmiin päätöspuun luokittelutarkkuus todetaan testausvaiheessa huonoksi, voidaan palata muodos-

tuksen alkuun, ja käyttää aiemmalla kierroksella hankittua kokemusta hyödyksi. Tarkkuutta voidaan esimerkiksi verrata johonkin ennalta päätettyyn odotettuun tarkkuuteen, joka valmiin puun halutaan vähintään saavuttavan. Usein puun rakentaminen onkin iteratiivista, eli ensimmäistä valmista puuta ei vain suoraan oteta käyttöön, vaan testauksen pohjalta tehdään useita vaihtoehtoja, joiden keskinäisen vertailun pohjalta valitaan lopulta paras.

3.5 Menetelmän edut ja ongelmat

Päätöspuiden vahvuus on niiden yksinkertaisuus. Etenkin pienten päätöspuiden tulkitseminen on helppoa, ja parhaimmillaan tapaus voidaan luokitella muutamalla nopealla testillä. Luokittelu on laskennallisesti nopeaa, koska yhtä tapausta kohden käydään puu vain kerran läpi. Tapauksen luokitteluun käytettävän kysymyssarjan merkitys on selvä alan asiantuntijalle, ja puuformalisi itsessään on intuitiivisesti vetoava [Murthy, 1998]. Selkeän ulosantinsa vuoksi puut ovat hyvä apukeino, kun halutaan ymmärtää datan rakennetta syvällisemmin. Päätöspuusta näkee helposti, mitä attribuutteja on käytetty luokitteluun, mistä voidaan päätellä, mitkä attribuutit ovat kyseisessä tunnistustehtävässä kiinnostavia. Puiden on myös todettu pärjäävän tarkkuuden vertailuissa muille luokittelumenetelmille [Tan et al., 2005].

Koska käsiteltävästä mallista ja datajakaumasta tarvitaan etukäteen vain vähän etukäteistietoa, voidaan päätöspuilla mallintaa hyvin erilaisia jakaumia [Murthy, 1998]. Päätöspuut pystyvät käsittelemään sekä uni- että multimodaalista dataa (eli data voi sisältää esimerkiksi sekä ääni- että kuvamateriaalia). Lisäksi datassa voi olla sekaisin numeerisia ja nominaalisia muuttujia, joten menetelmä ei aseta minkäänlaisia tyyppivaatimuksia luokiteltavalle materiaalille.

Päätöspuualgoritmit eivät ole kovin herkkiä melulle, etenkin jos ylisovittamista estäviä keinoja käytetään puunmuodostusvaiheessa. Myöskään redundantit attribuutit eivät haittaa puiden tarkkuutta, koska jos keskenään samankaltaisista attribuuteista yhtä on jo käytetty jakamisessa, ei muita enää käytetä [Tan et al., 2005]. Sen sijaan irrelevanttien attribuuttien suhteen päätöspuut eivät ole yhtä joustavia. Jos datassa on paljon luokittelussa tarpeetonta informaatiota, voi osa siitä päätyä mukaan muodostuksessa tehtäviin päätöksiin, jolloin päätöspuusta tulee laajempi kuin oikeastaan tarvitsisi. Tällaisessa puussa testataan tapauksia testeillä, jotka eivät todellisuudessa erottele tapauksia toisistaan mallin edellyttämällä tavalla. Edellä kuvailtuja ongelmia voidaan kiertää poistamalla irrelevantit attribuutit esikäsitelyvaiheessa, jolloin puun luokittelutarkkuus saattaa parantua.

Keskeinen ongelma päätöspuuinduktiossa on jo aiemmin mainittu ylisovittaminen. Ylisovittaminen liittyy päätösrajan asettamiseen. Jos puu on hyvin monimutkainen ja noudattaa tarkasti opetusdataa, se ei todennäköisesti suoriudu kovin hyvin uusien tapausten luokittelussa. Ylisovittamisongelmaa ja etenkin sen ratkaisemista puuta karsimalla on käsitelty paljon [Murthy, 1998], koska karsimisen on todettu olevan tehokas tapa parantaa päätöspuun luokittelutarkkuutta. Lisäksi karsiminen nopeuttaa luokittelua, koska sen avulla puun solmujen määrä vähenee.

Kun päätöspuuta rakennetaan rekursiivisilla algoritmeilla ja jakoihin käytettyillä attribuuteilla on useita mahdollisia arvoja, ongelmaksi muodostuu helposti datan fragmentoituminen alimmilla päätöstasoilla [Murthy, 1998]. Datan fragmentoituminen tarkoittaa sitä, että mitä enemmän alkuperäistä opetusaineistoa jaetaan osiin, sitä pienempiin otoksiin kukin jako perustuu. Siten puun viimeiset jaot perustuvat enää pienen aineiston perusteella tehtyihin päätöksiin. Kovin pienillä joukoilla tehtävien jakojen vaikutus puun lopulliseen luokittelukykyyn on harvoin merkittävä.

Jos testisolmuissa tapausten jakamiseen käytetään vain yhtä attribuuttia kerrallaan, päädytään helposti puussa toistuviin alipuihin [Tan et al., 2005]. Tällöin sama alipuu toistuu useammassa puun haarassa, ja puu kokonaisuudessaan on tarpeettoman monimutkainen ja kenties vaikeammin tulkittavissa.

4. Päätöspuiden soveltaminen hahmontunnistuksessa

Murthy [1998] esittää kolme tapaa käyttää päätöspuita datan tutkimisessa: kuvailu, yleistys ja luokittelu. Kuvailu tarkoittaa datan määrän pienentämistä muuntamalla data esitykseksi, joka säilyttää datan olennaiset piirteet. Samalla saadaan tarkka tiivistelmä alkuperäisestä datasta. Hahmontunnistuksessa tällaista kuvailua tehdään piirreirrotusvaiheessa. Yleistyksessä etsitään kuvausta riippumattoman ja riippuvaisen muuttuja välille, jotta osattaisiin ennustaa riippuvaisen muuttujan arvoja. Luokittelussa tutkitaan, sisältääkö data hyvin toisistaan erottuneisiin luokkiin jakautuneita objekteja, ja ovatko nämä luokat kiinnostavia kyseisen tunnistustehtävän kannalta. Hahmontunnistuksessa päätöspuita käytetään eniten juuri luokitteluun.

Päätöspuiden soveltaminen hahmontunnistukseen sai alkunsa 1970-luvulla tarpeesta tulkita kaukokartoitussatelliittien lähettämää kuvaa [Murthy, 1998]. Viime vuosina puiden käyttö kaukokartoitetun datan analyysissä on lisääntynyt juuri niiden nopean laskentakyvyn ja datalle asetettujen vähien vaatimusten ansiosta [Pal and Mather, 2003]. Tarkastelen seuraavaksi, miten päätöspuita

on sovellettu satelliittidatan käsittelyssä, ja kuinka menetelmän on todettu sopivan kyseiselle alalle.

4.1 Päättöpuut kaukokartoitusdatan luokittelussa

Fries ja muut [1998] ovat käyttäneet päätöspuumenetelmää Landsat-satelliiteilla hankitun kuva-aineiston käsittelyyn. Tarkoituksena oli luoda globaali maanpeitteen luokittelumenetelmä datan analysointiin. Lopullinen päätöspuun muodostukseen käytetty aineisto koostui 156 Landsat-maisemasta, joista haluttiin erotella eri kasvillisuustyyppien esiintymistä maan pinnalla. Opetusaineisto, 9306 kuvapikseliä, kattoi esikäsitteilyvaiheen jälkeen noin 10 % Landsat-aineiston kuvaamasta maa-alasta.

Päätöspuun muodostamisvaiheessa käytetty data jaettiin satunnaisesti kolmeen osaan. Varsinaisen puun kasvattamiseen käytettiin 60 % datasta. Karsinnassa hyödynnettiin 20 % aineistosta, ja loput 20 % jätettiin puun testaamiseen. Jako poikkeaa hieman puiden testaamisen yhteydessä esitetystä kaksijaosta, jossa aineiston jakoa hyödynnetään vain opetukseen ja testaamiseen. Erillisen aineiston soveltaminen karsimiseen on kuitenkin hyödyllistä samaan tapaan kuin erillisen aineiston käyttäminen testaamiseen. Käyttämällä karsimisessa tapauksia, joita ei ole käytetty puun muodostamiseen, saadaan parempi kuva puun yleistyskyvystä kuin käyttämällä koko ajan samaa aineistoa. Fries ja muut [1998] karsivat täyteen mittaan kasvanutta puuta niin kauan, että puussa oli enää se määrä solmuja, jolla saavutettiin pienin mahdollinen virheaste. Puuta siis testattiin jo karsimisen aikana, ja verrattiin väärin luokiteltujen tapausten suhteellista määrää toivottuun lopputulokseen. Opetusaineistosta muodostetun puuluokittelijan tarkkuus oli tutkimusta varten muokatulla testiaineistolla jopa 90 %, joten todellisen luokittelutarkkuuden arveltiin olevan 81-90 %.

Päätöspuiden on havaittu sopivan hyvin tämäntyyppiseen tutkimukseen, joissain asioissa jopa paremmin kuin perinteiset luokittelumenetelmät. Kuten aiemmin on esitetty, päätöspuut ovat hyvä keino saada syvällisempää tietoa käsiteltävän aineiston luonteesta. Päätöspuista selviää välittömästi, mitä attribuutteja luokkien erotteluun on käytetty. Lisäksi puut paljastavat epälineaariset ja hierarkkiset attribuuttien väliset suhteet luokkajäsennyksiä ennustaessaan. Tiedosta voi olla hyötyä, kun myöhemmin etsitään sopivimpia syötteitä maanpeitteen kuvailuun [Fries et al., 1998]. Päätöspuuta ei siis pelkästään käytetä maanpeitteen luokitteluun vaan myös kartoittavaan tutkimukseen, jossa analysoidaan eri attribuuttien kykyä erotella maanpeitteitä toisistaan.

Landsat-kuvia on myöhemmin hyödyntänyt myös Brown de Colstoun ja muut [2003], jotka ovat tutkineet päätöspuuluokittelua Yhdysvaltojen kansal-

lispuistojen kasvillisuuden analysointiin. Mielenkiinnon kohteena oli erityisesti päätöspuiden soveltuminen datalle, jonka erotuskyky on korkea, eli josta voidaan erottaa hyvinkin pieniä yksittäisiä kohteita. Myös tässä tapauksessa todettiin päätöspuiden selviytyvän tehtävästä hyvin: tarkkuus 82,05 % ylitti odotusarvon (joka tässä tapauksessa oli 80 %) ja kasvillisuuden ja ei-kasvillisuuden tunnistamisen virheaste oli lähes olematon.

Useiden käytännön sovellusten lisäksi päätöspuita on käsitelty myös teoreettisimmissa tutkimuksissa vertailemalla puita muihin luokittelijoihin juuri maanpeitteenluokittelutehtävissä. Esimerkiksi Pal ja Mather [2003] vertailivat puita neuroverkkoihin ja suurimman uskottavuuden estimoinniksi kutsuttuun tilastolliseen päättelymenetelmään. Päätöspuista testattiin kahta varianttia: yksi- ja moniulotteisia (eli käyttävät solmuissa yhtä tai useampaa muuttujaa jakokriteerinä). Luokittelijoita vertailtiin niiden tarkkuuksien perusteella.

Palin ja Matherin testeissä yksiulotteinen päätöspuu pärjäsikin tarkkuudessa neuroverkoille ja uskottavuuden estimoinnille. Hyvin moniulotteisella datalla päätöspuut kuitenkin häviävät vertailussa muille testatuille luokitteluille. Moniulotteinen päätöspuu ei tämä tutkimuksen mukaan ollut yksiulotteista parempi. Pal ja Mather [2003] kokeilivat myös päätöspuiden *tehostamista* (boosting), minkä havaittiin parantavan puiden tarkkuutta jonkin verran, muttei niin paljon, että puut olisivat voittaneet muut tekniikat. Tehostamisessa varsinaisen päätöspuun lisäksi muodostetaan sille eräänlaisia jatkoluokittelijoita, jotka pakotetaan keskittymään ”pääpuun” virheisiin [Brown de Colstoun et al., 2003]. Kyseistä tekniikkaa ovat kokeilleet myös Friedl ja muut [1999], joiden mukaan tehostamistekniikalla parannellut luokittelijat sopivat tietyin varauksin kaukokartoitusdatalle.

4.2 Päätöspuiden yhdistäminen muihin luokittelumenetelmiin

Luokittelussa käytettävän datan muuttuessa aina vain kompleksisemmaksi on yksittäisten luokittelutekniikoiden kyky käsitellä datan määrää joissain tapauksissa heikentynyt [Duin and Tax, 2000]. Ratkaisuksi on ehdotettu useiden luokittelijoiden yhdistämistä yhdessä luokittelutehtävässä. Eri luokittelijat toimivat eri tavalla erityyppisillä muuttujilla, ja erilaiset tekniikat voivat paljastaa erilaisia tapoja jaotella dataa. Siksi useiden menetelmien yhdistäminen monimutkaisissa tehtävissä on usein kannattavaa. Esittelen tässä lyhyesti kaksi tapaa, joilla päätöspuita käytetään yhdessä muiden luokittelijoiden, kuten neuroverkkojen ja tukivektoriluokittelijoihin (support vector classifier).

Yhdistetyt luokittelijat toimivat yksinkertaisesti esimerkiksi siten, että tehtävään valituille luokittelumenetelmille annetaan kullekin luokittelijalle parhaak-

si katsottu syöte (tai alkuperäisen syötteen piirteiden osajoukko). Eri syötteen antamisessa ajatuksena on juuri se, että kukin luokittelija on vahvimillaan tietynlaisen datan käsittelyssä. Käyttämällä sopivia luokittelijoita datan eri osille saadaan datasta paremmin irti kaikki mahdollinen informaatio. Kukin luokittelija antaa oman arvionsa kysytyn hahmon luokasta, minkä jälkeen lopullinen päätös sen luokkajäsenyydestä tehdään esimerkiksi sen perusteella, mitä enemmistö luokittelijoista sanoo, tai määräämällä hahmolle se luokka, jonka luotettavimmaksi katsottu luokittelija antaa.

Eri menetelmiä voi yhdistää myös muilla tavoilla kuin laittamalla kaikki ratkomaan samaa ongelmaa. Luokittelijoita voi nimittäin myös sulauttaa toisiinsa, kuten Farrell ja muut [1994] ovat tehneet yhdistäessään päätöspuun neuroverkkoihin. Farrellin ja muiden käyttämä *neuroverkkopuu* (neural tree network) on hierarkkinen luokittelija, joka yhdistää eteenpäinsyöttävän neuroverkon (feedforward neural network) ja päätöspuurakenteen [Farrell et al., 1994]. Neuroverkkopuussa jokainen solmu on erillinen neuroverkko, joka käyttää päätöksentekoon kaikkia piirre-elementtejä. Tällaisen rakenteen päätösrajat ovat perinteisiä päätöspuita joustavampia. Neuroverkkopuun rakentaminen ja karsiminen on samanlainen prosessi kuin päätöspuilla, solmuissa tehtävien jakojen mekanismi vain on siinä mielessä kehittyneempi, että jakoa tehtäessä käytettävissä on enemmän informaatiota. Myös neuroverkkopuun todettiin olevan tarkkuudeltaan samaa luokkaa muiden tutkimuksen luokittelumenetelmien kanssa.

5. Johtopäätökset

Päätöspuut ovat yksinkertainen ja joustava luokittelumenetelmä, joka soveltuu monenlaisten aineistojen mallintamiseen. Päätöspuiden muodostuksessa huomionarvoisimmat suunnitteluongelmat ovat jakojen lopetuskohta sekä jakoperuste. Etenkin jakojen lopetustapa tai valmiin puun karsiminen voivat vaikuttaa huomattavasti puun luokittelutarkkuuteen. Puun rakentamiseen liittyvät keskeiset huomionkohteet riippuvat pitkälti datasta, jota on tarkoitus käsitellä. Erityyppisillä aineistoilla ongelmat ovat erilaisia.

Hahmontunnistuksessa päätöspuita on käytetty paljon luokitteluun erilaisissa tehtävissä. Puut vaativat hyvin vähän ennakkotietoa käsiteltävästä datasta, mikä mahdollistaa hyvin erilaisten aineistojen mallintamisen. Lisäksi ne soveltuvat erityisen hyvin epämetrisen datan käsittelyyn, minkä takia ne ovat tärkeä työkalu hahmontunnistuksessa. Puiden käyttöä ja niiden yhdistelyä muihin luokittelumenetelmiin on tutkittu paljon, ja on todettu, että puut soveltuvat hyvin hahmontunnistustehtäviin niin yksinään kuin muihin luokittelijoi-

hin yhdistettynä. Erityisesti monimutkaisten aineistojen käsittelyssä puiden tukeminen muilla menetelmillä on kannattavaa.

Tässä työssä sekä hahmontunnistus että päätöspuut käsiteltiin hyvin yleisellä tasolla. Esimerkiksi päätöspuiden osalta ei otettu kantaa eri algoritmeihin vaan pitäydyttiin peruseriaaiteissa, jotka ovat suunnilleen samoja kaikissa eri induktiomenetelmissä. Keskittymällä esimerkiksi vain yhteen algoritmiin esityksestä olisi saanut tarkemman, mutta koska tarkoitus oli lähinnä antaa päätöspuista ja hahmontunnistuksesta tarpeelliset taustatiedot varsinaisten käytännön sovellusten ymmärtämistä varten, yleisempi esitystapa lienee perusteltu.

Viiteluettelo

- [Brown de Colstoun et al., 2003] Eric C. Brown de Colstoun, Michael H. Story, Craig Thompson, Kathy Commisso, Timothy G. Smith and James R. Irons, National Park vegetation mapping using multitemporal Landsat 7 data and a decision tree classifier. *Rem. Sens. Environ.* **85** (2003), 316-327.
- [Duda et al., 2001] Richard O. Duda, Peter E. Hart and David G. Stork, *Pattern Classification*. Wiley, 2001.
- [Duin and Tax, 2000] Robert P.W. Duin and David M.J. Tax, MML inference of decision graphs with multi-way joins. In: *Lecture Notes in Computer Science* **2557** (2000). Springer, 131-142.
- [Farrell et al., 1994] Kevin R. Farrell, Richard J. Mammone and Khaled T. Assaleh, Speaker recognition using neural networks and conventional classifiers. *IEEE Trans. Speech and Audio Proc.* **2**, 1 (1994), 194-205.
- [Friedl et al., 1999] Mark A. Friedl, Carla E. Brodley and Alan H. Strahler, Maximizing land cover classification accuracies produced by decision trees at continental to global scales. *IEEE Trans. Geosc. Rem. Sens.* **37**, 2 (1999), 969-977.
- [Fries et al., 1998] R.S. de Fries, M. Hansen, J.R.G. Townshend and R. Sohlberg, Global land cover classifications at 8 km spatial resolution: the use of training data derived from Landsat imagery in decision tree classifiers. *Int. J. Rem. Sensing* **19**, 16 (1998), 3141-3168 .
- [Jain and Duni, 2007] Anil K. Jain, Robert P. W. Duin, "pattern recognition" in: *The Oxford Companion to the Mind*. Richard L. Gregory. Oxford University Press 1987. Oxford Reference Online. Checked 16.12. 2009. <<http://www.oxfordreference.com/views/ENTRY.html?subview=Main&entry=t159.e656-s3>>.

- [Murthy, 1998] Sreerama K. Murthy, Automatic construction of decision trees from data: a multi- disciplinary survey. *Data Min. Knowl. Discov.* **2**, 4 (1998), 345-389.
- [Pal and Mather, 2003] Mahesh Pal and Paul M. Mather, An assessment of the effectiveness of decision tree methods for land cover classification. *Rem. Sens. Environ.* **86** (2003), 554-565.
- [Quinlan, 1986] John Ross Quinlan, Induction of decision trees. *Mach. Learn.* **1**, 1 (1986), 81-106.
- [Tan et al., 2005] Pang-Ning Tan, Michael Steinbach and Vipin Kumar, *Introduction to Data Mining*. Addison-Wesley, 2005.
- [Webb, 1999] Andrew R. Webb, *Statistical Pattern Recognition*. Hodder Arnold, 1999.

Ohjaaja: Jyri Saarikoski

Muurahaisyhdyskuntaoptimointi

Ilkka Virolainen

Tiivistelmä.

Muurahaisyhdyskuntaoptimointi on metaheuristiikka kombinatoristen optimointiongelmiin approksimointiin. Siinä keinotekoisiksi muurahaisiksi kutsutut agentit etsivät ratkaisun annettuun optimointiongelmaan, toimintamallinaan simulaatio oikeiden muurahaisien ravinnonetsintämekanismista. Tässä tutkielmassa esitellään metaheuristiikka, tärkeimpiä algoritmeja ja lyhyesti teoriaa menetelmän taustalla.

Avainsanat ja -sanonnat: Muurahaisyhdyskuntaoptimointi, kombinatorinen optimointi, parviälykkyys

CR-luokat: G.1.6, I.1.2

1. Johdanto

Optimoinnilla tarkoitetaan parhaan mahdollisen vaihtoehdon valintaa mahdollisten vaihtoehtojen joukosta. Eräs optimoinnin haara on *kombinatorinen optimointi*, jossa mahdollisten vaihtoehtojen joukko eli *hakuavaruus* on äärellinen. Hakuavaruus on useimmiten kuitenkin niin suuri, ettei ole mahdollista tai tarkoituksenmukaista käydä jokaista elementtiä läpi.

Tarvitaan siis toisenlainen lähestymistapa ongelman ratkaisemiseen. *Heuristiset menetelmät* ovat yksi tällainen lähestymistapa. Heuristiset menetelmät ovat menetelmiä, jotka tuottavat hyväksyttävän ratkaisun ongelmaan monissa tilanteissa. Löydetty ratkaisu ei kuitenkaan välttämättä ole optimaalinen. *Metaheuristiikka* on heuristinen menetelmä, joka soveltuu kokonaisen ongelmaluokan ratkaisemiseksi.

Optimointiongelmillä on suuri merkitys paitsi akateemisessa maailmassa, myös käytännössä. Käytännön esimerkki optimointiongelma on esimerkiksi aikatauluttaminen. Miten aikatauluttaa esimerkiksi junien kulkeminen mahdollisimman tehokkaalla tavalla? Yleinen malliesimerkki kombinatorisesta optimointiongelma on kauppamatkustajan ongelma, jossa kauppamatkustajan tehtävänä on kiertää jokainen annetuista kaupungeista tasan kerran, siten että kuljetun matkan kokonaispituus on mahdollisimman pieni.

Monet kombinatorisista optimointiongelmissa ovat *NP-kovia*, eli niiden ratkaisemiseen optimaalisen ratkaisun takaavilla menetelmillä vaadittava aika kasvaa pahimmassa tapauksessa eksponentiaalisesti ongelman koon suhteen. Siksi on usein tarkoituksenmukaisempaa käyttää optimointiin *approksimointimenetelmiä*. Approksimoimalla menetetään

takuu optimaalisen ratkaisun löytymisestä, mutta saavutetaan usein käytännöllinen suoritus-aika.

Muurahaisyhdyskuntaoptimointi (ant colony optimization) on metaheuristiikka kombinatoristen optimointiongelmiä ratkaisemiseen approksimoimalla. Siinä keinotekoisiksi muurahaisiksi kutsutut ohjelmalliset agentit rakentavat hyviä ratkaisuja annettuun optimointiongelmaan. Blumin [2005] mukaan muurahaisyhdyskuntaoptimoinnin luokittelu riippuu otetusta näkökulmasta: operaatiotutkimuksen näkökulmasta se on metaheuristiikka, kun taas tekoälyn näkökulmasta se on *parviälykkyyden* haara. Parviälykkyydellä tarkoitetaan tässä tekoälytutkimukseen lukeutuvaa tutkimusaluetta, jonka tavoitteena on suunnitella älykkäitä, monen toimijan järjestelmiä ottamalla vaikutteita sosiaalisten hyönteisten yhteistoiminnasta.

Tämän tutkielman tavoitteena on antaa yleiskäsitys muurahaisyhdyskuntaoptimoinnista, sen toimintaperiaatteista ja teoriapohjasta sekä esitellä sen toimintaa. Luvussa 2 esitellään muurahaisyhdyskuntaoptimoinnin historiaa ja biologista taustaa. Lisäksi käsitellään lyhyesti muurahaisten ravinnonetsintämekanismia, joka toimi inspiraationa menetelmän kehitykselle. Luvussa 3 käsitellään kombinatorista optimointia tarkemmin ja esitellään alakohtiin jaettuna kaksi lähestymistapaa kombinatoristen optimointiongelmiä ratkaisemaan approksimoimalla. Luvussa 4 esitellään muurahaisyhdyskuntaoptimoinnin metaheuristiikka yleisellä tasolla. Luvun kohdissa määritellään kombinatorinen optimointiongelma formaalisti, määritellään yksityiskohtaisesti keinotekoisien muurahaisten ominaisuudet ja käydään metaheuristiikka näiden tietojen valossa läpi. Luvussa 5 esitellään käytännön muurahaisyhdyskuntaoptimointialgoritmeja. Luku on jaettu kolmeen kohtaan, joista jokainen edustaa yhtä tärkeää algoritmia: Ant System, *MAX – MIN* Ant System ja Ant Colony System. Luvussa 6 käydään lyhyesti läpi tärkeimpiä osa-alueita menetelmän teoriapohjasta: konvergenssia ja aikakompleksisuutta. Luvussa 7 esitellään muita algoritmeja ja pohditaan aiheen kehitysnäkymiä. Luvussa 8 tehdään lyhyt yhteenveto tutkielmasta.

2. Muurahaisyhdyskuntaoptimoinnin historiaa

Muurahaisyhdyskuntaoptimointialgoritmeja julkaisi ensimmäisenä Marco Dorigo [1992] kollegoineen 1990-luvun alkupuolella. Idea näiden menetelmien kehitykseen tuli muurahaisyhdyskuntien ravinnonetsintämekanismista. Tämän mekanismin tärkeimpänä yksityiskohtana on se, miten muurahaiset löytävät lyhimmän polun muurahaispesän ja ravinnonlähteen välillä.

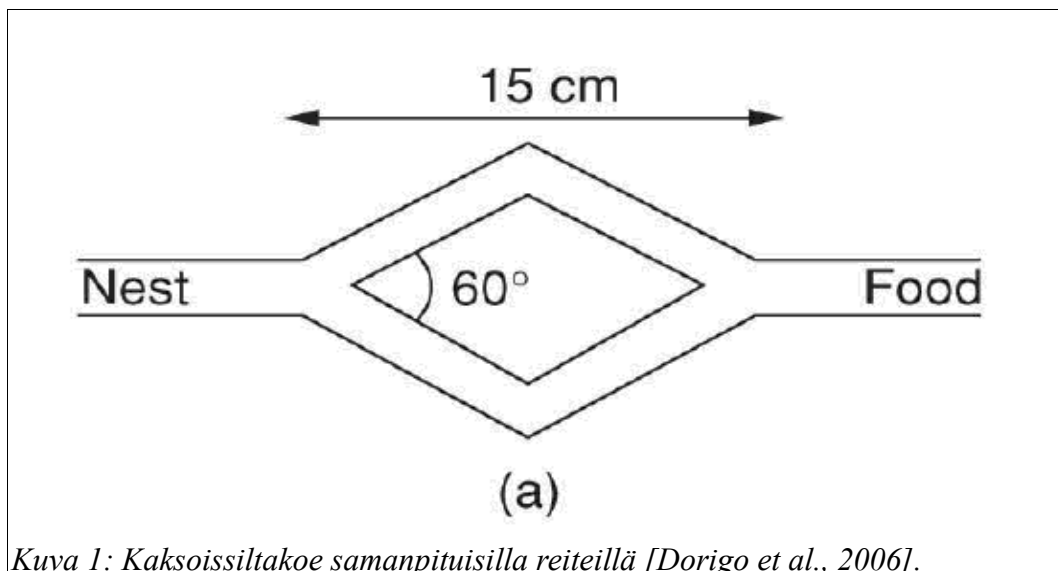
Ranskalainen entomologi Pierre-Paul Grassé [1946] tutki muurahaisten käyttäytymistä ja huomasi, että tietyt termitilajit reagoivat aiemmin tuntemattomaan ärsykkeeseen. Grassé käytti termiä *stigmergia* kuvailemaan tätä ilmiötä. Stigmergia on muurahais-

ten käyttämä kommunikaation muoto, joka eroaa muista kommunikaatiotavoista siten, että

1. Stigmergia on epäsuora, ei-symbolinen kommunikaation muoto, jossa ympäristö toimii viestin välittäjänä. Hyönteiset vaihtavat informaatiota muokkaamalla ympäristöä; ja
2. Stigmerginen informaatio on paikallista: sitä voivat käyttää vain ne muurahaiset, jotka fyysisesti vierailevat paikassa, jossa informaatio välitettiin.

Stigmergiaa voidaan nähdä luonnossa tarkkailemalla, miten muurahaisyhdyskunnat käyttäytyvät ja miten ne etsivät ravintonsa. Etsiessään ravintoa muurahaiset tutkivat pesän ympäristöä satunnaisesti. Liikkuessaan muurahaiset levittävät kulkemalleen polulle kemiallisen feromonijäljen. Muurahaiset pystyvät aistimaan reitille levitetyn feromonin ja tapaavat kulkea reittejä, joilla on korkea feromonipitoisuus. Tämän mekanismin avulla muurahaiset voivat kuljettaa ravintoa pesään tehokkaalla tavalla.

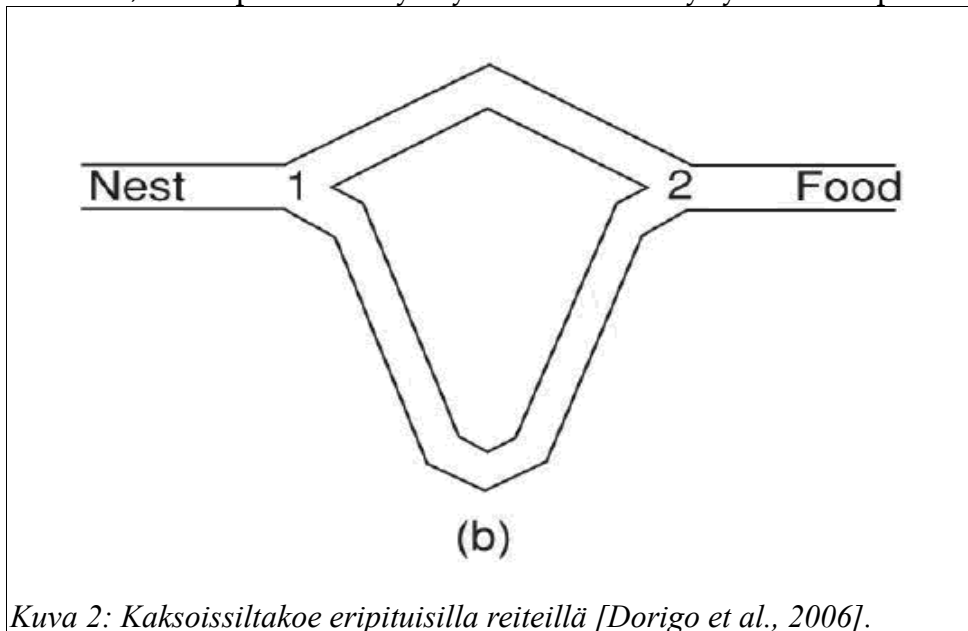
Mekanismia tutkivat kokeellisesti Deneubourg ja muut [1990] järjestämällä niin sanotun kaksoissiltakokeen (double bridge experiment). Kokeessa yhdistettiin yhdyskunta argentiinalaisia muurahaisia ravinnonlähteeseen kahdella yhtä pitkällä sillalla. Koejärjestely on esitelty kuvassa 1.



Muurahaiset tutkivat ympäristöään satunnaisesti ja löytävät lopulta ravinnonlähteen. Muurahaiset levittävät kulkemalleen reitille feromonin, joka saa toiset muurahaiset valitsemaan kyseisen reitin suuremmalla todennäköisyydellä. Vaikka kokeen alussa jokainen muurahainen valitsee satunnaisesti toisen kahdesta reitistä, satunnaisista vaihteluista joh-

tuen toiselle reitille kertyy ajan kuluessa suurempi määrä feromonia, joka houkuttelee reitille entistä enemmän muurahaisia. Muurahaisten määrän kasvu reitillä lisää myös reitille kertyvän feromonin määrää, mikä tekee reitistä yhä todennäköisemmän kulkuvaihtoehdon. Tästä seuraa, että ajan kuluessa koko yhdyskunta konvergoi kohti yhden reitin kulkeamista.

Aiheemme kannalta mielenkiintoisempi tapaus on kuitenkin muunnos äsken mainitusta koeasetelmasta. Goss ja muut [1989] muunsivat koetta siten, että toinen silta on toista huomattavasti pidempi (Kuva 2). Tässä tapauksessa sattumanvaraisella reitinvalinnalla kokeen alkuvaiheessa on pienempi merkitys ja toisen mekanismin merkitys korostuu suuresti. Muurahaiset, jotka valitsevat lyhyemmän reitin, pääsevät ensimmäisenä ravinnonlähteelle ja takaisin pesälle. Tällöin lyhyt reitti kerää feromonia pitkää reittiä aiemmin, mikä taas nostaa muiden muurahaisten todennäköisyyttä valita reitukseen lyhyempi reitti. Tämä tarkoittaa sitä, että lopulta koko yhdyskunta kulkee lyhyttä reittiä pitkin.



Goss ja muut [1989] kehittivät kokeen yhteydessä matemaattisen mallin muurahaisten käytöksestä: olettaen, että hetkellä m_1 muurahaiset ovat kulkeneet ensimmäistä reittiä ja hetkellä m_2 toista reittiä, on muurahaisten todennäköisyys p_1 valita ensimmäinen reitti

$$p_1 = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h}, \quad (1)$$

jossa parametreihin k ja h sijoitetaan arvot kokeellisen datan perusteella.

3. Kombinatorinen optimointi

Kombinatorisella optimoinnilla tarkoitetaan optimoinnin haaraa, jossa käsiteltävän ongelman mahdollisten ratkaisujen joukko on äärellinen. Papadimitrioun ja Steiglitzin [1982] mukaan kombinatorinen optimointiongelma $P=(S, f)$ on ongelma, jossa S on mahdollisten ratkaisujen äärellinen joukko ja f on optimoitava tavoitefunktio $f:S \rightarrow R^+$, joka asettaa jokaiselle ratkaisukandidaatille $s \in S$ kustannusarvon. Optimoinnin tehtävänä on minimoida $f(s)$ eli löytää funktiolle $f(s)$ mahdollisimman pieni arvo siten, että optimointiongelman annettujen rajoitteiden ovat voimassa. Huomautettakoon, että $f(s)$:n minimointi on sama asia kuin $-f(s)$:n maksimointi, joten jokainen kombinatorinen optimointiongelma voidaan kuvata minimointiongelmana.

Monet kombinatorisista optimointiongelmissä ovat NP-kovia (NP-hard = non-deterministic polynomial-time hard), joille ei ole löydetty ongelman koon suhteen polynomisesti rajatussa ajassa toimivaa ratkaisualgoritmia. Tämä tarkoittaa sitä, että ongelman ratkaisemiseen tarvittava aika kasvaa eksponentiaalisesti ongelman koon suhteen [Blum and Roli, 2003].

Ongelmien NP-kovuudesta johtuen optimaalisen ratkaisun löytämiseen tarvittava aika on useissa tapauksissa liian suuri ollakseen käytännöllinen. Tästä johtuen uhrataan usein takuu optimaalisen ratkaisun löytymisestä, jotta saataisiin hyviä ratkaisuja huomattavasti lyhyemmässä ajassa. Tämä toteutetaan käytännössä approksimaatiomenetelmillä, joiden kehittäminen on ollut alati kasvavan huomion kohteena viimeisten vuosikymmenien aikana. Approksimaatioalgoritmit voidaan jakaa karkeasti kahteen ryhmään: rakennusalgoritmeihin ja paikallinen etsintä -algoritmeihin, joita käsitellään seuraavaksi tarkemmin [Blum, 2005].

3.1. Rakennusalgoritmit

Rakennusalgoritmit (construction algorithms) toimivat inkrementaalisesti aloittamalla tyhjästä ratkaisusta ja lisäämällä siihen iteratiivisesti ongelman mukaisella tavalla määriteltyjä ratkaisukomponentteja palaamatta rakennusprosessissaan takaisin kunnes kokonainen ratkaisu on valmis. Lisättävät ratkaisukomponentit voidaan yksinkertaisimmillaan valita satunnaisesti, mutta yleensä on hyödyllisempää hyödyntää valinnassa heuristista arvioita siitä, millainen on ratkaisukomponentista saatava hyöty. Eräs voimakkaasti heuristista informaatiota paikallisissa valinnoissa painottava rakennusalgoritmi on niin kutsuttu ahne rakennusalgoritmi. Algoritmi 1 on pseudokoodin muotoon kirjoitettu listaus ahneesta rakennusalgoritmista. Metodi `ahneestiValittuKomponentti` palauttaa ratkaisukomponentin, jolla on suoritushetkellä paras heuristinen hyötyarvio [Dorigo and Stützle, 2003].

Algoritmi Ahne rakennusalgoritmi

Syötteet: Approksimoitava optimointiongelma

Tuloste: Optimointiongelman ratkaisu

algoritmi Ahne rakennusalgoritmi

s_p = tyhjä ratkaisu

while (s_p ei kokonainen ratkaisu) **do**

e = ahneestiValittuKomponentti(s_p)

$s_p = s_p + e$

end

return s_p

end Ahne rakennusalgoritmi

Algoritmi 1: Algoritminen luuranko ahneesta rakennusalgoritmista

Ahneiden rakennusheuristiikkojen kompastuskivenä on se, että suorituksen aikaisessa vaiheessa heuristisen informaation perusteella tehdyt paikallisesti optimaaliset komponenttivalinnat rajoittavat suuresti mahdollisten kokonaisratkaisujen määrää ja johtavat usein kokonaisuuden kannalta hyvin epäoptimaalisiin valintoihin algoritmin suorituksen loppuvaiheessa [Dorigo and Stützle, 2003].

3.2. Paikallinen etsintä

Paikallinen etsintä on approksimaatiomenetelmä, joka eroaa toimintaperiaatteiltaan suuresti rakennusalgoritmeista. Kun rakennusalgoritmit aloittavat tyhjästä ratkaisusta ja rakentavat siitä kokonaisen lisäämällä iteratiivisesti komponentteja, toimii paikallinen etsintä sen sijaan käsittelemällä kokonaisia ratkaisuja.

Paikallinen etsintä toimii siten, että algoritmi aloittaa valmiista ratkaisusta ja yrittää parantaa sitä käyttämällä avuksi kyseisen ratkaisun ympäristöä. Yksinkertaisena esimerkkinä paikallisesta hausta toimii iteratiivinen parannus -algoritmi, jossa algoritmi etsii ratkaisun ympäristöstä parempaa ratkaisua ja korvaa alkuperäisen löytyneellä, mikäli parempi ratkaisu löytyy. Mikäli tällaista ei löydy, palauttaa algoritmi käsittelyssä olevan ratkaisun, jolloin suoritus päättyy [Dorigo and Stützle, 2003]. Ratkaisutavan periaate on esitetty Algoritmissa 2.

Algoritmi Iteratiivinen parannus($s \in S$)

Syötteet: Optimointiongelman ratkaisu s , joka kuuluu hakuavaruuteen, eli mahdollisten ratkaisujen joukkoon S .

Tuloste: Mahdollisuuksien mukaan paranneltu optimointiongelman ratkaisu s .

algoritmi Iteratiivinen parannus($s \in S$)

$s' = \text{Paranna}(s)$

while $s' \neq s$ **do**

$s = s'$

$s' = \text{Paranna}(s)$

end

return s

end Iteratiivinen parannus

Algoritmi 2: Algoritminen luuranko iteratiivisesta parannusalgoritmista

Ympäristön käsite on oleellinen osa paikallista etsintää ja sen määrittelyyn tuleekin kiinnittää huomiota. Huomioitavaa on myös, ettei ratkaisun ympäristöä voi määrittellä abstraktisti yleisellä tasolla, vaan se on tehtävä ongelmakohtaisesti. Ympäristön käsite määrittelee sen ratkaisujen joukon, joka voidaan saavuttaa nykyisestä ratkaisusta yhden paikallisen etsinnän algoritmin askeleella.

Käytetään Dorigon ja Stützlen [2003] mukaisesti esimerkkinä kauppamatkustajan ongelmaa (TSP = travelling salesman problem) avuksi ympäristön käsitteen määrittelyssä. Paikallisessa etsinnässä perinteinen ympäristömäärittely on nk. *k-opt*-ympäristö, jossa ehdotettu ratkaisu eroaa parannettavasta korkeintaan k :lla komponentilla. TSP:n tapauksessa tarkoittaisi kaupunkien välisiä reittivalintoja, esimerkiksi 2-opt ratkaisussa voitaisiin tarkastella sellaisia naapuriratkaisuja, jotka eroavat parannettavasta enintään kahdella kahden kaupungin välisellä kaarella.

Paikallista etsintää käyttävien algoritmien yleinen ongelma on, että ne jäävät helposti jumiin paikallisiin minimeihin ja niiden tuottama ratkaisu riippuu suuresti alkupe-
räisestä, parannettavasta, ratkaisusta. Paikallisen etsinnän täsmälliseen määrittelyyn tarvittaisiin myös menetelmät, miten ympäristöä tutkitaan ja millä perusteilla ratkaisuehdokas valitaan korvaamaan parannettava [Dorigo and Stützle, 2003]. Näihin ei kuitenkaan tässä yhteydessä perehdytä tarkemmin.

4. ACO-metaheuristiikka

Muurahaisyhdyskuntaoptimoinnin (ACO = Ant Colony Optimization) on muodostanut metaheuristiikaksi Dorigo ja muut [1999]. Metaheuristiikalla tarkoitetaan käytännössä yleisellä tasolla määriteltyä algoritmista viitekehikkoa, jota voidaan soveltaa moniin erilaisiin optimointiongelmiin suhteellisen vähin muutoksin. ACO-metaheuristiikka kattaa useimmat nykyiset kombinatoriseen optimointiin suunnitellut muurahaisyhdyskuntaoptimoinnin muunnokset [Blum, 2005].

Muurahaisyhdyskuntaoptimointia voidaan pitää rakennusalgoritmimenetelmän laajennoksena. Tässä menetelmässä käytetyt keinotekoiset muurahaiset rakentavat ratkaisun iteratiivisesti lisäämällä ratkaisukomponentteja ottaen huomioon:

1. Heuristisen informaation ratkaistavasta ongelmasta; ja
2. Keinotekoiset feromonijäljet, jotka vaikuttavat ajonaikaisesti viestien muurahaisten etsinnän aikana keräämästä tiedosta.

Muurahaisyhdyskuntaoptimoinnin etuja aikaisemmin esiteltyihin kombinatoristen optimointiongelmiin approksimointimenetelmiin nähden on useita. Ensinnäkin tässä menetelmässä läsnä oleva stokastinen komponentti mahdollistaa useiden erilaisten ratkaisujen rakentamisen, jolloin mahdollisia ratkaisuvaihtoehtoja on huomattavasti suurempi määrä kuin ahneilla rakennusalgoritmeilla. Toisekseen usein saatavilla oleva tehtäväkohmainen heuristinen informaatio ohjaa muurahaisia rakentamaan hyvälaatuisia ratkaisuja. Lisäksi ehkä menetelmän tärkeimpänä yksityiskohtana, muurahaiset keräävät etsintänsä aikana tietoa, jota voidaan käyttää ohjaavana tekijänä algoritmin seuraavissa iteraatioissa. Myös muurahaisten yhdyskuntamainen luonne auttaa ratkaisemaan monia ongelmia: kollektiivinen vuorovaikutus eri muurahaisten välillä on usein eduksi mahdollisimman hyvän ratkaisun löytymisen kannalta [Dorigo and Stützle, 2003].

Tässä luvussa esitellään ACO-metaheuristiikka, mutta ensin määritellään ratkaistava ongelma formaalisti ja käsitellään muurahaisten käyttäytymistä tarkemmin.

4.1. Optimointiongelman määritelmä

Voidaksemme soveltaa muurahaisyhdyskuntaoptimointia kombinatoriseen optimointiongelmaan täytyy ongelma ensin määritellä formaalisti. Määritellään Dorigon ja muiden [2006] tapaan kombinatorinen optimointiongelma $P=(S, \Omega, f)$ siten, että

- Hakuavaruus S muodostuu äärellisestä päätösmuuttujien joukosta $X_i, i=1, \dots, n$;
- Ω on joukko rajoituksia, jotka muuttujien tulee täyttää; ja
- tavoitefunktio $f: S \rightarrow R_0^+$ on minimoitava funktio.

Päätösmuuttujien joukko $X_i, i=1, \dots, n$, saa arvoja $v_i^j \in D_i = \{v_i^1, \dots, v_i^{|D_i|}\}$. S :n elementit ovat täysiä sijoituksia, eli sijoituksia, joissa jokaisella päätösmuuttujalla X_i on arvo v_i^j sijoitettuna tämän määrittelyalueesta D_i . Hyväksyttävä ratkaisu $s \in S$ on elementti, joka täyttää kaikki joukon Ω rajoitukset. Ratkaisu s^* on *globaali optimi* jos ja vain jos se täyttää ehdon $f(s^*) \leq f(s)$ kaikilla alkioilla $s \in S$.

Tätä esitystä hyödyntäen määritellään muurahaisyhdyskuntaoptimoinnin *feromonimalli*: jokaiselle mahdolliselle ratkaisukomponentille asetetaan feromoniarvo, eli asetus tehdään jokaiselle mahdolliselle muuttujan arvosijoitukselle. Tarkemmin: feromoniarvo $\tau_{i,j}$ liitetään ratkaisukomponenttiin $c_{i,j}$, joka koostuu sijoituksesta $X_i = v_i^j$. Kaikkien mahdollisten ratkaisukomponenttien joukkoa merkitään C .

Keinotekoiset muurahaiset rakentavat ratkaisut liikkumalla täysin yhdistettyä rakennegraafia $G_C(V, E)$ pitkin, jossa V on solmujen ja E kaarien joukko. Rakennegraafi saadaan ratkaisukomponenttien joukosta C kuvaamalla komponentteja joko graafin solmuina tai kaarina. Muurahaiset liikkuvat solmusta toiseen graafin kaaria pitkin ja jättävät ratkaisukomponenteille, yleensä kaarille, tietyn määrän feromonia. Tämän jälkeen seuraavat muurahaiset käyttävät feromonia ohjausmekanismina kohti hyviä reittejä [Dorigo *et al.*, 2006]. Ongelman rajoitteet joukossa Ω toteutetaan muurahaisten käyttäytymismekanismeilla, jota käsitellään seuraavassa kohdassa.

4.2. Keinotekoisten muurahaisten käyttäytymismekanismi

Kuten Dorigo ja Stützle [2003] ilmaisevat, keinotekoiset muurahaiset ovat eräänlaisia stokastisia rakennusmetodeita, jotka rakentavat ratkaisuja liikkueessaan rakennegraafilla $G=(V, E)$. Liikkumisella graafilla tarkoitetaan tässä yhteydessä sitä, että muurahaiset rakentavat ratkaisuja ongelman rajoitteiden joukon Ω määrittämän funktion mukaisesti. Muurahaiset tallentavat rakennusprosessissa saavutettua informaatiota eräällä tavalla algoritmin muistiin käyttäen feromonijälkiä. Feromonijäljet liitetään ratkaisukomponentteihin, siis yleensä rakennegraafin kaariin. Solmuun i liittyvästä feromonijäljestä käytetään merkintää τ_i ja solmuja i ja j yhdistävään kaareen liittyvästä feromonijäljestä merkintää $\tau_{i,j}$. Lisäksi ratkaisun rakentamisessa voidaan käyttää heuristista arvoa, jota käytetään merkitsemään *a priori* -tietoa ongelmasta, tai ajonaikaisesti saatua tietoa, joka ei ole muurahaistilta peräisin. Tästä arvosta käytetään merkintää η_i , jos se liittyy solmuun i , ja merkintää $\eta_{i,j}$, jos se liittyy solmut i ja j yhdistävään kaareen.

Määritellään seuraavaksi muurahaisten ominaisuudet tarkemmin. Jokaisella yhdyskunnan muurahaisella k on seuraavat ominaisuudet:

- Muurahainen käyttää graafia $G=(V, E)$ etsiäkseen hyväksyttävää ratkaisua s , jonka kustannus on pienin mahdollinen.
- Muurahaisella on muisti M^k , johon se tallentaa informaatiota kulkemastaan polusta. Muistia voidaan käyttää:
 1. hyväksyttävien ratkaisujen rakentamiseen, eli rajoitteiden Ω täyttämiseen,
 2. löydetyn ratkaisun laadun arvioimiseen ja
 3. kuljetun reitin peruuttamiseen feromonin asettamista varten.
- Muurahaisella voi olla alkutila x_s^k ja yksi tai useampia lopetusehtoja e^k .
- Muurahainen valitsee reittinsä soveltamalla todennäköisyyksiin perustuvaa päätös-sääntöä. Tämä sääntö on funktio:
 1. paikallisesti saatavilla olevista feromonijäljistä ja heuristisesta informaatiosta;
 2. muurahaisen omasta muistista, johon on tallennettu tämän kulkuhistoriaa; ja
 3. optimointiongelman rajoitteista.
- Muurahaisen k rakennustehtävä päättyy, kun vähintään yksi lopetusehdoista e^k täyttyy.
- Kun muurahainen lisää ratkaisukomponenttia c_{ij} , se voi päivittää kyseiseen komponenttiin asetetun feromonijäljen.
- Kun muurahainen on rakentanut ratkaisun, se voi palata kulkemaansa reittiä takaisin ja päivittää läpikäymiinsä komponentteihin liittyvät feromonijäljet.

Kuten Dorigo ja Stützle [2003] toteavat, muurahaiset toimivat rinnakkaisesti ja toisiinsa nähden itsenäisesti sekä jokainen muurahainen pystyy rakentamaan ratkaisun myös yksin. Hyvien ratkaisujen saavuttamiseksi on kuitenkin oleellista, että hyödynnetään yhdyskuntaa kokonaisuutena. Feromonijälkien avulla muurahaiset pystyvät kommunikoimaan epäsuorasti keskenään, jolloin reittivalintaan vaikuttaa yhdyskunnan muiden jäsenten toiminta. Useampia muurahaisia käyttämällä saavutetaankin monesti parempia tuloksia, kun feromonimallin merkitys korostuu.

4.3. Metaheuristiikka

Muurahaisyhdyskuntaoptimoinnin lähestymistapa optimointiongelmaan on käytännössä kahden vaiheen iterointi:

1. Ratkaisuehdokkaat rakennetaan käyttäen feromonimallia T , joka on ongelmaan ja sen komponentteihin liitettyjen feromoniarvojen joukko.
2. Feromonimallia muokataan ratkaisuehdokkaita käyttämällä siten, että algoritmin tulevat iteraatiot painottuvat kohti hyviä ratkaisuja.

Algoritmissa 3 esitellään muurahaisyhdyskuntaoptimoinnin metaheuristiikka pseudokoodin muodossa, Dorigon ja Stützlen [2003] hahmottelemana.

```

Algoritmi MuurahaisYhdyskuntaOptimointi
  while (lopetusehtoja ei voimassa) do
    aikataulutaToiminnot
      hallitseMuurahaisia()
      päivitäFeromonijäljet()
      daemonActions() // valinnainen
    end aikataulutaToiminnot
  end while
end MuurahaisYhdyskuntaOptimointi

```

Algoritmi 3: Algoritminen esitys ACO-metaheuristiikasta

Käydään algoritmia läpi tarkemmin. Algoritmin aikaTaulutaToiminnot-metodia ei ole määritelty metaheuristiikassa tarkemmin. On algoritmin suunnittelijan tai implementoijan vastuulla, miten kyseiset toiminnot aikataulutetaan ja sovitetaan toimimaan keskenään.

Metodi hallitseMuurahaisia() pitää sisällään ratkaisun rakentamisvaiheen. Siinä m kappaletta muurahaisia rakentaa ratkaisuja saatavilla olevista ratkaisukomponenteista joukosta $C = \{c_{ij}\}$, $i=1, \dots, n$, $j=1, \dots, |D_j|$. Ratkaisun rakentaminen alkaa tyhjästä osittaisratkaisusta $s^p = \emptyset$ ja tähän osittaisratkaisuun lisätään jokaisessa rakennusvaiheessa hyväksyttävä ratkaisukomponentti joukosta $N(s^p) \subseteq C$. Joukko $N(s^p)$ määritellään sellaisten ratkaisukomponenttien joukkona, jotka voidaan lisätä ratkaisuehdokkaaseen s^p rikkomatta asetettuja rajoitteita Ω [Dorigo *et al.*, 2006].

Ratkaisukomponentin valinta joukosta $N(s^p)$ suoritetaan stokastisen mekanismin avulla, jossa valintaa ohjaa komponenttiin liitetty feromonijälki. Prosessin tilaan vaikuttavat sekä ennalta arvioitava informaatio että satunnainen elementti. Blumin [2005] mukaan useimmissa ACO-algoritmeissa tämä todennäköisyyksiin perustuva mekanismi määritellään muodossa

$$p(c_{ij}|s^p) = \frac{[\tau_{ij}]^\alpha * [\eta(c_{ij})]^\beta}{\sum_{c_{il} \in N(s^p)} [\tau_{il}]^\alpha * [\eta(c_{il})]^\beta}, \quad \forall c_{ij} \in N(s^p), \quad (2)$$

jossa η on valinnainen painotusfunktio. Sen arvot ovat heuristista informaatiota, jolla voidaan painottaa ratkaisukomponentin valintaa tiettyyn suuntaan. Eksponentit α ja β ovat positiivisia parametrejä, joilla voidaan säätää, missä suhteessa feromoniarvojen ja heuristisen informaation hyödyntäminen ovat ratkaisujen rakentamisessa. Symbolit t_{ij} ja $\eta(c_{ij})$ merkitsevät ratkaisukomponentille liitettyjä feromoniarvoa ja heuristista informaatiota.

Metodi päivitäFeromoniJäljet() sisältää feromoniarvojen päivityssäännön. Feromonijälkien päivityksen tarkoituksena on kasvattaa arvoja, jotka liittyvät hyviin ratkaisuihin ja vähentää niitä arvoja, jotka liittyvät huonoihin ratkaisuihin. Blumin [2005] mukaan päivitys koostuu kahdesta osasta:

1. Feromonin haihtuminen, joka vähentää kaikkien komponenttien feromonijälkiä. Tämä takaa sen, ettei algoritmi lähde aikaisessa vaiheessa konvergoimaan kohti epäoptimaalisia ratkaisuja. Näin ei suljeta reittejä pois liian aikaisessa vaiheessa.
2. Yhtä tai useampaa ratkaisua algoritmin aiemmista iteraatioista käytetään kasvattamaan tiettyihin ratkaisuihin kuuluvien komponenttien feromonijälkiä.

Feromoniarvojen päivityksessä käytetään sääntöä

$$\tau_{ij} = (1-p) * \tau_{ij} + p * \sum_{\{s \in S_{upd} | c_{ij} \in s\}} w_s * F(s), \quad (3)$$

jossa $i=1, \dots, n$. Merkinnällä S_{upd} tarkoitetaan niiden ratkaisujen joukkoa, jota käytetään päivityksessä. Parametri $p \in]0,1]$ merkitsee feromonin haihtumistahtia. Funktio $F(s)$ on sellainen laatufunktio, että ehdosta $f(s) < f(s')$ seuraa $F(s) \geq F(s')$, kun $s \neq s'$ eli jos tavoitefunktion arvo ratkaisusta s on parempi kuin tavoitefunktion arvo ratkaisusta s' , niin ratkaisun s laatu on vähintään yhtä hyväkuin ratkaisun s' laatu. Parametri $w_s \in R^+$ on valinnainen, ja sitä voidaan käyttää laatufunktion painottamiseen.

Valinnainen daemonActions()-metodi on tarkoitettu muurahaisten ulkopuolisen toiminnan hallitsemiseen. Metodia käytetään usein muurahaisten rakentaman ratkaisun parantamiseen paikallista etsintää käyttäen. Toisena esimerkkinä tällä metodilla hallittavasta toiminnasta voitaisiin kerätä globaalia, muurahaisten ulottumattomissa olevaa tietoa ja käyttää sitä etsintäprosessin ohjaamiseen tiettyyn suuntaan. Käytännössä voitaisiin esimerkiksi kasvattaa edelleen tämän hetkiseen parhaaseen ratkaisuun liittyvien komponenttien feromoniarvoja [Blum, 2005].

5. ACO-algoritmit kauppamatkustajan ongelman ratkaisuun

Tässä luvussa käsitellään erilaisia muurahaisyhdyskuntaoptimointialgoritmeja kauppamatkustajan ongelman ratkaisemiseksi, joka oli myös ensimmäinen ACO-algoritmin sovellus [Dorigo *et al.*, 1999]. Kyseisen ongelma on hyvä testitapaus muurahaisyhdyskuntaoptimoinnille monestakin syystä. Ensinnäkin se on NP-kova kombinatorinen optimointiongelma, eli sen ratkaisu approksimoimalla on järkevä lähestymistapa. Toisekseen siinä etsitään lyhintä polkua, mikä tekee muurahaisyhdyskuntametaforasta helposti ymmärrettävän käytännön sovelluksessa. Lisäksi kauppamatkustajan ongelma on didaktinen ongelma, mikä merkitsee sitä, että ongelman periaatteet on helppo ymmärtää ja se sopii hyvin erilaisten ratkaisumenetelmien demonstrointiin.

Kauppamatkustajan ongelma voidaan määritellä seuraavalla tavalla. On olemassa graafi G , joka koostuu solmujen joukosta N ja kaarien joukosta E , missä N merkitsee kaupunkien ja E kaupunkien välisten reittien joukkoja. Kaupunkien i ja j , missä $i, j \in N$, välistä etäisyyttä merkitään d_{ij} , joka on kaaren $(i, j) \in E$ pituus. Tehtävänä on etsiä graafista $G=(N, E)$ lyhin mahdollinen polku, jossa käydään jokaisessa kaupungissa täsmälleen yhden kerran. Polun pituus on kaikkien sen sisältämien kaarien painojen summa. Tällaista polkua kutsutaan nimellä *Hamiltonin polku*.

5.1. Ant System

Ant System -algoritmi oli ensimmäinen kirjallisuudessa esitelty ACO-algoritmi [Dorigo, 1992]. Sen pääpiirre on, että jokainen kierroksensa päättänyt muurahainen päivittää feromoniarvot kaarille, joilla tämä on käynyt.

Algoritmit toimii pääpiirteittäin seuraavalla tavalla. Jokainen muurahainen generoi kokonaisen polun valitsemalla kaupungit käyttäen hyväksi todennäköisyyksiin perustuvaa *tilasiirtymäsääntöä* (state transition rule), jossa muurahaiset suosivat kaaria, joilla on pieni paino ja suuri feromoniarvo. Kun kaikki muurahaiset ovat päättäneet kierroksensa, päivitetään kaariin liitetyt feromoniarvot käyttäen *globaalia feromonien päivityssääntöä*: feromoniamäärä vähennetään ensin jokaiselta kaarelta, minkä jälkeen jokainen muurahainen asettaa vieraillemilleen kaarille feromonimäärän, joka on suhteessa löydetyn kokonaispolun pituuteen. Tällöin kaaret, jotka kuuluvat hyviin ratkaisuihin, saavat suuremman määrän feromoniamäärää kuin muut kaaret, jolloin niitä suositaan algoritmin seuraavilla iteraatioilla [Dorigo and Gambardelle, 1997b].

Kun muurahaiset rakentavat ratkaisua, ne kulkevat graafilla ja tekevät todennäköisyyksiin perustuvan päätöksen seuraavan solmun valinnasta. Kun muurahainen k on kaupungissa i , sen todennäköisyys siirtyä kaupunkiin j saadaan tilasiirtymäsäännöllä:

$$p(c_{ij}|s_k^p) = \begin{cases} \frac{[\tau_{ij}]^\alpha * [\eta(c_{ij})]^\beta}{\sum_{c_{il} \in N(s_k^p)} [\tau_{il}]^\alpha * [\eta(c_{il})]^\beta}, & \text{jos } j \in N(s_k^p) \\ 0 & \text{, muutoin} \end{cases}, \quad (4)$$

jossa $N(s_k^p)$ on niiden komponenttien joukko, jotka eivät vielä kuulu muurahaisen k rakentamaan osaratkaisuun s_k^p ja kaareen c_{ij} liitettyä heuristista informaatiota merkitsevä funktio $\eta(c_{ij})$ on $1/d_{ij}$, jossa d_{ij} on kaaren c_{ij} paino.

Kun jokainen muurahainen on päättänyt kierroksensa, päivitetään jokaiseen kaareen liitetty feromoniarvo. Globaali feromonien päivityssääntö on muotoa

$$\tau_{ij} \leftarrow (1-p) * \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k, \quad (5)$$

jossa $p \in]0,1]$ on feromonin haihtumistahti ja $\Delta \tau_{ij}^k$ k :nnen muurahaisen kaarelle (i, j) lisäämä feromonimäärä. Feromonin haihtumistahti ei voi olla nolla, sillä muuten siirtymäsäännössä seuraa nollalla jakaminen. Määritellään lisättävä feromonimäärä seuraavasti:

$$\Delta \tau_{ij}^k = \begin{cases} \frac{1}{L_k}, & \text{jos muurahainen } k \text{ vieraili kaarella } (i, j) \\ 0 & \text{, muutoin} \end{cases}, \quad (6)$$

jossa L_k on k :nnen muurahaisen löytämän ratkaisun kokonaispituus.

Kuten Dorigo ja Gambardelle [1997b] toteavat, feromonien päivityssäännöstä seuraa se, että kaarille, jotka kuuluvat lyhyisiin polkuihin, kertyy suurempi määrä feromoniamäärää kuin muille kaarille. Sääntö simuloi feromonimäärän muutosta, joka aiheutuu paitsi feromonin haihtumisesta myös muurahaisten tuottaman uuden feromonin lisäyksestä. Tällä tavalla algoritmissa toteutetaan stigmerginen kommunikaatio muurahaisten välillä reittien laadusta.

Ant System -algoritmi löytää hyviä tai optimaalisia ratkaisuja kauppatkustajan ongelmaan, jos käsitellään vain pientä määrää kaupunkia (alle 30). Suurempien ongelmataustojen kohdalla ratkaisuiden löytämiseen tarvittava aika kasvaa kuitenkin nopeasti epäkäytännölliseksi. Kun algoritmia vertaa ahneeseen rakennusalgoritmiin, huomataan, että oikeilla parametreilla saadaan parempia ratkaisuja ja ratkaisut ovat paljon monipuolisempia kuin ahneen rakennusalgoritmin kanssa. Suurilla kaupunkimäärillä Ant System -algoritmi kuitenkin kuluttaa epäkäytännöllisen paljon aikaa, kun taas ahne rakennusalgoritmi suoriutuu tehtävästään hyvinkin nopeasti, joskin tuottaen epäoptimaalisen ratkaisun.

5.2. MAX – MIN Ant System

MAX – MIN Ant System -algoritmi on Stützlen ja Hoosin [2000] esittelemä parannus Ant System -algoritmiin. Se eroaa alkuperäisestä algoritmista kolmella tavalla:

1. Parhaita löytyneitä ratkaisuja korostetaan siten, että jokaisella iteraatiolla vain yksi muurahainen lisää feromonia. Kyseinen muurahainen voi olla esimerkiksi sellainen, joka on löytänyt parhaan ratkaisun kyseisessä iteraatiossa tai joka on löytänyt parhaan tähän asti löydetyn ratkaisun.
2. Koska parhaita ratkaisuja korostetaan voimakkaasti, asetetaan liian aikaisen konvergenssin estämiseksi feromoniarvoille eksplisiittiset rajat välille $[\tau_{min}, \tau_{max}]$.
3. Lisäksi feromonijäljet alustetaan arvoon τ_{max} , tavoitteena korostaa uusien ratkaisujen etsimistä algoritmin ajon alkuvaiheessa.

Tällöin feromonien päivityssäntö saadaan muotoon

$$\tau_{ij} \leftarrow (1-p) * \tau_{ij} + \Delta \tau_{ij}^{\text{best}}, \quad (7)$$

jossa $\Delta \tau_{ij}^{\text{best}} = 1/L_{\text{best}}$, jos parhaan reitin löytänyt muurahainen käytti kaarta (i, j) kieroksellaan ja $\Delta \tau_{ij}^{\text{best}} = 0$ muutoin. L_{best} on parhaan löydetyn reitin kokonaispituus. Parhaana reittinä voidaan käyttää joko iteraation parasta reittiä, parasta tähän asti löydettyä reittiä tai molempien yhdistelmää.

Dorigon ja muiden [2006] mukaan feromoniarvojen rajat määritetään usein empiirisesti ja hienosäädetään käsiteltävän ongelman mukaisesti. Stützle ja Hoos [2000] esittelevät kuitenkin myös analyttisiä menetelmiä rajojen määrittämiseen. Maksimiarvo τ_{max} voidaan laskea, mikäli optimaalisen polun pituus on tiedetty. Kauppatkustajan ongelman ollessa kyseessä $\tau_{max} = 1/(p * L^*)$, jossa L^* on optimaalisen polun pituus. Mikäli L^* ei ole tiedossa, voidaan sitä approksimoida käyttäen parhaan löydetyn polun pituutta.

5.3. Ant Colony System

Ant Colony System -algoritmi oli ensimmäinen merkittävä parannus Ant Systemiin. Sen esittelivät kirjallisuudessa Dorigo ja Gambardelle [1997b]. Kuten MAX – MIN Ant System, tämäkin algoritmi eroaa alkuperäisestä algoritmista pääasiassa kolmella tavalla.

Ensinnäkin tilasiirtymäsääntöön on otettu mukaan satunnaismuuttuja q , jonka arvot jakautuvat tasaisesti välille $[0,1]$, ja parametri q_0 , joka määrittää suhteellisen painotuksen uusien ratkaisujen etsimisen ja olemassa olevan informaation hyödyntämisen välillä. Jos $q \leq q_0$, niin muurahainen valitsee seuraavaksi kohteekseen kaupungin j säännöllä

$$j = \arg \max_{c_{i,l} \in N(s^p)} \{ \tau_{i,l} * \eta(c_{i,l})^\beta \} . \quad (8)$$

Jos taas $q > q_0$, niin käytetään Ant System -algoritmin sääntöä. Tällaista tilasiirtymäsääntöä kutsutaan *pseudosatunnaiseksi suhteellisuussäännöksi* (pseudo-random proportional rule) [Dorigo and Gambardelle, 1997b].

Toisena eroavaisuutena globaalia feromonien päivityssääntöä sovelletaan ainoastaan kaariin, jotka kuuluvat parhaaseen löytyneeseen ratkaisuun. Yhdessä uuden tilasiirtymäsäännön kanssa tästä seuraa se, että ratkaisujen etsinnästä tulee ohjatumpaa: muurahaiset etsivät ratkaisuja parhaan tähän asti löydetyn ratkaisun ympäristöstä. Tässä, kuten *MAX-MIN* Ant Systemissä, paras ratkaisu voi olla paras iteraatiossa löydetty ratkaisu tai paras tähän asti löydetty ratkaisu. Päivityssääntö

$$\tau_{i,j} \leftarrow (1-p) * \tau_{i,j} + p * \Delta \tau_{i,j}^{\text{best}} , \quad (9)$$

on kuitenkin hieman erilainen. Tässä, kuten MMAS:ssä, asetetaan $\Delta \tau_{i,j}^{\text{best}} = 1/L_{\text{best}}$, jos parhaan reitin löytänyt muurahainen käytti kaarta (i, j) kierroksellaan ja $\Delta \tau_{i,j}^{\text{best}} = 0$ muutoin. L_{best} on parhaan löydetyn reitin kokonaispituus. Parhaana reittinä voidaan käyttää joko iteraation parasta reittiä, parasta tähän asti löydettyä reittiä tai molempien yhdistelmää. Tällaista feromonipäivitystä kutsutaan *erilliseksi feromonien päivitykseksi* (offline pheromone update) erotukseksi kolmannesta Ant Colony System -algoritmin eroavaisuudesta aiempiin nähden eli *paikallisesta feromonien päivityksestä*.

Paikallinen feromonien päivitys on ehkä kiinnostavin lisäys algoritmiin. Siinä jokainen muurahainen lisää feromonia kaareen, jossa vieraili viimeksi jokaisen rakennusaskelen jälkeen [Dorigo and Gambardelle, 1997b]. Paikallinen päivityssääntö on nyt muotoa

$$\tau_{i,j} = (1-p) * \tau_{i,j} + p * \tau_0 , \quad (10)$$

jossa $\tau_{i,j}$ on feromonille alustettu arvo. Paikallisen feromoniarvojen päivityksen tarkoituksena on monipuolistaa samalla iteraatiolla seuraavien muurahaisten suorittamaa etsintää. Kun kaarien feromoniarvoja vähennetään etsinnän aikana, valitsevat saman iteraation toiset muurahaiset todennäköisemmin muita kaaria ja tuottavat täten erilaisia ratkaisuja [Dorigo and Gambardelle, 1997b].

Dorigo ja Gambardelle [1997a] ovat tutkineet Ant Colony System -algoritmin tehokkuutta verrattuna muihin tunnettuihin algoritmeihin. Tulokset ovat olleet ACS:n kannalta rohkaisevia. Esimerkiksi symmetrisen kauppamatkustajan ongelman tapauksessa, jossa kaupunkien määräksi oli asetettu 50 ja verrattiin saatuja keskimääräisiä polkujen pituuksia, ACS-algoritmi menestyi yhtä lukuun ottamatta kaikissa testitapauksissa kilpailijoitaan

paremmin. Vertailtavina olivat simulated annealing, farthest insertion, elastic net ja self organizing map -algoritmit.

6. Teoreettista taustaa – konvergenssi ja aikakompleksisuus

Muurahaisyhdyskuntaoptimoinnissa, toisin kuin esimerkiksi satunnaismekanismilla toimivissa stokastisissa etsintämenetelmissä, on täysin mahdollista, että optimaalinen ratkaisu suljetaan hakuprosessin aikana pois valintamekanismin ja feromoniarvojen yhteisvaikutuksen vuoksi. Tästä syystä on kiinnostavaa tutkia algoritmien konvergenssia voidaksemme varmistua siitä, että näin ei tapahdu.

Konvergenssi kohti optimaalista ratkaisua on geneerisen ACO-algoritmin ollessa kyseessä avoin ongelma. Joissain tapauksissa, kuten Stützlen ja Hoosin [2006] *MAX – MIN Ant System*issä, voidaan olla varmoja siitä, että optimaalinen ratkaisu ei muutu saavuttamattomaksi algoritmin suorituksen aikana: feromoniarvoille asetettu alaraja saa aikaan sen, että minkään siirtymän todennäköisyydestä ei tule nollaa. Tällöin kaikki ratkaisut säilyvät saavutettavissa [Dorigo and Stützle, 2003].

Gutjahr [2000] on todistanut konvergenssin kohti optimaalista ratkaisua muurahaisyhdyskuntaoptimointialgoritmile nimeltä Graph-based Ant System. Tämä algoritmi muistuttaa Ant System -algoritmia monessa suhteessa: käytännössä ainoa ero näiden kahden algoritmin välillä on se, että Gutjahrin algoritmista asetetaan ylimääräisiä rajoitteita feromoniarvojen päivitykselle. Gutjahrin konvergenssitodistuksen mukaan, kun $\epsilon > 0$ ja algoritmin parametreille on asetettu jotkut määrätyt arvot, iteraatiomäärän $t \geq t_0$ jälkeen algoritmi löytää optimaalisen ratkaisun todennäköisyydellä $P_t \geq 1 - \epsilon$, jossa $t_0 = f(\epsilon)$. Tämä tärkeä tulos voi johtaa myös muiden ACO-algoritmien konvergenssitodistuksiin [Dorigo and Stützle, 2003].

Myös muurahaisyhdyskuntaoptimoinnin aikakompleksisuusanalyysi on yksi aiheen avoimista ongelmista. Konvergenssitodistukset eivät kerro mitään optimaalisen ratkaisun löytämiseen tarvittavasta ajasta; se voi olla astronomisen suuri. Hao ja muut [2006] ovat esitelleet kaksi ACO-algoritmia aikakompleksisuusanalyysiin. Kyseiset algoritmit ovat yhden muurahaisen algoritmeja, joilla ratkaistaan totuusarvosyötteitä saava lineaarifunktio, jolloin päätösmuuttujat saavat siis arvoja $[0,1]$. Eroavaisuus näiden kahden algoritmin välillä on se, että vain toisessa otetaan huomioon ongelmaan liitetty heuristinen informaatio. Analyysin perusteella kyseiseen ongelmatapaukseen sovellettu algoritmi löytää optimaalisen ratkaisun aikakompleksisuudella $O(n^k)$, mikäli heuristista informaatiota ei oteta huomioon, ja $O(n^2)$, kun heuristinen informaatio huomioidaan. Tämä todistaa heuristisen informaation hyödyllisyyden sovellettaessa muurahaisyhdyskuntaoptimointia lineaaristen funktioiden ongelmiin.

7. Muita ACO-algoritmeja ja kehitysnäkymiä

Kuten on aiemmin todettu, muurahaisyhdyskuntaoptimointi on alunperin tarkoitettu ap-proksimointimenetelmäksi kombinatoristen optimointiongelmiin ratkaisuuksiin ja käsitellyt ongelmat ovat olleet luonteeltaan staattisia. Muurahaisyhdyskuntaoptimointia on kuitenkin pyritty soveltamaan alati laajemmalti. Esimerkkejä tällaisista sovelluksista ovat jatkuvien optimointiongelmiin ratkaisemiseksi tarkoitettut ACO -algoritmit, muiden tekniikoiden, kuten tekoälyn kanssa toimivat hybridialgoritmit sekä dynaamisessa ympäristössä toimivat tietoverkkojen reititykseen suunnitellut ACO -algoritmit.

Reitityksellä tarkoitetaan reitin valitsemista tietoverkossa algoritmisiin menetelmiin. Tietoliikenne ohjataan kulkemaan tietoliikenneverkossa reittiä, joka kuluttaa vähiten resursseja. Kun tietoverkko esitetään graafina, voidaan reititysongelma esittää lyhimmän kahden solmun välisen polun etsimisenä. Tähän tarkoitukseen on kehitetty esimerkiksi muurahaisyhdyskuntaoptimointialgoritmi AntNet. On huomattava, että vaikka kahden solmun välisen minimipainoisen polun etsiminen on helposti ratkaistava ongelma, ovat kaarien painot tietoverkkojen tapauksessa ajan mukaan muuttuvia stokastisia muuttujia [Dorigo and Stützle, 2003], mikä tekee ongelmasta huomattavasti vaikeamman. AntNet-algoritmia on verrattu moniin parhaisiin tunnettuihin reititysalgoritmeihin erilaisissa simulaatioissa ja sen on osoitettu olevan tehokas ratkaisu vertailtaviansa nähden. Algoritmin on osoitettu olevan myös hyvin joustava suhteessa erilaisiin liikenneolosuhteisiin.

Monet käytännön optimointiongelmiin voidaan mallintaa jatkuvina optimointiongelmiin [Blum, 2005]. Tällaisille ongelmille on tyypillistä, että päätösmuuttujilla on jatkuvat alueet. Kuten aiemmin on todettu, ACO-algoritmeissa kombinatoristen optimointiongelmiin ratkaisuuksiin ratkaisut rakennetaan käyttäen jokaisessa rakennusvaiheessa diskreettiä, feromoniarvoista ja heuristisesta informaatiosta saatavaa todennäköisyysjakaumaa. Jatkuvien optimointiongelmiin ratkaisemiseksi tarkoitetuissa ACO-algoritmeissa sen sijaan hyödynnetään tiheysfunktioita. Tiheysfunktio saadaan ratkaisuiden populaatiosta, jota algoritmi ylläpitää suorituksen ajan. Tällaisia ACO-algoritmeja on onnistuneesti sovellettu sekä akateemisiin testitapauksiin, että käytännön ongelmiin [Blum, 2005].

Muurahaisyhdyskuntaoptimointialgoritmien hybridisointi muiden tekniikoiden kanssa on suhteellisen uusi aihe ACO:ta koskevassa kirjallisuudessa. Blumin [2005] mukaan hybridisointi tunnustetaan nykyisin oleelliseksi osaksi suorituskykyisiä algoritmeja: hybridisoidut algoritmit ovat lähes aina puhtaita vastineitaan suorituskykyisempiä. Ensimmäiset ACO-hybridit olivat algoritmeja, joihin otettiin mukaan paikalliseen etsintään perustuvia tekniikoita. Tällaiset lähestymistavat saavuttavat kuitenkin rajansa nopeasti, kun ratkaistavana olevan ongelmainstanssin koko kasvaa tai hyväksyttävälle ratkaisulle asetetut rajoitteet kiristyvät. Tästä syystä hybridisointikeskustelun keskiöön on asettunut perinteisempien tekoälytekniikoiden ja operaatiotutkimuksen menetelmien sulauttaminen

muurahaisyhdyskuntaoptimointiin. Yksi syy tähän on ACO-algoritmien soveltuvuus tämänkaltaiseen hybridisointiin: ACO-algoritmit rakentavat ratkaisut kartoittamalla hakuavaruutta puurakenteeksi, jossa polku juurisolmusta lehteen vastaa rakennusprosessia [Blum, 2005]. Nykyisten ACO-hybridialgoritmien toiminnan tavoitteena on yrittää rajoittaa algoritmin tutkittavana olevaa hakuavaruutta erilaisilla tekniikoilla, esimerkiksi sädehakualgoritmeilla (beam search).

Keskustelu muurahaisyhdyskuntaoptimoinnin saralla on hyvinkin eloisaa. Viime vuosina aiheen saralta tehty tutkimus on jakautunut käytännössä kahtia. Teoreettisen tutkimuksen osalta tutkijat ovat yrittäneet joko laajentaa nykyisiä tuloksia koskemaan laajempaa aluetta tai etsiä perusteltuja tapoja asettaa arvoja parametreille. Kokeellisen tutkimuksen osalta taas suurin osa tutkimuksesta on keskittynyt kasvattamaan ACO -algoritmeilla ratkaistujen ongelmien määrää, mukaan lukien käytännön sovellukset [Dorigo and Stützle, 2003].

8. Lopuksi

Kombinatoriset optimointiongelmat ovat usein epäkäytännöllisiä ratkaista optimaalisen ratkaisun takaavilla menetelmillä. Tästä syystä niistä ratkaistaan useimmiten approksimointimenetelmillä, joiden joukkoon myös muurahaisyhdyskuntaoptimointi kuuluu. Muurahaisyhdyskuntaoptimointi tarjoaakin hyväksi todetun menetelmän kaikenlaisten kombinatoristen optimointiongelmiin ratkaisuuksi.

Kuten aiemmin on todettu, käytännön sovelluksista hybridialgoritmit ovat tutkimussuuntaus, joka todennäköisesti tarjoaa hyviä mahdollisuuksia muurahaisalgoritmien kehitykseen. Toinen tärkeä käytännön sovelluksia koskeva suuntaus on menetelmällä ratkaistujen ongelmien kirjon laajentaminen entisestään. Myös muurahaisyhdyskuntaoptimoinnin soveltaminen dynaamisiin ongelmiin ja useiden tavoitteiden ongelmiin tulee todennäköisesti olemaan keskeinen suuntaus lähitulevaisuuden tutkimuksessa [Dorigo *et al.*, 2006].

Parempi ymmärrys muurahaisyhdyskuntaoptimoinnin teoreettisesta pohjasta on tarpeellista. Konvergenssin todistamisen eteen menetelmän eri variaatioille on tehty paljon työtä ja aiheita koskevia julkaisuja nähdään varmasti vastaisuudessakin. Myös aika-kompleksisuusanalyysijä koskevaa tutkimusta on tekeillä.

Muurahaisten ravinnonetsintä optimointialgoritmien suunnittelun lähtökohtana tuntuu ensikuulemalta omituiselta idealta. On kuitenkin mielenkiintoista, miten 1990-luvun alkupuolelta nykyhetkeen tultaessa kyseisen menetelmä on muuttunut kurioositeetista tärkeäksi optimointiteorian alaksi.

Viiteluettelo

- [Blum, 2009] Christian Blum, Ant colony optimization. In: *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*.
- [Blum, 2005] Christian Blum, Ant colony optimization: introduction and recent trends. *Physics of Life Reviews* **2** (Oct. 2005), 353–373.
- [Blum and Roli, 2003] Christian Blum and Andrea Roli, Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys* **35**, 3 (Sep. 2003), 268–308.
- [Deneubourg *et al.*, 1990] Jean-Louis Deneubourg, Serge Aron, Simon Goss and Jacques Pasteels, The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior* **3**, 2 (Mar. 1990), 159–168.
- [Dorigo, 2001] Marco Dorigo, Ant algorithms solve difficult optimization problems. In: *Advances in Artificial Life*, Springer Berlin, Heidelberg, 2001, 11–22.
- [Dorigo, 1992] Marco Dorigo, *Optimization, learning and natural algorithms*, PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.
- [Dorigo and Blum, 2005] Marco Dorigo and Christian Blum, Ant colony optimization theory: a survey. *Theoretical Computer Science* **344**, 2-3 (Nov. 2005), 243–278.
- [Dorigo and Gambardelle, 1997a] Marco Dorigo and Luca Maria Gambardelle, Ant colonies for the travelling salesman problem. *BioSystems* **43**, 2 (1997), 73–81.
- [Dorigo and Gambardelle, 1997b] Marco Dorigo and Luca Maria Gambardelle, Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* **1**, 1 (1997), 53–66.
- [Dorigo and Stützle, 2003] Marco Dorigo and Thomas Stützle, The ant colony optimization metaheuristic: algorithms, applications and advances. In: Fred Glover and Gary A. Kochenberger (eds.), *Handbook of Metaheuristics*. Springer, New York, 2006, 250–285.
- [Dorigo *et al.*, 1999] Marco Dorigo, Gianni Di Caro and Luca Maria Gambardelle, Ant algorithms for discrete optimization. *Artificial Life* **5**, 2 (1999), 137–172.
- [Dorigo *et al.*, 2006] Marco Dorigo, Mauro Birattari and Thomas Stützle, Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique, Université Libre de Bruxelles, IRIDIA, Report **TR/IRIDIA/2006-023**, September 2006.
- [Goss *et al.*, 1989] Simon Goss, Serge Aron, Jean-Louis Deneubourg and Jacques Pasteels, Self-organized shortcuts in the Argentine ant. *Naturwissenschaften* **76**, 12 (Dec. 1989), 579–581.
- [Grassé, 1946] Pierre-Paul Grassé, *Les Insectes Dans Leur Univers*. Ed. du Palais de la découverte, Paris, France, 1946.

- [Gutjahr, 2000] Walter Gutjahr, A graph-based Ant System and its convergence. *Future Generation Computer Systems* **16**, 9 (Jun. 2000), 873–888.
- [Hao *et al.*, 2006] Zhifeng Hao, Han Huang, Xili Zhang and Kun Tu, A time complexity analysis of ACO for linear functions. In: *Simulated Evolution And Learning: 6th International Conference, SEAL 2006, Lecture Notes In Computer Science* **4247** (2006), Springer, Berlin, 513–520.
- [Papadimitriou and Steiglitz, 1982] Christos H. Papadimitriou and Kenneth Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, N.J., 1982.
- [Stützle and Hoos, 2000] Thomas Stützle and Holger Hoos, MAX-MIN Ant System. *Future Generation Computer Systems* **16**, 8 (Jun. 2000), 889–914.

Social Human-Robot Interaction: Review of Existing Literature

Jakub Zlotowski

Abstract

Human-Robot Interaction is an emerging field in the Interactive Technology. Advancements in the area of Robotics allow first robots to appear on the public market, however there was limited research done up to date to investigate how they could optimally serve their users. In this paper I will review the literature on social aspects of Human-Robot Interaction and show how it can benefit from the research conducted on Human-Computer Interaction.

Keywords: human-robot interaction, social robot, social spaces, interactive robot, human interface.

CR-classification: I.2.9.

1. Introduction

Human-Computer Interaction (HCI), as a field, emphasized importance of understanding and improving interaction with digital systems. Usability, usefulness and technology's social impact [Kiesler and Hinds, 2004] as well as user centered design became a norm and goals in computing. In the recent years research on Human-Robot Interaction (HRI) became an emerging field within HCI.

Up until recently there was very little research done on HRI. As Thrun [2002] rightly points out, the development of this area was hampered by the state of robots at the time. They were mainly robust and programmable industrial machines with very little need of interaction with humans. However, in the last few years the area of robotics advanced rapidly and first commercial robots like Philip's iCat or Sony's Aibo were released on the market. This progress in robotics resulted also in increased interest and a need for HRI research.

Since HRI is rather a new field, the methodologies are not yet well established. HRI can greatly benefit from an existing research in related disciplines. Norman [2001] proposed Human-Computer Interaction, Automation, Science fiction, Computer-Supported Cooperative Work and Human Consciousness, Emotion and Personality as best sources for developing understanding of a human interaction with robots. Acquiring the knowledge from the other disciplines,

and effectively using it in robotics, seems to be especially promising, due to the fact, that the area is in its' early stages, and therefore, by the time robots will become everyday companion in human environment, we can optimize their interactivity. The work on HCI began long time after the beginning of computation and reached the state of widely accepted field of science only when PCs became a norm in households. Unfortunately, it took many years (and still is ongoing process) for the researchers to conduct studies on usability of computers, before it was possible to provide guidelines for an user interface design, to improve computer's usefulness and make it more accessible to ordinary people. Therefore, it is important to ensure that HRI will be well designed already before the advancement of technology will allow commercial production of robots for everyday use.

In this paper I will investigate existing literature on social aspects of HCI and Human Factors (described by Norman [2001] as human consciousness, emotion and personality) and their consequences in a relation to HRI. Weber [1947] defined social interaction, as a behavior oriented towards other's selves, which tries to affect and considers their actions and intentions. I will use this definition. In the next chapter I will discuss whether HCI or Human-Machine Interaction in general is basically social, based on only textual or audio interfaces. In chapter 3 I will focus on a research done on Embodied Conversational Agents and how this embodiment can affect and improve interaction. Finally, in chapter 4 I will present existing research in HRI and its comparison with HCI, as well as discuss what could be the benefits of having machine physically present in human environment.

2. Social Aspects of HCI

In the early days of a research on people's social responses to technology, the researchers thought about it as an abnormality. Most of the early experiments suggested that social behavior while interacting with computers is a result of person's dysfunction. The most common view supported, for example, by Barley [1988], Turkle [1984], Winograd and Flores [1987] and Zuboff [1988], was that only people who are young, lack of knowledge about technology or have either psychological or social dysfunction would respond socially towards machine. According to the same authors "normal", educated and mentally adjusted people would not express any social behaviors while interacting with machine.

On the other hand, probably almost all computer users either spoke themselves to the computer or heard other doing it in various situations, e.g., swearing at the computer after losing important data or muttering while playing computer games. Since majority of the people are well adjusted individuals, previously mentioned theories could hardly explain this type of behavior. Dennett [1987] proposed another explanation. Since the technology is simply proxy for the programmer [Searle, 1981], the people's social behavior towards it, is in fact directed at its' creator not a technology itself.

However Nass et al. [1997] challenged those views and in a series of experiments on Computers Are Social Actors (CASA) paradigm they showed that social responses to different types of media are normal, and common, and are not a result of dysfunction or unconsciously directing those responses towards human creator. Moreover, people are engaging in this kind of relationship with a machine without being aware of it and often in contradiction to their conscious declarations about not seeing machines as social. Nass et al. [1997] suggested that we can take any single theory about human-human interaction from Psychology and replace one human with a computer to test its validity in HCI. In addition, he stated that those findings are universal and impact not only HCI, but interaction with all the media and machines. Therefore, the conclusions from this research are also valid for HRI and provide framework for conducting experiments on social aspects in robotics.

2.1. Computers are Social Agents

Furthermore, in the same series of experiments [Nass et al., 1997] it was shown that to invoke user's social responses there is no need of using advanced technology. Changing a text style in text based HCI or voice of machine, were enough cues to cause different social reactions of people. Moreover, sociality of HCI spans over many areas and seems to be more universal than people would expect. People showed tendency to behave politely towards computers, the same way they do while interacting with human, as if they did not want to hurt their feelings. In addition, they observed that people preferred computers which had a similar personality to their own. Moreover, computers which flattered users, even insincerely, made them feel about themselves, interaction and computers more positively. Furthermore, just a vocal cues alone, like manipulating with gender of voice, were enough to elicit a stereotypic responses from the users [Lee et al., 2000]. They unconsciously and automatically behaved towards the com-

puter as if it had gender. Even when the voice was deliberately made to sound like produced by a machine, to remind users that they are interacting with computer, they still showed tendency to attribute personality to it [Nass and Lee, 2000]. Moreover, Fogg and Nass [1997] revealed that based on the rule of reciprocity, computers can motivate people to change behavior, and as a result, participants of their experiment were more helpful to the computer which helped them in previous task.

Those results bring important consequences for designing HRI. Since people behave politely while interacting with machines, it seems to be reasonable to assume that they would also expect machine to be polite while interacting with them. Therefore, it is important to ensure that robots will communicate with humans in a clear and polite way. In addition, Rewarding the user for successfully interacting with a robot, especially in the early stages, when people will be learning how to use and live with robots, could potentially lead to more positive perception of machines in human environment and encourage them to continue interacting in the future. Furthermore, this assumption can be supported by the research of Klein et al. [1999] who found that users are interacting longer with a system that has caused their frustration, if the system is affect-supportive. As a consequence, even if robot would not provide desired results in the early stages, while the technology will not be yet precise enough, it can still undo negative impact by helping a user control her emotional state.

On the other hand, those findings bring important questions for designers. Since even the smallest cues can result in a user perceiving machine as having personality or gender, and behaving differently based on that, it is extremely important to understand human psychology when choosing robot's personality or gender and base that choice on information about the users and are the tasks. It is crucial to ensure that an agent's or robot's personality will be consistent, as consistency of personality is one of the reasons why people tend to group others in their mental models. Since we assume that others behavior is a result of their personality, which is stable over time, it allows easier prediction how those others will behave in the future and increases feeling of controllability. However when a person behaves unpredictably and we cannot easily assign it to one of the existing schemas of personality, we tend to feel mental discomfort. Indeed, since the human machine interaction is social, we can assume that people will prefer machines, which behavior they would be able to predict and assign certain personality features. This assumption is supported by Lee and Nass [2003; Nass et

al., 2000] as they found that when a computer expressed a mismatching personality behaviors in different modalities, people liked the interaction less than when it was consistent. This conclusion is especially important for HRI since robots will have many modalities and will be able to express their personalities using various methods.

2.2. Human Computer Relationship

It is well documented [e.g., Ekman and Friesen, 1969; Maslow, 1943; Alderfer, 1972; Hinde, 1987] in Sociology and Social Psychology that people behave socially while interacting with each other and that need for establishing and maintaining a relationship is natural and universal for the humankind. Relationships provide many functions: belonging and a sense of “reliable alliance”, an emotional integration and stability, a chance for each partner to talk about themselves, a physical, psychological and emotional support as well as a reassurance of worth and value, and an opportunity to help others [Duck, 1991]. Since HCI is social, as discussed previously in this paper, we could also expect that it would be possible to build a social relationship with a machine which would fulfill all or at least some of those functions.

One of the foundations for establishing any kind of a relationship is empathy [Bickmore and Picard, 2005]. Despite of the fact that computers are not able to express it, since they do not have feelings, Klein et al. [1999] were able to show that even if a computer only appears to be emphatic and provides an accurate feedback, it was enough to achieve a user’s behavior which was similar to human empathy. Furthermore, Moon [1998] suggested that even in only text based HCI, a progressive reciprocal self disclosure increases liking and trust. Another widely investigated technique of increasing liking of HCI was an introduction of humor [Stafford and Canary, 1991; McGuire, 1994; Cole and Bradac, 1996; Morkes et al., 1998].

Another important need fulfilled by a relationship – stability – was investigated by Planalp [Planalp, 1993; Planalp and Benson, 1992]. He stated that people can differentiate a talk between strangers and acquaintances, based on whether they talk about mutual past or future and refer to mutual knowledge. This finding was developed further [Gilbertson et al., 1998] and a continuity behaviors, like greetings, farewells or talk about the time spent apart, were found to be crucial to create sense of persistence in a relationship. All those techniques could be

employed by robots, which are also especially promising to establish a relationship with, as they are mobile and will coexist with humans in their environment.

2.3. Critique of CASA paradigm

Not all the researchers agree with the view that an interaction with computers is the same or very similar to an interaction of people. While Nass et al. [1997] insists that people are responding to social cues, no matter whether they are coming from other people or media behaving like people, and proposed previously mentioned CASA paradigm for verifying which socio-psychological theories apply also to HCI. Shechtman and Horowitz [2003] proposed a bit different methodology for investigating social aspects of HCI. Instead of measuring non-conversational behaviors, they were interested to see whether there would be a difference in a conversation between people and human-machine conversation. Participants were informed whether they would be talking with another person using text-based computer mediated program or their conversational partner would be a computer. Afterwards the conversation was analyzed.

Participants put more effort in a conversation with human partner as they used more words and spent more time interacting with them. Moreover, those conversations seemed to differ in quality. Participants who knew that they are interacting with human partner wrote more relationship statements by trying to build a connection or influence the partner. In addition, they expressed more yielding and hostile behaviors towards their discussants. Finally, participants who were assertive exhibited that skill only when they were interacting with a human partner, and when the partner was assertive only those participants showed influencing behaviors, who were not talking with a computer.

Those results imply that social HCI is not the same as an interaction between people. It is important to remember that it is not enough to make a system just replicate human interaction and expect that people will respond to it as if it is another person. Probably people will communicate with robots in more distant manner and they will not be willing to interact with the machines. However, in my opinion, if the dialogue system will be advanced enough and allow robot to pursue normal conversation, the user will learn to trust it more and notice that robot actually can “understand” him, which would probably lead to a better quality of following conversations. It is possible that participants of Shechtman and Horowitz [2003] experiment did not put that much effort while talking with a computer, because they assumed that it will not understand them if they make

more complicated statements. Moreover, trying to influence the computer will not bring the desired results because it is only a preprogrammed device. Therefore, while it is good to remember that HRI will differ in some areas from interaction between people, it is still going to be social as the robot will be able to evoke certain social behaviors in the user.

3. Embodied conversational agents

Previously mentioned experiments were conducted using text or audio for HCI. However, advancements in a graphical technology allowed making animated, human like, embodied conversational agents (ECA) to improve that interaction. It seems to be an especially promising perspective, since major part of human communication is on a nonverbal level and vocal message only supplements it. Argyle [1988] identified five main functions of a bodily conversation: expressing emotions, communicating interpersonal attitudes, accompanying and supporting speech, self-presentation (personality) and rituals (e.g., greetings). Since those functions are part of a social interaction between people, and cannot be fulfilled by verbal communication, possibility of adding them to HCI would potentially enrich communication. Therefore, displaying a computer character with a body would allow it to express nonverbal information. As robots will be present in human environment and have physical appearance it would be interesting to see how embodiment affects interaction. In this chapter I will discuss how communication with ECA differs from a verbal or textual HCI and what are the advantages of making machine being able to communicate nonverbally.

3.1. When embodiment means progress

Lopez et al. [2006] investigated at which stages of communication, capacity of nonverbal behaviors of ECA could improve existing interfaces and what type of behavior should be employed. All these findings could be used in HRI similarly to the way in which Lopez and her colleagues presented them for conversational agents. They identified six stages of a human-machine communication which benefit the most from introduction of embodiment:

- Initiation – human interaction very often begins with some kind of welcome gesture. Despite of the fact that the gestures are not universal and differ between cultures (e.g., in Western cultures people shake hands when meeting others, while in some Eastern cultures bow has the same

- function) ECA and in future robots should be able to begin interaction with human in a more natural way.
- Termination – similarly to initiation the system should be able to greet the user properly for a farewell.
 - Turn management – which includes turn taking and giving. Humans have established numerous nonverbal cues for informing their conversational partner who should be talking. It is necessary that both, speaker and listener, are aware of those cues for the conversation to go smoothly. ECAs and robots should be able to express whether they need to take a turn for speaking or they want to pass it to the user and could do it by moving gaze or hand gesture, as some of the most effective cues for turn taking.
 - Recognition confidence schema – if the communication with the user will happen on a verbal level, the system might have different levels of confidence of understanding user's instructions. Therefore it is important that a person will be able to see whether an agent understood the user correctly or needs further explanation. Lopez et al. [2006] proposed that ECA could nod its head, smile and have eyes fully open when confidence is high; while in the low confidence situation an agent could move its head slightly to one side, stop smiling and mildly squint.
 - Recognition problems – if the system will understand user incorrectly and provide results different than expected, it should ensure that the user is able to repeat the message again and the system will be willing to correct itself. Lopez et al. [2006] implied that it could be done by expressing surprise and interest, by leaning towards user and rising eyebrows.
 - Help offers and requests – by being more open and accessible for the user, the user will be more willing to ask agent for a help as well as receive it.

Moreover, Rickel and Johnson [2000] proposed following roles for a nonverbal communication:

- Interactive demonstrations – ECA could demonstrate physical tasks, which could be more effective than describing them verbally. Potentially enough qualified robot could do the same by showing solution of real problems to the user with a major benefit.
- Navigational guidance – an agent can show where important objects are located.
- Gaze, gesture, and body orientation as attentional guides – an agent can guide user's attention using nonverbal communication in a natural way.

- Nonverbal tutorial feedback – without the need of using verbal communication, which could stop the conversation flow, an agent is able to provide feedback to the user (e.g., by nodding head).
- Conversational signals – are the same as discussed by Lopez et al. [2006] in turn management.
- Activities of virtual teammates – ECA can work as a teammate in tasks which require practicing groupwork with a benefit for future actual groupwork with real members.

3.2. Trustworthiness

In addition to those stages of interaction in which embodiment improves communication between a human and an agent or a robot, it can also affect other social aspects of that interaction. Since the robots will live with people at homes and be potentially capable of doing variety of tasks varying at the degree of importance for the user, it is inevitable that humans will trust them both in providing confidential information, as well as ordering to do different tasks without worrying about the results.

Milewsky and Lewis [1997] claim that trustworthiness of an agent will depend on its competence and the way in which an agent will present itself to the user. It seems that embodiment can have positive effect on the trustworthiness, by making the agents more human-like. People show tendency to express distrust while interacting with computers and trust during interaction with a human [Muir, 1987]. Furthermore, De Paulo [1992] claimed that facial expressions, body gestures and vocal intonations can be used to express trustworthiness and are interpreted in terms of it. However, van Mulken et al. [1999] found no support for that claim in the experiment.. While level of agent's competence was positively correlated with trustworthiness, whether agent was personified or not, did not affect how much people trusted it. Therefore, it is more important to ensure that robots will be competent in the tasks for which they are planned rather than focusing on making them more anthropomorphic.

3.3. Other benefits of embodiment

In spite of the fact that personification does not affect trustworthiness of an agent, there are other advantages of using it. In a learning environment students exhibited performance gains after interacting with an animated pedagogical agent [Lester et al., 1997]. Furthermore, in their experiment pupils had more

positive perception of their learning experience while being taught by an agent. In addition Lester et al. [1997] found that expressiveness of an agent plays a role in perceiving interaction. The more expressive was an agent, the more liked was the interaction.

In addition, Bickmore and Picard [2005] examined whether ECA can change user's behavior on course of an interaction. They introduced an agent which was able to express range of nonverbal behaviors such as a hand gestures, body posture shifts, gazing at and away from user, raising and lowering eyebrows, head nods, facial expressions, variable proximity, and walking on and off the screen. The researchers were interested to see how such an agent will be able to build and maintain relationship with the user and influence user's behavior. They found that people were willing to engage in a relationship with ECA, and perceived that relationship more positively, than while interacting with not a relational agent. Participants created emotional-bond with an agent. On the other hand, the influence of an agent on participant's behavior ended when the interaction between them was terminated at the end of experiment. However, since the robots are going to be an everyday human companion there is no need for ending interaction and therefore they can support a healthier behavior of their user's continuously.

Another reason for humanizing computers could be to make them easier and more comfortable to use [Laurel, 1990; Shneiderman, 1987]. Sproull et al. [1997] investigated how talking-face display would differ from text display. Subjects in their experiment attributed some of the personality features differently for those displays. On the negative side, they felt less relaxed, and confident, and expressed higher arousal while interacting with an agent. In addition, they tried to present themselves in a more positive light in that condition, as if they were interacting with a human partner. Moreover, this research shows that there are gender differences in HCI. Men enjoyed more interaction with a humanized agent and they spent more time on it than women, who rather preferred a text based interface. Those results are consistent with previous findings about men liking new technology more than women [Chen, 1985]. As the robots, in human environment, are going to be a major difference in comparison with currently existing technology, we could expect that in the early stages men will be keener on interacting with them. Nevertheless, there should be no assumptions yet as more research is needed on a gender differences in regard to technology as it is

possible that if the face would be more natural or opposite gender women participants would respond differently to it.

There is evidence that also factors like agent's ethnicity has effects on user's attitudes and behaviors. As it can be drawn from numerous Social Psychology studies, people usually prefer people from the same ethnic group and treat them as in-group members, what results in perceiving them more favorably on many scales. Based on that Nass et al. [2000] showed similar pattern with ECA. When an agent was of the same ethnicity, subjects in their experiment perceived it as more similar to them, more socially attractive and trustworthy. Moreover, they conformed more to it while making group decision and perceived its arguments as better.

Non human appearance of an agent was studied by Kiesler and Sproull [1997]. They wanted to see whether there are any differences between anthropomorphic and animal-looking agents. They found that while dog agent was more liked (or even loved according to authors), the participants interacted less with it. This experiment showed how important it is which metaphor is used for representing ECA or a robot. Based on the physical appearance people create different expectations about capabilities and interaction with a system. If the robot looks too infantile (like a pet) it will be treated that way. People might love it, but they are not going to expect from it being able to do more complicated tasks.

This opinion is also supported by Nass et al. [1994] who claim that people anticipate the behavior of embodied agents and those expectations drive their own behavior. Furthermore, Halasz and Moran [1982] suggested that metaphoric designs in HCI can cause numerous problems, as people might overextend their expectations based on the metaphor. Churchill et al. [2000] proposed that the agent should explicitly and implicitly (through appearance and behavior) provide cues about its limitations and offer alternatives for user goals. Therefore it is designer's responsibility to ensure that the robot will present itself as capable of doing expected tasks.

4. Human Robot Interaction

In the previous two chapters I discussed existing knowledge about social aspects of HCI and how it is relevant to the field of robotics. Based on the Nass and colleagues [1997] research under CASA paradigm I assumed that also HRI will be social and suggested utilizing findings from HCI on that issue, while planning,

designing and investigating HRI. In the previous chapter I focused on the benefits and costs of personifying ECA and potentially robots. While the results of those studies can help researchers in investigating HRI, the advancements of robotics finally also allowed examining directly this field. In this chapter I will discuss the current state of art in HRI and discuss various views on making robots socially capable.

4.1. Differences between HCI and HRI

Despite of the fact that there are many similarities between HCI and HRI, there are also some differences. Therefore, I will present them in this section as they can provide boundaries for utilizing results of research from one field to another.

According to Kiesler and Hinds [2004] people perceive robots differently than most other computer technologies. Partially as a consequence of Science Fiction, as well as a result of perception of autonomous movement [Scholl and Tremoulet, 2000], people's mental models of robots are more anthropomorphic in a comparison with other systems [Friedman et al., 2003]. Therefore we can assume that interaction with robots will be even more social than with other media.

Another significant difference is that the majority of the robots will be fully mobile and the interaction will take place in a rapidly changing human environment [Kiesler and Hinds, 2004]. The robots will need to negotiate their interaction in a dynamic and physically challenging environment. Use of nonverbal communication in those situations seems to be especially promising and the embodiment of robot might be a major help to solve that problem. Severinson-Eklundh et al. [2003] argue that robot must be capable of social interaction not only with its user, but also a group of people who are present in its environment, which can include also strangers.

Finally, Kiesler and Hinds [2004] imply that another important difference is that robots make decisions, learn about themselves and surrounding world, and impact the information which they process and actions they emit. Since this difference does not affect sociality of HRI, I will not discuss it in details in this paper.

4.2. Should the robots be social agents?

The question whether robots should be socially capable machines should be asked at this point. Vast majority of the research gives positive answer. Furthermore, Norman [2001] presents more arguments in favor of that opinion. He

states that emotion is a communicative device, both within and among people. "The variety of subtle behavioral changes can convey a rich communicative message about the current state of affairs and the intentions for future responses" [Norman, 2001]. By providing them to robots, the human-robot communication can be improved and become more natural. In addition, he claims that personality (which can be implied through robot's movement [Nakata et al., 1998]) provides a good contextual model for interpreting and understanding robot's behavior and helps in behaving in interaction and in giving commands. Those claims are also supported by Butler and Agah [2001] who assert that emotional robots will be more predictable by humans.

In support with my assumption that HRI is social, Friedman et al. [2003] conducted content analysis of AIBO's (a robotic pet) discussion forum. They were interested to see whether users are attributing any social aspects to AIBO while referring to it. While they attributed mental states (like capability of a robotic pet for intentions, desires and feelings) and social rapport (capability for engaging in social relationships) they did not believe that AIBO has moral standing. This research proves that people see robots as socially capable (at least to some degree).

On the other hand Schmidt [2005] proposes that the robots should not be anthropomorphic. He opposes the idea of robots becoming socially similar to people as it will be harder to distinguish between them and consequently it will not be beneficial for human. Schmidt proposes distinction between challenge-driven and purpose-guided technology. According to him making robots socially capable is only challenge-driven approach, as it is an attempt of replicating human beings in a digital form. Since it does not have any real purpose and does not provide any benefits to humans, he suggests stopping research on humanizing machines and making distinction between humans and technology more visible. He supports his claims by mentioning that in some situations social cues provided by robot might be counter-productive and slow down the user.

Many science fiction movies were produced around the idea that humans and robots will be in future alike and fighting for the dominance in the world. While I agree with Schmidt that developing technology just for sake of it is not purposeful, I still do not see a need for not trying to make machines more humanlike. There is a major distinction between making robots sensitive to human social cues or capable of behaving socially, and trying to substitute humans by them. Many innovations if used improperly can cause negative consequences, however that is not a reason for stopping their development. There is enormous evidence

discussed in this paper that socially capable technologies will benefit people and make the interaction more natural. Even if there are situations in which it is not a case, it is possible to investigate them and limit a use of new technology, while benefiting from it in all the others.

Since robots are going to become part of human environment it would be interesting to ask them how they would like the robots to look. As Khan [1998] suggests human mental model of robots might be highly influenced by Science Fiction movies and literature. He conducted series of interviews to investigate the issue and the results confirmed that hypothesis. When his participants were asked to draw robot as they imagine it, the drawings were based on different movies. However, during the interview he found also other interesting patterns. People did not have any specific expectations for robot's appearance; however they showed tendency to prefer a domestic devices' look. Moreover, the desired appearance was dependent on the machine's tasks (with basic home tasks chosen as the most beneficial). Participants did not want the robot to have any gender or age, but to be rather small (child size).

5. Conclusions

The advancement of technology allows constructing more sophisticated machines. We can currently observe appearance of the first commercial robots on the market and predict that in the near future they will become present in human living and working environments. Those first HRIs might be critical in defining whether this technology will be accepted by the society in the early stages, and affect pace of further development of robotics. Therefore, it is crucial to ensure that this communication will match human skills, needs and expectations. Despite of the relatively limited existing research on HRI, this field can benefit greatly from introducing solutions from HCI. Creating robots which could interact socially with humans is not a Science Fiction anymore, it can become a reality.

References

- [Alderfer, 1972] Clayton P. Alderfer, *Existence, Relatedness, and Growth*. New York, NY: Free Press, 1972
- [Argyle, 1988] Michael Argyle, *Bodily Communication*. Madison, CT: International Universities Press, 1998.
- [Barley, 1988] Stephen R. Barley, The social construction of machine: ritual, superstition, magical thinking, and other pragmatic responses to running CT

- scanner. In: Margaret M. Lock & Deborah R. Gordon (Eds.), *Biomedicine Examined*. Hingham, MA: Reidel, 1988, 497-540.
- [Bickmore and Picard, 2005] Timothy W. Bickmore, & Rosalind W. Picard, Establishing and maintaining long-term human-computer relationships. *ACM Transactions on Computer-Human Interaction (TOCHI)*. **12** (2005), 293-327.
- [Butler and Agah, 2001] John T. Butler, & Arvin Agah, Psychological effects of behavior patterns of a mobile personal robot. *Autonomous Robots*. **10** (2001), 185-202.
- [Chen, 1985] Milton Chen, Gender differences in adolescents' uses of and attitudes toward computers. In: Margaret J. McLaughlin (Ed.), *Communication Yearbook*. Beverly Hills, CA: Sage, 1985, 200-216.
- [Churchill et al., 2000] Elisabeth F. Churchill, Linda Cook, Peter Hodgson, Scott Prevost, & Joseph W. Sullivan, "May I help you?" Designing embodied conversational agent allies. In: Justine Cassell, Joseph Sullivan, Scott Prevost & Elisabeth Churchill (Eds.), *Embodied Conversational Agents*. Cambridge, MA: MIT Press, 2000, 64-94.
- [Cole and Bradac, 1996] Tim Cole, & James J. Bradac, A lay theory of relational satisfaction with best friends. *Journal of Social and Personal Relationships*. **13** (1996), 57-83.
- [Dennett, 1987] Daniel C. Dennett, *The Intentional Stance*. Cambridge, MA: MIT Press, 1987.
- [DePaulo , 1992] Bella M. DePaulo, Nonverbal behavior and self-presentation. *Psychological Bulletin*. **111** (1992), 203-243.
- [Duck, 1991] Steve Duck, *Understanding Relationships*. New York, NY: Guilford Press, 1991.
- [Ekman and Friesen, 1969] Paul Ekman & Wallace V. Friesen, The repertoire of nonverbal behavior: Categories, origins, usage, and coding. *Semiotica*. **1** (1969), 49- 98.
- [Fogg and Nass, 1997] BJ Fogg, & Clifford Nass, How users reciprocate to computers: An experiment that demonstrates behavior change. In: *Conference on Human Factors in Computing Systems*. New York, NY: ACM, 1997, 331-332.
- [Friedman et al., 2003] Batya Friedman, Peter H. Kahn, & Jennifer Hagman, Hardware companions? – What online AIBO discussion forums reveal about the human-robotic relationship. In: *Conference on Human Factors in Computing Systems*. New York, NY: ACM, 2003, 273-280.

- [Gilbertson et al., 1998] Jill Gilbertson, Kathryn Dindia, & Mike Allen, Relational continuity constructional units and the maintenance of relationships. *Journal of Social and Personal Relationships*. **15** (1998), 774-790.
- [Halasz and Moran, 1982] Frank Halasz, & Thomas P. Moran, Analogy considered harmful. In: *Conference on Human Factors in Computing Systems*. New York, NY: ACM, 1982, 383-386.
- [Hinde, 1987] Robert A. Hinde, *Individuals, Relationships & Culture: Links Between Ethology and the Social Sciences*. Bath: Cambridge University Press, 1987.
- [Khan, 1998] Zayera Khan, *Attitudes towards intelligent service robots*. Royal Institute of Technology, IPLab, NADA, Report TRITA-NA-E98421 - IPLab-154, 1998.
- [Kiesler and Hinds, 2004] Sara Kiesler, & Pamela Hinds, Introduction to this special issue on human-robot interaction. *Human-Computer Interaction*, **19** (2004), 1-8.
- [Kiesler and Sproull, 1997] Sara Kiesler, & Lee Sproull, 'Social' human-computer interaction. In: Batya Friedman (Ed.), *Human Values and the Design of Computer Technology*. Stanford, CA: Cambridge University Press, 1997, 191-199.
- [Klein et al., 1999] Jonathan Klein, Youngme Moon, & Rosalind W. Picard, This computer responds to user frustration. In: *Conference on Human Factors in Computing Systems*. New York, NY: ACM, 1999, 242-243.
- [Laurel, 1990] Brenda Laurel, *The Art of Human-Computer Interface Design*. Reading, MA: Addison-Wesley, 1990.
- [Lee et al., 2000] Eun J. Lee, Clifford Nass, & Scott Brave, Can computer generated speech have gender? An experimental test of gender stereotype. In: *CHI '00 Extended Abstracts on Human Factors in Computing Systems*. New York, NY: ACM, 2000, 289-290.
- [Lee and Nass, 2003] Kwan M. Lee, & Clifford Nass, Designing social presence of social actors in human computer interaction. In: *Conference on Human Factors in Computing Systems*. New York, NY: ACM, 2003, 289-296.
- [Lester et al., 1997] James C. Lester, Sharolyn A. Converse, Susan E. Kahler, S. Todd Barlow, Brian A. Stone, & Ravinder S. Bhogal, The persona effect: Affective impact of animated pedagogical agents. In: *Conference on Human Factors in Computing Systems*. New York, NY: ACM, 1997, 359-366.
- [Lopez et al., 2006] Beatriz Lopez, Alvaro H. Trapote, David D. Pardo de Vera, Doroteo T. Toledano, Luis A.H. Gómez, & Eduardo L. Gonzalo, A good

- gesture: Exploring nonverbal communication for robust SLDSs. In: *IV Jornadas en Tecnologia del Habla*, 2006, 39-44.
- [Maslow, 1943] Abraham H. Maslow, A theory of human motivation. *Psychological Review*. **50** (1943), 370-396.
- [McGuire , 1994] Anne M. McGuire, Helping behaviors in the natural environment: dimensions and correlates of helping. *Personality and Social Psychology Bulletin*. **20** (1994), 45-56.
- [Milewsky and Lewis, 1997] Allen E. Milewsky, & Steven H. Lewis, Delegating to software agents. *International Journal of Human-Computer Studies*. **46** (1997), 485-500.
- [Moon, 1998] Youngme Moon, *Intimate self-disclosure exchanges: Using computers to build reciprocal relationships with consumers*. Cambridge, MA: Harvard Business School, 1998.
- [Morkes et al., 1998] John Morkes, Hadyn H. Kernal, & Clifford Nass, Humor in task-oriented computer-mediated communication and human-computer interaction. In: *Conference on Human Factors in Computing Systems*. New York, NY: ACM, 1998, 215-216.
- [Muir, 1987] Bonnie M. Muir, Trust between humans and machines, and the design of decision aids. *International Journal of Human-Computer Studies*. **27** (1987), 527-539.
- [Nakata et al., 1998] Toru Nakata, Tomomasa Sato, & Taketoshi Mori, Expression of emotion and intention by robot body movement. In: *Proc. of the Conference of International Autonomous Systems*. **5** (1998), 352-359.
- [Nass et al., 2000] Clifford Nass, Katherine Isbister, & Eun J. Lee, Truth is beauty: Researching embodied conversational agents. In: Justine Cassell, Joseph Sullivan, Scott Prevost & Elisabeth Churchill (Eds.), *Embodied Conversational Agents*. Cambridge, MA: MIT Press, 2000, 374-402.
- [Nass and Lee, 2000] Clifford Nass & Kwan M. Lee, Does computer-generated speech manifest personality? An experimental test of similarity-attraction. In: *Conference on Human Factors in Computing Systems*. New York, NY: ACM, 2000, 329-336.
- [Nass et al., 1997] Clifford Nass, Youngme Moon, John Morkes, Eun Y. Kim, & B.J. Fogg, Computers are social actors: A review of current research. In: Batya Friedman (Ed.), *Human Values and the Design of Computer Technology*. Stanford, CA: Cambridge University Press, 1997, 137-162.

- [Nass et al., 1994] Clifford Nass, Jonathan Streuer, & Ellen R. Tauber, Computers are social actors. In: *Conference on Human Factors in Computing Systems*. New York, NY: ACM, 1994, 72-78.
- [Norman, 2001] Don Norman, How might humans interact with robots? Retrieved, November 23, 2009. Available as http://jnd.org/dn.mss/how_might_humans_interact_with_robots.html.
- [Planalp, 1993] Sally Planalp, Friends' and acquaintances' conversations II: coded differences. *Journal of Social and Personal Relationships*. **10** (1993), 339-354.
- [Planalp and Benson, 1992] Sally Planalp, & Anne Benson, Friends' and acquaintances' conversations I: Perceived differences. *Journal of Social and Personal Relationships*. **9** (1992), 483-506.
- [Rickel and Johnson, 2000] Jeff Rickel, & W. Lewis Johnson, Task-oriented collaboration with embodied agents in virtual worlds. In: Justine Cassell, Joseph Sullivan, Scott Prevost & Elisabeth Churchill (Eds.), *Embodied Conversational Agents*. Cambridge, MA: MIT Press, 2000, 95-122.
- [Schmidt, 2005] C.T.A. Schmidt, Of robots and believing. *Minds and Machines*. **15** (2005), 195-205.
- [Scholl and Tremoulet, 2000] Brian J. Scholl, & Patrice D. Tremoulet, Perceptual causality and animacy. *Trends in Cognitive Science*. **4** (2000), 299-309.
- [Searle, 1981] John R. Searle, Minds, brains and programs. In: Douglas R. Hofstadter & Daniel C. Dennett (Eds.), *The Minds I*. Toronto: Bantam, 1981.
- [Severinson-Eklundh et al., 2003] Kerstin Severinson-Eklundh, Anders Green, & Helge Huttenrauch, Social and collaborative aspects of interaction with a service robot. *Robotics and Autonomous Systems*. **42** (2003), 223-234.
- [Shechtman and Horowitz, 2003] Nicole Shechtman, & Leonard M. Horowitz, Media inequality in conversation: how people behave differently when interacting with computers and people. In: *Conference on Human Factors in Computing Systems*. New York, NY: ACM, 2003, 281-288.
- [Shneiderman, 1987] Ben Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Boston, MA: Addison-Wesley, 1987.
- [Sproull et al., 1997] Lee Sproull, Mani Subramani, Sara Kiesler, Janet Walker, & Keith Waters, When the interface is a face. In: Batya Friedman (Ed.), *Human Values and the Design of Computer Technology*. Stanford, CA: Cambridge University Press, 1997, 163-190.

- [Stafford and Canary, 1991] Laura Stafford & Daniel J. Canary, Maintenance strategies and romantic relationship type, gender and relational characteristics. *Journal of Social and Personal Relationships*. 8 (1991), 217-242.
- [Thrun, 2002] Sebastian Thrun. Probabilistic robotics, *Communications of the ACM*. 45 (2002), 52-57.
- [Turkle, 1984] Sherry Turkle, *The Second Self: Computers and the Human Spirit*. New York, NY: Simon & Schuster, 1984.
- [van Mulken et al., 1999] Susanne van Mulken, Elisabeth Andre, & Jochen Muller, An Empirical Study on the Trustworthiness of Life-Like Interface Agents. In: *Proc. of the HCI International '99 on Human-Computer Interaction: Communication, Cooperation, and Application Design*. Hillsdale, NJ: L. Erlbaum Associates Inc., 1999, 152-156.
- [Weber, 1947] Max Weber, *The Theory of Social and Economic Organization*. New York, NY: Oxford University Press, 1947.
- [Winograd and Flores, 1987] Terry Winograd & Fernando Flores, *Understanding Computers and Cognition: A New Foundation for Design*. Reading, MA: Addison-Wesley, 1987.
- [Zuboff, 1988] Shoshana Zuboff, *In the Age of the Smart Machine: The Future of Work and Power*. New York, NY: Basic Books, 1988.

Supervisor: Poika Isokoski