

Lukijalle

Tähän julkaisuun on koottu syyslukukaudella 2003 pitämälläni Tutkimuskurssilla tehdyt suomenkieliset tutkielmat. Englanninkieliset tutkielmat on julkaistu erillisessä raportissa B-2004-4.

Toimittaja

Sisällysluettelo

Varjoalgoritmeista	1
<i>Tero Alatalo</i>	
Sykeinformaation käyttömahdollisuudet käyttäjän emootioiden tunnistamisessa	16
<i>Jenni Anttonen</i>	
Tuntopalautteen hyödyntäminen näkövammaisten esikoululaisten opetuksessa.....	38
<i>Taina Arjanmaa</i>	
Puhekäyttöliittymä osana virtuaaliodellisuusympäristöjä	54
<i>Mark Claydon</i>	
Muutosvastarinta ilmiönä	70
<i>Anne Jokinen</i>	
Kämmentietokone ryhmän tukemisessa.....	81
<i>Tuomo Kivinen</i>	
Lapsille suunnattujen verkkosivujen käytettävyys	99
<i>Matleena Koivisto</i>	
WWW-hakukoneiden käyttö opiskelun tiedonlähteenä	120
<i>Anne Kunnari</i>	
Suorakäyttöisten kaavionmuokkaustyökalujen käytettävyys	163
<i>Päivi Kärkkäinen</i>	
Dynaamisesti ladattavan koodin tietoturva	180
<i>Timo Linden</i>	
Esimaksettu GSM-puhelu ja sen vaikutus käyttäjiin.....	195
<i>Jani Lundan</i>	
Kansainvälisen WWW-sivuston suunnitteluhaasteet ja käytettävyydestaus.....	208
<i>Laura Mustonen</i>	
Kulttuurierojen huomioiminen käyttöliittymäsuunnittelussa.....	219
<i>Minna Nurminen</i>	

Suorakäyttöisyys kolmiulotteisessa veisto-ohjelmassa.....	237
<i>Jyrki Parviainen</i>	
Loppukäyttäjät osana tietojärjestelmäprojektia	251
<i>Juha Pieviläinen</i>	
Matkapuhelinten käytettävyysongelmia pelaajan näkökulmasta.....	263
<i>Valtteri Pihlajamäki</i>	
Tietokone musiikintekijän välineenä	284
<i>Harri Pora</i>	
Sähköpostipalvelun ongelmista	300
<i>Janne Repo</i>	
Terveydenhuollon tietojärjestelmien tietoturva.....	310
<i>Jussi Saarinen</i>	
Kevyt agenttialusta heterogeenisessä ympäristössä.....	325
<i>Rami Saarinen</i>	
3D-veisto-ohjelman työkalujen käytettävyys	350
<i>Nina Sainio</i>	
Relaatiotietokantojen takaisinmallintaminen.....	372
<i>Ari Seppi</i>	
Katsaus mallipohjaiseen käyttöliittymäkehitykseen	391
<i>Perttu Sliden</i>	
Kuntien verkkosivujen sisällön kartoitus.....	403
<i>Minna Sundström</i>	
Rinnakkaisuuden toteutus Symbian OS käyttöjärjestelmässä	417
<i>Liisa Supponen</i>	
Laajojen hajautettujen sovellusten ja tietojärjestelmien valvonta.....	431
<i>Hannu Tanhuamäki</i>	
Apuvälinepalvelun tietojärjestelmät terveydenhuollossa.....	443
<i>Outi Toivanen</i>	
Sukututkimusohjelmat.....	461
<i>Virpi Tuohisto</i>	
C++:n ja Prologin käyttö toisiaan täydentäen	482
<i>Jouni Vaaramo</i>	
Ratkaisuja roskienkeruun ongelmaan	490
<i>Zoltán Varga</i>	
Teknologian vaikutus opetukseen	511
<i>Matti Voutilainen</i>	
Tietojärjestelmät perusterveydenhuollossa	528
<i>Suvi Vuorela</i>	

Varjoalgoritmeista

Tero Alatalo

Tiivistelmä.

Tutkielmassa esitellään muutamia olennaisimpia varjoalgoritmeja sekä pohditaan hiukan niiden tehokasta toteutusta nykylaitteiston kannalta.

Avainsanat ja -sanonnat: Varjoalgoritmit, tietokonegrafiikka

CR-luokat: I.3.7, I.3.3

1. Johdanto

Varjot ovat tärkeä visuaalinen vihje kappaleiden välisten etäisyyksien ja sijaintien hahmottamisessa. Näkymän tilan hahmottamisen lisäksi varjot parantavat tietokonegrafiikalla luodun näkymän realistisuutta huomattavasti – onhan oikeassa maailmassakin varjot. Myös tekniikan ulkopuolella monet taiteenlajit pitävät varjoja merkityksekkäänä ja vahvana osana kuvaa. Koska realismiin pyrkivällä tietokonegrafiikalla on perinteisesti haluttu saavuttaa mahdollisimman *oikealta ja uskottavalta* näyttävä kuva, ovat varjoalgoritmit ehdottoman tärkeä osa kuvan muodostamista.

Varjojen luonti on yksi tietokonegrafiikan tutkituimmista ja ongelmallisimmista pulmista lähinnä sen haastavuuden ja laskennallisen raskauden vuoksi. Nykytekniikka ja uudet laitteistot mahdollistavat tehokkaampia toteutustapoja vanhoille tekniikoille ja avaavat ovia kokonaan uusille varjonpiirtotekniikoille. Esimerkkinä voidaan pitää vasta viimeaikoina mielekkäästi laitteistokiihdytettynä ja reaaliaikaisena massamarkkinoilla saatavilla olevalla laitteistolla toteutettavaa varjosyvyyspuskurialgoritmia (engl. *shadow depth buffering*, *shadow maps*), joka on kuvattu ensimmäisen kerran SIGGRAPHin julkaisuissa jo 70-luvulla [Williams, 1978].

Varjoalgoritmeja tarvitaan käytännössä kaikissa realistista 3D-grafiikkaa esittävässä sovelluksissa. Reaaliaikaisena algoritmeja käytetään yleisesti virtuaalitodellisuussovelluksissa ja digitaalisessa viihteessä kuten peleissä. Viihdeteollisuus lieneekin ollut viimevuosien ajan teknologiaa eteenpäin vievä tekijä. Erilaiset 3D-mallinnusohjelmat ja CAD-työkalut käyttävät myös usein hyväkseen varjoja, mutta näissä sovelluksissa laskentaa ei yleensä tarvitse tehdä useita kertoja sekunnissa reaaliajassa. Tuotetun kuvan laatu on sen sijaan tärkeää, sillä esimerkiksi elokuvaan tehtävien erikoistehosteiden tulee luonnollisesti yleensä olla mahdollisimman aidon näköisiä.

Tässä tutkielmassa keskitytään nimenomaan reaaliaikaisen grafiikan ja dynaamisten varjojen esittämiseen pääasiassa perinteisiä monikulmiopohjaisia

piirtotekniikoita käyttäen. Etenkin keskitytään nykylaitteistolla mielekkäästi toteutettaviin tekniikoihin. Osa tutkielmassa kuvattavista tekniikoista soveltuisi melko suoraan esimerkiksi vokselipohjaisen näkymän esittämiseen, mutta näihin menetelmiin ei tässä tutkimuksessa tulla sen enempää puuttumaan. Toisaalta suurin osa tekniikoista on mahdollista siirtää melko vaivattomasti - tosin vaihtelevin tehokustannuksin - monikulmiopohjaisiksi ja täten ainakin epäsuorasti sovellettavissa tämän tutkimuksen tulosten pohjalta.

2. Ongelman luonne

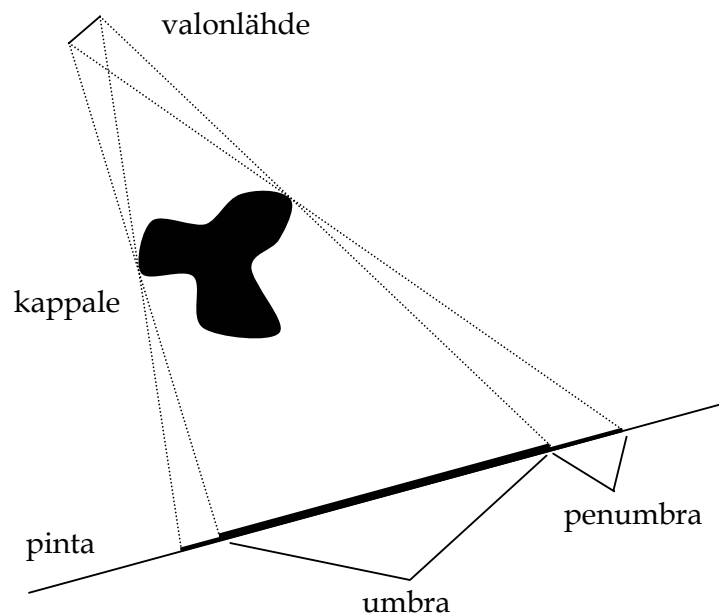
Reaaliaikaisuusvaatimus asettaa muutamia erityisehtoja ongelman ratkaisulle. Varjoalgoritmin täytyy sinänsä olla mahdollisimman tehokas ruudunpäivitysnopeuden pitämiseksi riittävän korkeana, mutta toisaalta sen tulee myös tuottaa mahdollisimman laadukas lopputulos. Juuri näiden syiden vuoksi perinteisissä menetelmissä joudutaankin usein tekemään kompromisseja laadussa tyydyttävän nopeuden vuoksi. Perinteisesti ala onkin täynnä pieniä "huijauksia", joilla pyritään ratkaisemaan jokin ongelma tai osaongelma jollain *riittävän hyvällä* approksimaatiolla täydellisestä ratkaisusta. Tämä tuleekin vahvasti ilmi osassa esitetyistä ratkaisuksista.

Suppeasti määriteltynä ongelmana on tuottaa mielivaltaiselle monikulmioista koostuvalle 3D-näkymälle ja sen hahmoille uskottavat varjot tarpeeksi nopeaa reaaliajassa esitettäväksi. Vaikka grafiikan esittämistapa on rajoitettu ainoastaan monikulmioihin, ei silti ole olemassa mitään yhtä ainoaa "oikeaa" tapaa toteuttaa näkymän piirtoa ja ns. 3D-grafiikkamoottoria. Syynä tähän on pääasiassa se, että monella kuvatuista tekniikoista on omat hyvät ja huonot puolensa - osa soveltuu paremmin toisiin käyttötarkoituksiin kuin toiset ja päinvastoin. Virtuaalitodellisuussovelluksen kaikkien eri osa-alueiden saaminen toimimaan yhteen on sekin itsessään hyvin haastava - ja kokonaan toinen - ongelma. Muun muassa David Eberly on jäsennellyt 3D-moottorin toimintaa ja käytännöntoteutusta kirjassaan *3D Game Engine Design* [Eberly, 2001]. Tietokone pelit lienevätkin tällä hetkellä monimutkaisimpia reaaliaikaista 3D-grafiikkaa käyttäviä sovelluksia.

Vaatimuksia varjoalgoritmille asettavat muun muassa näkymässä olevien monikulmioiden määrä sekä tarpeet animoida ja liikutella kappaleita. Jos esimerkiksi voidaan olettaa, että valonlähteiden sijainti pysyy suhteellisesti liikkumattomana kappaleiden sijaintiin ja orientaatioon nähden, voidaan valtaosa laskennasta suorittaa yleensä esilaskentana. Tämä lähestymistapa voi keventää reaaliajassa tarvittavan laskennan määrää, mutta vastaavasti käyttää enemmän muistia ja rajoittaa liikkeen dynaamisuutta.

2.1. Varjoista

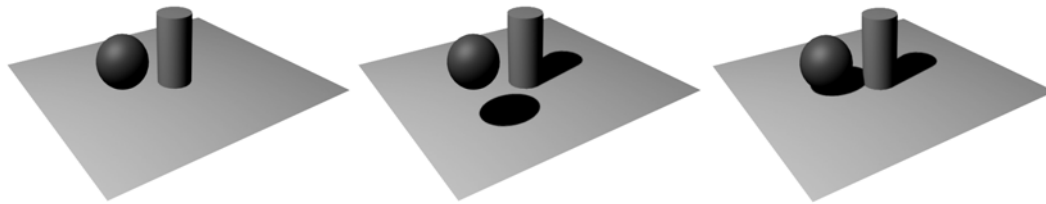
Varjoja on siellä, missä valoa ei ole. Varjon intensiteetti riippuu osaksi sitä ympäröivien alueiden kontrastieroista sekä siitä kuinka paljon valoa varjoalueelle pääsee heijastumaan. Piste on varjossa valonlähteen suhteen mikäli pisteen ja valonlähteen välinen jana leikkaa matkalla jotain kappaletta. Yksinkertaisen varjon voidaan ajatella koostuvan valonlähteestä, varjon heittävästä kappaleesta (*occluder*), varjon vastaanottavasta pinnasta (*receiver*), sekä itse varjosta, johon kuuluu täysin varjossa oleva *umbra* ja osaksi varjossa oleva *penumbra* (kuva 1). Umbran ja penumbran koko riippuu valonlähteen koosta sekä kappaleen, valonlähteen ja vastaanottavan pinnan välisistä etäisyyksistä. Jo Leonardo Da Vinci määritteli tämän asetelman aikoinaan.



Kuva 1 Valonlähde, kappale ja varjon osat

Perinteisessä tietokonegrafiikassa varjoja ei määritellä näin tarkasti. Valonlähteen ajatellaan yleensä olevan yksi äärettömän pieni piste tai yksittäinen suunta. Jotta voitaisiin määrittää kuvan 1 mukaiset pehmeäreunaiset varjot (*soft shadows*), täytyisi valoa säteilevän lähteen koko ottaa jotenkin laskennassa huomioon (*area lighting*). Tässä tutkielmassa keskitytään lähinnä edellä mainitun äärettömän pienen tai suunnatun valonlähteen heittämiin varjoihin, mutta osa kuvatuista algoritmeista olisi – joskin potentiaalisen tehottomasti – muokattavissa ottamaan huomioon valoa heijastavan pinnan koko [Assarson *et al.*, 2003].

Varjojen olemassaolo tietokoneella generoidussa kuvassa on ehdottoman tärkeää kuvan havaitsemisen kannalta. Kuvasta 2 voidaan selvästi huomata miten varjojen käyttö auttaa hahmottamaan kappaleiden välisiä suhteita ja etäisyyksiä. Varjottoman näkymän tulkinta voisi olla esimerkiksi kumpi tahansa varjollisista näkymistä. Vaikka ilmiö korostuukin yksittäisessä kuvassa, ei ongelma ole ihan niin vakava liikkuvassa kuvassa: perspektiivimuutokset sekä tulkinta kappaleiden liikkeestä selkeyttävät tilan hahmottamista. Lisäksi animaatioissa on yleensä havaittavissa järkevä kausaalisuus tapahtumien välillä: kappaleet ovat liikkeessä paikasta toiseen ja kuvasarjan seuraavat kuvat sisältävät verrattain pieniä muutoksia edelliseen kuvaan.



Kuva 2 Näkymä ilman varjoja ja kaksi tulkintaa varjojen kanssa

2.2. 3D-olioista ja -liukuhihnasta

Määritellään 3D-malli olioksi, joka koostuu kolmioista. Kolmio koostuu kolmesta tietyssä järjestyksessä määritellystä kärkipisteestä (engl. *vertex*). Nämä kärkipisteet määrittävät edelleen yksikäsitteisesti tason, jolla pisteet sijaitsevat. Yksikäsitteisyys saadaan järjestyksestä, jossa pisteet on määritelty kolmioon kuuluviksi.

3D-malli voi olla pinnoitettu tekstuurilla (*texture mapping*). Jokaiselle kulmapisteelle määritellään *tekstuurikoordinaatti*, jonka perusteella päätellään, mikä kohta tekstuurista kuvautuu kolmion kulmapisteeseen. Kolmion sisälle jäävät pisteet voidaan interpoloida tekstuurista kärkipisteiden tekstuurikoordinaattien avulla.

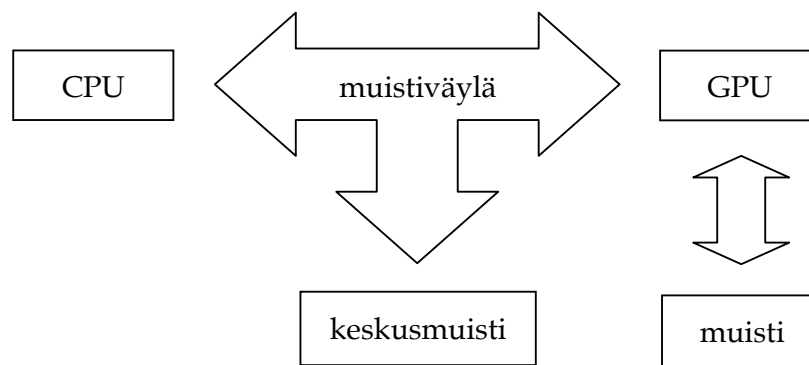
Näkymä voi koostua useista 3D-malleista. Samaa mallia voidaan käyttää useaan kertaan erilaisella siirrolla ja orientaatiolla. Esimerkiksi muistissa voisi olla vain yhden puun geometrinen tieto, mutta yksinkertaisten siirtojen ja rotaatioiden avulla voitaisiin muodostaa kokonainen metsä.

Kuvan muodostuksesta on eriteltävissä muutamia yleisiä vaiheita. Puhutaankin 3D-liukuhihnasta, jonka eri vaiheissa kuva koostetaan geometriasta ja kuvadatasta aritmeettisten operaatioiden avulla. Muun muassa Jim Blinn ja Alan Watt ovat kuvanneet tämän prosessin eri vaiheita tarkemmin

kirjoissaan [Blinn, 1996; Watt, 2000]. Nykylaitteisto noudattaa piirrosta pitkälti samoja vaiheita esimerkiksi OpenGL-rajapinnan kautta [Segal *et al.*, 1992a].

2.3. Laitteisto

Ongelmaa ja sen mahdollisia ratkaisuja tarkasteltaessa ei riitä ainoastaan, että sille löydetään *matemaattisesti hieno* ja täydellinen ratkaisu. Koska tavoitteena on grafiikan reaaliaikainen esittäminen, ollaan luonnollisesti kiinnostuneita mahdollisimman tehokkaasta tavasta ratkaista ongelma.



Kuva 3 Laitteiston komponentit

Tietokoneen keskussuoritin (*CPU, Central Processing Unit*) ja erillinen 3D-kiihdytinpiiri (*GPU, Graphics Processing Unit*) muodostavat muistijärjestelmän ja -väylien kanssa (kuva 3) mielenkiintoisen alustan optimoinnille. Järjestelmän voidaan ajatella yleisesti koostuvan näistä kolmesta erillisestä osasta. Ei riitä ainoastaan saada jotain osa-aluetta kuvan piirrosta mahdollisimman tehokkaaksi ajan suhteen, sillä huomioon tulee ottaa koko järjestelmän ja sen komponenttien rinnakkainen toiminta (*parallelism*). Sillä välin kun piirtoon erikoistunut laitteisto suorittaa sille annettua tehtävää, tulisi pääprosessorille antaa jotain *muun laitteiston tehtävien tuloksista riippumatonta* laskentaa: muutoin keskussuoritin joutuu vain odottelemaan laskentatuloksia muulta laitteistolta laskentatehon jäädessä käyttämättä. Sama pätee tietysti myös päinvastoin. Tilannetta toki tasoittaa hiukan se, että kuvia piirretään yleensä jatkuvasti peräkkäin muodostaen animaatiota.

Nykylaitteisto piirtää tehokkaasti kolmioita, ei mitä tahansa monikulmioita. Kappaleiden optimaalisen tehokas piirto vaatii mahdollisimman suurien kolmiomäärien piirron mahdollisimman pienellä määrällä käskyjä, geometrisen datan tullessa mieluiten suoraan 3D-

kiihdyttimellä sijaitsevalta muistilta. Mikäli kappaleiden piirtoa varten täytyy määrittää suuria määriä kolmioita uudestaan jokaista piirrettyä kuvaa varten, kuluu osa ajasta ja tehosta pelkkään tiedon siirtoon muistiväylää pitkin. Koska varjonpiirtoalgoritmien halutaan luonnollisesti olevan niin tehokkaita kuin mahdollista, myös tämä seikka tulee ottaa huomioon toteutuksessa.

2.4. Valonlähteiden tyypit

Valonlähteet jaetaan perinteisessä reaaliaikaisessa tietokonegrafiikassa tyypillisesti kolmeen tai neljään eri tyyppiin [Segal *et al.*, 1992a]. Valaistusalgoritmeilla ei välttämättä ole paljoakaan yhteistä reaali maailman fysiikan valaistusmallien kanssa; tavoitteena on ainoastaan saada uskottavan näköinen valaistus mahdollisimman tehokkaasti. Jos jokin *näyttää* hyvältä, se *toimii*. Yleensä tämä tarkoittaa asioiden karkeaa yksinkertaistamista, mutta laitteistotehon kasvaessa tutkimus ja kiinnostus entistä paremman näköisiä ja monimutkaisempia valaistusmalleja kohtaan on kasvanut. Yleensä valaistusalgoritmit eivät ota huomioon varjoja, toisin sanoen ne ainoastaan valaisevat pintoja, joten varjoalgoritmit ovat eroteltavissa valaistusyhtälöistä. Tässä tutkielmassa ei puututa varsinaisiin valaistusmalleihin. Akenine-Möller *et al.* [2002] kuvaavat kirjassaan muutamia yleisempiä ja nykylaitteistoa hyödyntäviä tekniikoita. Varjojen kannalta merkityksestä on ainoastaan valonlähteen tyyppi.

Ambientti- eli yleisvalaistus simuloi ympäristöstä heijastuvia valonsäteitä (*global illumination*), joiden käsitteleminen olisi liian raskasta eksplisiittisesti. Ambienttivalaistus vaikuttaa jokaiseen kappaleeseen samalla voimakkuudella riippumatta sen sijainnista. Tästä syystä ambienttivaloja ei ole mielekäs olla useampaa yhtä kappaletta kohden. Eri alueilla sijaitseviin kappaleisiin voidaan silti ajatella vaikuttavan eri vahvuinen yleisvalo (vrt. sisä- ja ulkotilat). Ambienttivalolle on mahdotonta määrittellä varjoja perinteisessä tietokonegrafiikan mielessä.

Pistevalo (engl. *omnidirectional point light*) simuloi yhdestä äärettömän pienestä pisteestä jokaiseen suuntaan säteilevää valoa. Varjojen kannalta nämä valonlähteet ovat tässä kuvatuista kaikkein raskaimpia, sillä varjot tulee "heittää" ympäristöön suhteessa valonlähteen sijaintiin. Tämä seikka tarkoittaa yleensä ylimääristä laskentaa.

Suunnatut valonlähteet (engl. *directional light*) valaisevat kappaletta tietystä suunnasta ilman erityistä sijaintia. Valonlähdettä voidaan ajatella äärettömän kaukana jossain suunnassa sijaitsevana pistevalona, jonka intensiteetti ei heikkene etäisyyden myötä. Varjojen muodostus näille valonlähteille voidaan luonnollisesti myös supistaa kappaleesta yhteen suuntaan heittäytyville varjoille.

Neljänneksi valotyyppiä voidaan määrittää kohdevalot (engl. *spot light*), jotka ovat suunnattuja valoja, joiden valaisema alue on rajoitettu kaartiolla. Varjojen kannalta niiden voidaan ajatella käyttäytyvän joko pistevalon tai suunnatun valon tavoin edellä mainitun kartion leveydestä riippuen.

3. Ratkaisumenetelmiä

Alan kehittyessä esiin on noussut muutama hallitseva varjoalgoritmityyppi. Woo *et al.* [1990] esittävät 90-luvun alkuun mennessä kehitetyt tekniikat hyvässä yleiskatsauksessa *A survey of shadow algorithms*. Laitteisto on kuitenkin kehittynyt noista ajoista huomattavasti ja se vaikuttaa merkittävästi siihen miten algoritmit kannattaa käytännössä toteuttaa. Perusperiaatteet ovat silti toki pysyneet samoina.

Esiteltävät menetelmät on jaettu analyttisiin ja kuvapohjaisiin ratkaisuihin. Analyttiset menetelmät pyrkivät muodostamaan varjoalueet 3D-mallien geometriasta ilman kuvatulkinna hyväksikäyttöä. Kuvapohjaiset menetelmät puolestaan eivät käytä hyväkseen tietoa mallin geometrisistä yksityiskohdista vaan perustuvat piirretyn kuvan hyväksikäyttöön.

3.1. Tasolle projisoitavat varjot

Yksi yksinkertaisimmista tavoista toteuttaa varjot on geometrian projisointi jollekin tasolle. Tämä tapa ei luonnollisesti riitä esittämään mielivaltaisen näkymän varjoja, mutta joissain erikoistapauksissa menetelmä saattaa olla *tarpeeksi hyvä* ratkaisu. Kuvitellaan esimerkiksi näkymä, joka käsittää ainoastaan yhden tason (esimerkiksi nurmikko), jonka päällä muut kappaleet (esimerkiksi puita) sijaitsevat. Jokaisen tason päällä sijaitsevan kappaleen heittämät varjoalueet voidaan muodostaa projektiomatriisilla kappaleen geometriasta. Tekniikka ei vaadi laitteistolta mitään lisäominaisuuksia; varjoalueet piirretään samaan tyyliin kuin tavallinenkin geometria.

Olkoon \mathbf{n} tason suunnan määrittävä normaalivektori ja d sen etäisyys origosta, \mathbf{l} valon sijainti ja \mathbf{v} kappaleen kolmioverkkoon kuuluva kärkipiste. Nyt projisoimalla piste \mathbf{v} em. tasolle, saadaan kaava

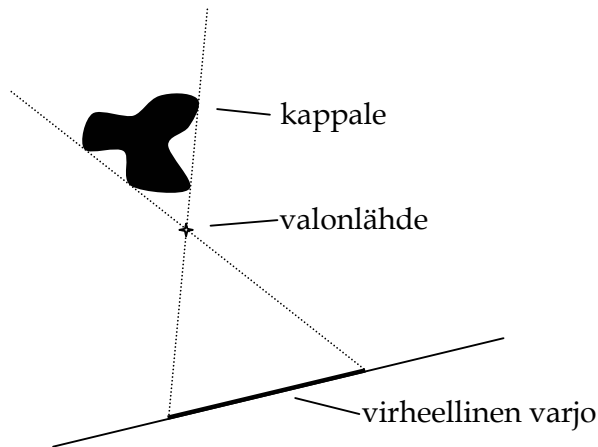
$$\mathbf{v}' = \mathbf{l} - \frac{d + \mathbf{n} \cdot \mathbf{l}}{\mathbf{n} \cdot (\mathbf{v} - \mathbf{l})} (\mathbf{v} - \mathbf{l}),$$

joka on edelleen ilmaistavissa matriisimuodossa $\mathbf{v}' = \mathbf{vM}$,

$$\mathbf{M} = \begin{bmatrix} \mathbf{n} \cdot \mathbf{l} + d - \mathbf{l}_x \mathbf{n}_x & -\mathbf{l}_x \mathbf{n}_y & -\mathbf{l}_x \mathbf{n}_z & -\mathbf{l}_x d \\ -\mathbf{l}_y \mathbf{n}_x & \mathbf{n} \cdot \mathbf{l} + d - \mathbf{l}_y \mathbf{n}_y & -\mathbf{l}_y \mathbf{n}_z & -\mathbf{l}_y d \\ -\mathbf{l}_z \mathbf{n}_x & -\mathbf{l}_z \mathbf{n}_y & \mathbf{n} \cdot \mathbf{l} + d - \mathbf{l}_z \mathbf{n}_z & -\mathbf{l}_z d \\ -\mathbf{n}_x & -\mathbf{n}_y & -\mathbf{n}_z & \mathbf{n} \cdot \mathbf{l} \end{bmatrix}$$

[Akenine-Möller *et al.*, 2002]. Tämä matriisimuotoinen esitys on suoraan yhdistettävissä piirrettävän 3D-mallin kuvausmatriisiin.

Tässä lähestymistavassa on muutamia selviä puutteita. Valonlähteen täytyy sijaita tasoon nähden kappaleiden yläpuolella tai varjojen muodostuksessa esiintyy ongelmia (kuva 4). Varjo saattaa heijastua virheellisesti tasolle vastaprojektion kautta, vaikka valonlähde onkin kappaleen ja tason välissä. Ongelma on helposti vältettävissä yksinkertaisesti kummalla puolella varjotason suuntaista, valonlähteen kautta kulkevaa tasoa kappale on.



Kuva 4 Vastaprojektio-ongelma

Tekniikka siis käytännössä kuvaa 3D-mallin jokaisen kolmion mielivaltaiselle tasolle. Tuloksena on yleensä useita samalla tasolla ainakin osittain päällekkäin olevia monikulmioita. Tällä tavoin muodostetun varjon piirto ei ole täysin mutkatonta. Toki voitaisiin piirtää varjomonikulmiot täysin läpikuultamattomiksi tason päälle, mutta paremman lopputuloksen aikaan saamiseksi yleensä ollaan kiinnostuneita saamaan varjojen alla oleva kuva jäämään osittain näkyviin. Jos taas kaikki kolmiot piirrettäisiin tummentamalla alla olevaa kuvaa hiukan, törmäisimme ongelmiin toistensa kanssa leikkaavien varjokolmioiden kanssa. Ongelma on ratkaistavissa näppärästi esimerkiksi merkkausepuskuria (*stencil buffer*) hyväksikäyttäen merkitsemällä ensin kaikki varjossa olevat kuvapistet ja vasta sen jälkeen käsittelemällä ne.

Kuvattu tekniikka alkaa olla nykypäivänä jo käyttökelpoton sen rajoittuneisuuden vuoksi. Nykystandardit asettavat lisävaatimuksia näyttävyydelle ja pelkkä litteä taso kaiken alustana ei yleensä riitä esittämään haluttua näkymää. Lisäksi, koska tällä tekniikalla toteutetut varjot ovat aina yhdellä tasolla ja täten litteitä, ne eivät tietenkään voi varjostaa kaarevia ei-tasomaisia pintoja. Tekniikkaa voitaisiin kehittää eteenpäin esimerkiksi

heittämällä varjot jokaisen 3D-mallin jokaiselle kolmion määrittämälle tasolle leikaten pois kolmion ulkopuolelle jäävät osat. Tämä lähestymistapa tuottaisi kuitenkin ongelmallisen suuria määriä kolmioita ja kasvattaisi leikkausten laskemiseen kuluvaan aikaan. Laskentaa voitaisiin tosin tehostaa piilopintojen poistoa (*hidden surface removal*) hyödyntävillä tekniikoilla [Aherton *et al.*, 1978]. Seuraavaksi esitetty tekniikka kiertää osan ongelmista siirtämällä tekniikan tekstuuripohjaiseksi.

3.2. Projisoitavat varjotekstuurit

Varjojen esittäminen monikulmioina tuo mukanaan muutamia ongelmia; varjoalueiden muodostaminen riippuu suuresti geometrian kompleksisuudesta tehontarpeen kasvaessa monikulmiomäärien mukana. Eräs toinen vaihtoehto on hyödyntää piirrosta tekstuuripinnoitusta [Segal *et al.*, 1992b].

Seuraavaksi kuvatus algoritmin vaatimuksena on, että valonlähde on suunnattu, mutta varjojen generoiminen myös pistevaloille on mahdollista yhdessä läpiajossa (engl. *pass*) kuutiotekstuureita (engl. *cube maps*) käyttäen [Dietrich, 2001]. Läpiajojen lukumäärällä tarkoitetaan kertoja, jonka näkymän kappaleet tulevat piirretyksi.

Etsitään näkymästä kaikki kappaleparit, joista toinen heittää varjon toiselle. Jokaiselle näille pareille:

1. Tyhjennetään varjotekstuuuri valkoiseksi
2. Piirretään varjoa heittävä kappale valonlähteestä katsottuna varjotekstuurin mustana
3. Generoidaan varjon vastaanottavalle kappaleelle tekstuurikoordinaatit projisoimalla varjotekstuuuri valonlähteestä
4. Piirretään varjoa vastaanottava kappale pinnoitettuna generoidulla tekstuurilla

Listaus 1 Algoritmi varjostetun näkymän piirrosta

Vaiheessa 3. tarvittava tekstuurikoordinaattienmuodostamiskuvaus on johdettavissa kappaleessa 3.1 esitetystä projektiokaavasta.

Tekniikka voi tuottaa melko laadukkaan lopputuloksen, mutta voi toisaalta olla raskas mikäli näkymässä on runsaasti kappaleita. Esitettyä algoritmia voidaan toki tehostaa käsittelemällä useampia kappaleita kerralla pelkkien parien sijasta. Näkymä voidaan jakaa kahteen osaan: varjoja heittäviin

ja varjoja vastaanottaviin pintoihin. Tällä tavalla koko lopullinen kuva pystytään piirtämään kahdessa läpiajossa.

Toinen mahdollinen ongelma koskee käytetyn varjotekstuurin resoluutiota: mitä kauemmaksi varjoa heittävästä kappaleesta varjo lankeaa, sitä tarkempi tekstuurin tulisi olla. Ongelma tulee selvästi esiin pieniresoluutioisilla puskureilla kun katsoja on suoraan valonlähdeä päin ja näkee lähelle heijastuvan varjon. Tällöin yhdelle kuvapisteelle voi olla kuvattuna liian pieni osa varjotekstuurista.

Koska kappale ei voi olla yhtä aikaa varjoa heittävä ja vastaanottava, kappaleet eivät voi varjostaa itseään. Kappaleita esittäviä 3D-malleja voitaisiin toki rikkoa useampiin osiin varjosilhuetin kohdalta ja käsitellä syntyneet kappaleet erikseen, mutta tämä vaatisi luonnollisesti lisää laskentaa. Ongelmaan on olemassa myös tehokkaampia ratkaisutapoja, kuten seuraavassa luvussa esitetty tekniikka.

3.3. Varjosyvyyspuskurit

Varjosyvyyspuskurit (engl. *shadow depth buffer*) hyödyntävät tietokonegrafiikassa yleisiä syvyyspuskureita (engl. *depth buffer*), jotka ovat olleet jo pitkään laitteistopohjaisesti tuettuina. Tekniikkaa käyttävä varjoalgoritmi kuvattiin jo 70-luvulla [Williams, 1978]. Menetelmää on kehitetty sen jälkeen eteenpäin [Reeves *et al.*, 1987], mutta yksinkertainen perusperiaate on pysynyt samana. Silti vasta viimeaikoina on ollut mahdollista toteuttaa algoritmi mielekkäästi laitteistokiihdytettynä massamarkkinoilla saatavilla olevilla laitteilla. Ensimmäinen toteutus lienee ollut NVIDIAN GeForce3-näytönohjainpiiri. SGI:n kehittämät *Reality Engine*- ja *Infinity Reality* -systemit mahdollistivat algoritmin toteutuksen laitteistopohjaisena jo aiemmin, mutta hintansa vuoksi ne eivät ikinä yleistyneet raskaan työasemakäytön ulkopuolella.

Syvyyspuskuri sisältää jokaiselle kuvapisteelle tiedon siitä kuinka kaukana se on katsojasta. Perustapauksessa jokaista kuvapistettä piirrettäessä verrataan sen syvyysarvoa syvyyspuskurissa olevaan arvoon ja hylätään piirrettävä kuvapiste, mikäli sen syvyysarvo on suurempi kuin verrokkiarvo. Uusi arvo kirjoitetaan syvyyspuskuriin. Käytännössä siis vain lähimmät kuvapistet jäävät näkyviin peittäen takanaan olevat pisteet. Lämpikuultavuuksien käsittely luonnollisesti monimutkaistaa prosessia. Nykyrajapinnat mahdollistavat syvyystestin määrittelyn tarkemmin kuin esimerkkitapauksen pienempi-kuin -vertailu [Segal *et al.*, 1992a].

Varjosyvyyspuskurit ovat valon näkökulmasta muodostettuja syvyyspuskureita. Käytännössä puskuriin lasketaan kuinka pitkälle valonsäde ehtii matkata ennen kuin törmää johonkin. Varsinaista näkymää piirrettäessä,

jokainen kuvapiste kuvataan pisteeksi varjosityyppuskurissa ja verrataan kuvattua syvyyssarvoa puskurin syvyyssarvoon. Mikäli varjosityyppuskurissa oleva arvo on pienempi kuin kuvapisteen kuvattu syvyyssarvo, on piste valon ja kappaleen takana – eli varjossa. Kuvapisteen kuvausta varjosityyppuskuriin voidaan ajatella projektiona kappaleessa 3.2 esitellyn kaavan tapaan. Koko algoritmi varjoalueiden muodostamisesta yhdelle valonlähteelle on kuvattu korkealla tasolla listauksessa 2. Tarkempaa tietoa tehokkaasta käytännöntoteutuksesta on saatavilla muun muassa laitteistovalmistajilta [Everitt, 2001; Kilgard, 2001b].

-
1. Piirretään näkymän syvyyssiedot varjosityyppuskuriin valonlähteen kuvakulmasta.
 2. Piirretään näkymän syvyyssiedot syvyysspuskuriin.
 3. Piirretään näkymä käyttäen varjosityyppuskuria valonlähteestä projisoituna tekstuurina. Kuvapistettä piirrettäessä verrataan kuvapisteen syvyyssarvoa projisoituun varjosityyppuskurin syvyyssarvoon: jos ja vain jos varjosityyppuskurissa on pienempi arvo kuin syvyysspuskurissa, kuvapiste on varjossa.

Listaus 2 Varjosityyppuskurialgoritmi varjostetun näkymän piirrosta

Käytännössä listauksen 2 vaiheessa 3 tehtävässä vertailussa tarvitaan pieni poikkeama (engl. *bias*) pois päin valonlähteestä, sillä muutoin kappaleiden varjoja heittävät pinnat saattaisivat epätoivotusti heittää varjon itselleen.

Algoritmia vaivaavat samat heikkoudet kuin projisoituja varjotekstuureitakin. Varjosityyppuskurin resoluution täytyy olla riittävä koko kuva-alalla riippumatta katselukulmasta. Tekniikka mahdollistaa kappaleiden itsevarjostuksen ja se soveltuu mainiosti esimerkiksi tietokoneanimaatio-elokuvissa käytettäväksi. Niissä kuvakulmat ja puskurien koko voidaan määrittää tarkalleen halutuiksi: jos kuvaa ei ikinä tulla katsomaan tarkkuuden kannalta huonosta kulmasta, voidaan resoluutiota tiputtaa – ja tietysti päinvastoin. Virtuaalitodellisuussovelluksissa, joissa halutaan mahdollistaa vapaa mielivaltainen katselukulma ja valon sijainti, törmätään sen sijaan usein ongelmiin. Suuriresoluutioiset puskurit vievät runsaasti muistia ja ovat hitaampia generoita. Ongelman vaikutusta voidaan lieventää rajoittamalla puskurin sisällön vain kuva-alan kattavaksi, mutta tällöin syvyysspuskuri

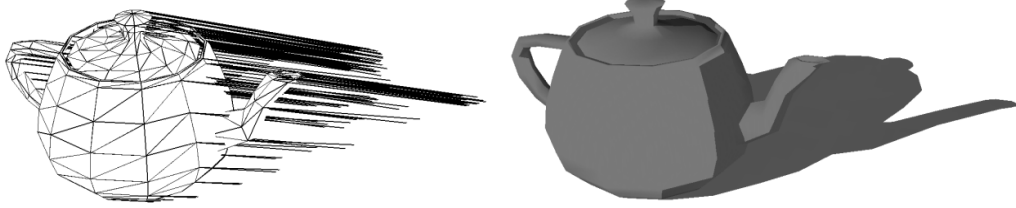
joututaan muodostamaan uudestaan jokaista kuvaa varten [Fernando *et al.*, 2001].

3.4. Varjotilavuudet

Frank Crow esitteli idean varjotilavuuksista (engl. *shadow volume*) 70-luvun lopun SIGGRAPH-julkaisuissa [Crow, 1977]. Merkkauspuskurit mahdollistavat algoritmin luontevan toteutuksen OpenGL-rajapinnalla [Heidmann, 1991]. Algoritmin toteutus vaatii 3D-mallin geometrian käsittelyä aina valonlähteiden ja kappaleiden suhteellisten sijaintien ja orientaatioiden muuttuessa. Tämä tarkoittaa yleensä tiedonsiirtoa CPU:n ja GPU:n välillä. Eräs algoritmin versio on esitetty listauksessa 3. Tämä lienee yksinkertaisin versio algoritmista. Menetelmä tarvitsee käyttöönsä merkkauspuskurin, jossa on ainoastaan yksi tai useampi bitti per kuvapiste. Algoritmi ei toimi oikein mikäli varjotilavuudet menevät päällekkäin ja ongelmia esiintyy myös mikäli varjotilavuudet ovat kameran projektiotasoa lähempänä katsojaa. Nämä rajoitukset on rikottu jatkokehitellyissä versioissa [Bilodeau *et al.*, 1999; Carmack 2000]. Kuvassa 5 olevan kuuluisan teekannun varjot on luotu tällä jatkokehitellyllä menetelmällä.

-
1. Muodostetaan varjotilavuutta kuvaava geometria
 - a. Siirretään varjoja heittävä valonlähde malliavaruuteen
 - b. Merkitään 3D-mallin jokaiselle monikulmiolle onko se valoon päin vai valosta poispäin oleva.
 - c. Etsitään monikulmioiden väliset reunat, joissa toinen monikulmioista on valoon päin oleva ja toinen siitä poispäin oleva: nämä reunat ovat mahdollisia silhuettireunoja.
 - d. Luodaan silhuettireunoille nelikulmiot, joista toinen on silhuettireunalla ja toinen on projisoituna valonlähteestä poispäin äärettömyyksiin.
 2. Tyhjennetään merkkauspuskuri nolllalla
 3. Piirretään näkymän syvyysarvot syvyyspuskuriin
 4. Piirretään varjotilavuutta kuvaava geometria merkkauspuskuriin muuttaen merkkauspuskurin bitit käänteisiksi jos syvyydesti läpäistään
 5. Merkkauspuskurin kuvapisteet, joiden arvo on erisuuri kuin 0 ovat varjossa

Listaus 3 Eräs algoritmi varjoalueiden muodostuksesta varjotilavuuksien avulla



Kuva 5 Varjotilavuusgeometriaa ja näkymä varjojen kanssa

Yksi varjotilavuuksien ongelmista liittyy tiedonvälitykseen pääsuorittimelta piirtämiseen erikoistuneelle laitteistolle. Varjosilhuettien määrittäminen nykylaitteistolla vaatii käytännössä pääsuorittimella tehtävää laskentaa ja generoi pahimmillaan runsaita määriä geometriaa. On olemassa menetelmiä, joilla voidaan siirtää osa laskennasta piirtämiseen erikoistuneelle laitteistolle, mutta ongelmaa ei ole ratkaistu tehokkaasti kokonaisuudessaan [Kilgard, 2001b]. Mark Kilgard on käsitellyt syvällisemmin algoritmin tehokasta toteutusta OpenGL-rajapinnalla ja nykylaitteistolla [Kilgard, 2001b].

Toinen ongelma on kappaleiden varjoja heittävät reunasilhuettilla olevat monikulmiot. Monikulmio joko on varjoa heittävä tai ei ole. Tästä seuraa artefakteja kappaleiden varjoja heittäville reunoille. Valaistuksen kanssa ongelma ei ole niin näkyvä, mutta siltikin pahimmillaan häiritsevä. Ongelman voisi kiertää lisäämällä malleihin riittävän paljon kolmioita esimerkiksi alijakomenetelmillä, mutta tämä ei yleensä ole tehokkuuden kannalta mielekästä. Reunoja voisi myös pehmentää piirtämällä niille valaistussyhtälön mukaan osittain läpikuultavia monikulmioita. Tämä luonnollisesti hidastaisi algoritmin toimintaa, mutta parantaisi lopputulosta.

3.5. Muita menetelmiä

Edellä kuvatut menetelmät ovat yleisimpiä laitteistokiihdytettynä toteutettavia menetelmiä. Kuvattuja tekniikoita on myös yhdistelty ja kehitetty eteenpäin [Assarson *et al.*, 2003; McCool, 2001].

On olemassa myös aivan toisen tyyppisiä menetelmiä. Säteenjäljitykseen (engl. *raytracing*) perustuvat tekniikat hoitavat varjojen muodostuksen suoraan tuloksena säteen jäljityksestä – jos jokin estää valon pääsyn pisteeseen, piste on varjossa. Säteenjäljitys on varsin ongelmallista toteuttaa laitteistokiihdytettynä nykylaitteiston ja -rajapintojen luonteen vuoksi. Tulevaisuudessa laitteiston ominaisuuksien kehittyessä säteenjäljitys saattaa silti olla tehokkaasti toteutettavissa [Purcell *et al.*, 2002].

Eräs toinen varjostusta huomioon ottava tekniikka perustuu palloharmonisiin funktioihin (engl. *spherical harmonics*). Menetelmä on toteutettavissa rajoittuneesti laitteistokiihdytettynä [Jensen *et al.*, 2001]. Tämä tekniikka ei kuitenkaan ole kovin dynaaminen, sillä kappaleiden sijaintien ja orientaatioiden muuttaminen vaatii raskasta esilaskentaa.

4. Lopuksi

Nykylaitteistot mahdollistavat muutamien erilaisten varjonpiirtoalgoritmien toteutuksen tehokkaasti reaaliajassa. Tulevaisuudessa yhä suurempi osa laskennasta siirtyy tietokoneen pääsuorittimelta piirtämiseen erikoistuneelle laitteistolle. Optimitilanteessa tiedonvälitys tämän laitteiston ja pääsuorittimen välillä tulisi minimoida; tämä asia täytyy ottaa huomioon varjoalgoritmien toteutuksessa. Tästä seikasta johtuen nykylaitteistolla on vaikea saada toteutettua tehokkaasti esimerkiksi varjotilavuuspohjaisia algoritmeja. Mikään kuvatuista algoritmeista ei ole täydellinen. Toisaalta tulevaisuudessa laitteistot saavat uusia ominaisuuksia ja mahdollistavat jälleen erityyppisten algoritmien mielekkään toteutuksen. Erityisesti säteenjäljitykseen perustuvat tekniikat voisivat tarjota laitteistokiihdytettynä mielenkiintoisen jatkotutkimusaiheen.

Viiteluettelo

- [Aherton *et al.*, 1978] Peter Aherton, Kevin Weiler, Donald Greenberg, Polygon shadow generation, *Proceedings of International Conference on Computer Graphics and Interactive Techniques* (1978), 275-281.
- [Assarson *et al.*, 2003] Ulf Assarsson, Tomas Akenine-Möller, A geometry-based soft shadow volume algorithm using graphics hardware, *ACM Transactions on Graphics (TOG)*, July 2003, **22** (3).
- [Akenine-Möller *et al.*, 2002] Tomas Akenine-Möller, Eric Haines, *Real-time Rendering (2nd edition)*, A K Peters Ltd, 2002.
- [Bilodeau *et al.*, 1999] Bill Bilodeau, Mike Songy, Creative Labs sponsored game developer conference, unpublished slides, Los Angeles, May 1999.
- [Blinn, 1996] Jim Blinn, *Jim Blinn's Corner: A Trip Down the Graphics Pipeline*, Morgan Kaufmann Publishers, 1996.
- [Carmack, 2000] John Carmack, Carmack on shadow volumes, correspondence between Mark Kilgard and John Carmack, 2000.
- [Crow, 1977] Frank Crow, Shadow algorithms for computer graphics, *Proceedings of SIGGRAPH*, (1977), 242-248.

- [Dietrich, 2001] Sim Dietrich, Shadow techniques, *Game Developers Conference 2001*.
- [Eberly, 2001] David Eberly, *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*, Morgan Kaufmann Publishers, 2001
- [Everitt, 2001] Cass Everitt, Shadow Mapping, *NVIDIA Developer Relations Website* (saatavilla http://developer.nvidia.com/object/shadow_mapping.html) (tarkistettu 24.10.2003)
- [Fernando *et al.*, 2001] Randima Fernando, Sebastian Fernandez, Kavita Bala, Donald Greenberg, Adaptive Shadow Maps, *Proceedings of SIGGRAPH 2001*.
- [Heidmann, 1991] Tim Heidmann, Real shadows, real time, Iris Universe, Silicon Graphics Inc., November 1991.
- [Kilgard, 2001a] Mark Kilgard, Robust Stencil Shadow Volumes, *CEDEC 2001*.
- [Kilgard, 2001b] Mark Kilgard, Shadow Mapping with Today's OpenGL hardware, *CEDEC 2001*.
- [McCool, 2001] Michael McCool, Shadow volume reconstruction from depth maps, *ACM Transactions on Graphics*, Jan. (2001), pp. 1-25.
- [Jensen *et al.*, 2001] Henrik W. Jensen, Steve R. Marschner, Marc Levoy, Pat Hanrahan, A practical model for subsurface light transport, *SIGGRAPH 2001*.
- [Purcell *et al.*, 2002] Timothy J. Purcell, Ian Buck, William R. Mark, Pat Hanrahan, Ray tracing on programmable graphics hardware, *International Conference on Computer Graphics and Interactive Techniques*, 2002.
- [Reeves *et al.*, 1987] William Reeves, David Salesin, Robert Cook, Rendering antialiased shadows with depth maps, *SIGGRAPH 87*, 1987.
- [Segal *et al.*, 1992a] Mark Segal, Kurt Akeley, OpenGL 1.1 specification, 1992.
- [Segal *et al.*, 1992b] Mark Segal, Carl Korobkin, Rolf van Widenfelt, Jim Foran, Paul Haeberli, Fast shadows and lighting effects using texture mapping, *SIGGRAPH 92*
- [Watt, 2000] Alan Watt, *3D Computer Graphics (3rd edition)*, Addison-Wesley, 2000.
- [Williams, 1978] Lance Williams, Casting curved shadows on curved surfaces, *Computer Graphics*, **12** (3) (1978), 270-274
- [Woo *et al.*, 1990] A. Woo, P. Poulin, and A. Fournier, A survey of shadow algorithms, *IEEE Computer Graphics and Applications*, Nov. (1990), 13-32.

Sykeinformaation käyttömahdollisuudet käyttäjän emootioiden tunnistamisessa

Jenni Anttonen

Tiivistelmä

Emootiot vaikuttavat ihmisen rationaaliseen käyttäytymiseen. Siksi on tärkeää tutkia, kuinka emotionaalisesti merkityksellistä informaatiota voidaan käyttää ihmisen ja tietokoneen vuorovaikutuksen parantamiseen. Tunteet huomioon ottava teknologia mukauttaa toimintaansa käyttäjän emootioiden perusteella. Mukauttaakseen toimintaansa järjestelmän on kyettävä havaitsemaan emotionaalisesti merkityksellistä informaatiota. Yksi menetelmä emotionaalisen informaation havaitsemiseen on fysiologisten signaalien käyttö. Tutkin sydämen sykkeen käyttömahdollisuuksia käyttäjän emootioiden tunnistamisessa ihmisen ja tietokoneen vuorovaikutuksessa. Esittelen uudenlaisen sykkeen mittaamenetelmän ja tutkimussuunnitelman sen validoimiseksi kokeellisella tutkimuksella.

Avainsanat ja -sanonnat: Emootiot, tunteet huomioon ottava teknologia, hahmottamiskykyiset käyttöliittymät, fysiologiset käyttöliittymät, sydämen syke.

CR-luokat: H.5.2

1. Johdanto

Viime aikoina on alettu pohtia emootioihin liittyvien tekijöiden merkitystä ihmisen ja tietokoneen vuorovaikutuksessa. Psykologisen tutkimuksen perusteella tiedetään, että emootioilla on keskeinen merkitys kognitiivisissa toiminnoissa. Emootiot ovat osallisia muun muassa tarkkaavaisuudessa, havaitsemisessa, muistitoiminnoissa, oppimisessa, päätöksenteossa, luovassa ajattelussa ja sosiaalisessa vuorovaikutuksessa [Davidson, 2003; Oatley and Jenkins, 1996, pp. 251-284; Schulkin *et al.*, 2003; Davidson and Cacioppo, 1992; Zajonc, 1980]. Tämän vuoksi on tärkeää tutkia, kuinka emotionaalisesti merkityksellistä informaatiota voidaan käyttää ihmisen ja tietokoneen vuorovaikutuksen parantamiseen. *Tunteet huomioon ottava teknologia* (affective computing) mukauttaa toimintaansa vuorovaikutuksen kuluessa tulkitsemalla käyttäjästä tekemiään emootioihin liittyviä havaintoja ja päätelmiään kontekstista. Turk ja Robertson [Turk and Robertson, 2000] esittelivät käsitteen *hahmottamiskykyiset käyttöliittymät* (perceptual user interfaces, PUIs). Heidän

mukaansa hahmottamiskykyisten käyttöliittymien keskeisenä motivaationa on kehittää yleisempiä, luonnollisempia ja intuitiivisempia tapoja olla vuorovaikutuksessa tietokoneen kanssa. Osa tällaisista käyttöliittymistä hyötyisi myös käyttäjän emootioihin liittyvien tekijöiden havaitsemisesta ja ymmärtämisestä.

Jotta järjestelmä voisi olla emootioihin reagoiva, sen on kyettävä havaitsemaan emotionaalista informaatiota käyttäjästä. Yksi menetelmä käyttäjän emootioiden havaitsemiseen ja tunnistamiseen on fysiologisten signaalien käyttö. *Fysiologiset käyttöliittymät* (physiological computing, physiological user interfaces, electrophysiological HCI) hyödyntävät käyttäjän fysiologisia signaaleja syötteenään [Allanson, 2002; Allanson and Wilson, 2002]. Yksi autonomisen hermoston ohjaama fysiologinen signaali on sydämen syke. Sykettä on käytetty ihmisen ja tietokoneen vuorovaikutuksessa laajasti käyttäjän kognitiivisen kuormituksen, stressin ja ahdistuksen monitoroinnissa [Hilbert and Redmiles, 1999, p. 390; Rowe *et al.*, 1998; Wastell 1990], mutta sitä voidaan käyttää myös käyttäjän emootioiden havaitsemisessa ja tunnistamisessa.

Tutkimukseni laajempaan tavoitteena on tutkia ihmisen ja tietokoneen vuorovaikutuksen kehittämiseen soveltuvia mittausmenetelmiä, joilla voidaan havaita käyttäjän emootioihin liittyviä fysiologisia muutoksia. Fysiologisia signaaleja on tärkeä tutkia psykologisen perustutkimuksen lisäksi nimenomaan ihmisen ja tietokoneen vuorovaikutuksen näkökulmasta. Tutkin sykkeen ja emootioiden riippuvuutta sekä sykkeen käyttökelpoisuutta ihmisen ja tietokoneen vuorovaikutuksessa. Esittelen BKG-tuolin, joka on uudenlainen häiritsemätön (unobtrusive) sykkeen mittausmenetelmä. Raportoin myös tutkimussuunnitelman BKG-tuolin validoimiseksi.

Seuraavassa luvussa kerron, mitä emootiot ovat esittelemällä niiden määritelmiä ja funktioita. Kolmannessa luvussa tarkastelen, mitä tunteet huomioon ottava teknologia on ja miksi ymmärrystä käyttäjän emootioista tarvitaan ihmisen ja tietokoneen vuorovaikutuksessa. Neljännessä luvussa käsittelen psykofysiologisten signaalien hyödyntämistä käyttöliittymissä ja erityisesti sydämen sykettä, joka on tässä tutkimuksessa käytettävä psykofysiologinen signaali. Viidennessä luvussa esittelen tutkimukseen kuuluvan kontrolloidun kokeen suunnitelman. Lopuksi pohdin, mitä merkitystä tutkimuksen tuloksilla mahdollisesti on.

2. Mitä emootiot ovat?

Tässä luvussa esittelen aluksi emootioiden määritelmiä ja funktioita. Käsittelem myös dimensionaalista emootiomallia, joka on tutkimukseni keskeinen teoreettinen lähtökohta.

2.1. Emootioiden määritelmiä ja funktioita

Sana emootio juontaa juurensa latinan kieleen, jossa verbi *emovere* tarkoittaa liikkua, liikuttaa ja siirtää ulos- tai eteenpäin. Emootioista eli tunnereaktioista ei ole yhtä selkeää määritelmää, josta kaikki tutkijat olisivat yhtä mieltä. Nesse [1990] toteaa, että biologisia systeemejä kuvataan usein niiden funktioiden kautta. Hänen määritelmänsä emootioille pohjautuukin emootioiden evolutiivisiin funktioihin [Nesse 1990, p. 268]¹:

”Emootiot ovat luonnonvalinnan muovaamia erikoistuneita toimintaohjelmia, joiden tarkoituksena on säätää organismin fysiologisia, psykologisia ja behavioraalisia reaktioita sillä tapaa, että kapasiteetti ja pyrkimys reagoida adaptiivisesti niin uhkaaviin kuin mahdollisuuksia tarjoaviin tilanteisiin lisääntyä.”

Evolutiivisesta näkökulmasta katsottuna emootioiden nopea viriäminen on ollut tärkeää yksilön selviytymisen kannalta [Dimberg, 1997]. Vaaran uhatessa kognitiivinen päättely on liian hidasta ja siksi tarvitaan emootioita, jotka ohjaavat toimintaa nopeasti.

Suurin osa emootiotutkijoista on yksimielisiä siitä, että emootioihin kuuluu kokemuksellisia, fysiologisia ja ekspressiivisiä komponentteja ja että ne ovat adaptiivisia prosesseja, joilla on biologinen perusta [Oatley and Jenkins, 1996, pp. 95-132; Zajonc, 1980]. Emootioita aiheuttavat tietoisesti tai tiedostamattomasti yksilön kannalta merkityksellisiksi arvioidut tapahtumat. Emootioon liittyy keskeisesti muutos toimintavalmiudessa ja tendenssi tietynlaiseen käyttäytymiseen. Siten emootiot ohjaavat yksilön toimintaa kohti tiettyjä tavoitteita ja päämääriä priorisoimalla tietynlaista toimintaa ja käyttäytymistä. [Oatley and Jenkins, 1996, pp. 95-132] Emootiot ovat läheisessä yhteydessä myös motivaatioon. Emootio on hyvin lyhytkestoinen ilmiö. Oatleyn ja Jenkinsin [Oatley and Jenkins, 1996, pp. 124-125] mukaan emootio on lyhytkestoisempi kuin tunnetila (*feeling*) tai mieliala (*mood*) kestäen yleensä sekunneista korkeintaan muutamaan minuuttiin. Mielialat ja tunnetilat eroavat

¹ Suomennos: Jari K. Hietanen, Neuropsykologian luento 26.3.2002 klo 13.00-14.00, Psykonet, Tampereen yliopisto, Psykologian laitos.

keston lisäksi emootioista myös siten, että niihin ei liity toimintavalmiuden muutosta. [Oatley and Jenkins, 1996, pp. 96-97]

2.2. Emootiomalleja

Emootioiden suhteita toisiinsa ja niiden sijoittumista emootioavaruuteen (affective space) on tutkittu paljon. Emootioavaruuden rakennetta on mallinnettu pääasiassa kahdella tavalla. Ensimmäinen lähestymistapa on diskreetti malli ja toinen dimensionaalinen malli [Izard and Ackerman, 2000]. Diskreetin mallin mukaan emootioavaruus koostuu erillisistä perusemootioista [Ekman *et al.*, 1983; Tomkins, 1982]. Emootiotutkijat sisällyttävät perusemootioihin eri määrän hieman eriäviä emootioita. Ortony ja Turner [Ortony and Turner, 1990] listasivat kirjallisuudessa yleisimmin esiintyviä perusemootioita, joihin kuului muun muassa viha, pelko, suru, ilo, hämmästyminen, inho, halveksunta, häpeä ja syyllisyys. Ekmanin [1992, 1999] mukaan perusemootioiden ominaisuuksia ovat muun muassa tunnusomainen universaali ilmaisu, emootiokohtainen fysiologia ja biologinen perusta. Perusemootioiden universaalius tarkoittaa sitä, että niitä tuotetaan ja tunnistetaan kaikissa kulttuureissa ja niitä voi havaita ihmisen lisäksi myös muissa kädellisissä [Ekman, 1992, 1999]. Kullakin perusemootiolla katsotaan olevan yksilön selviytymistä tietyssä tilanteessa edistävää adaptiivinen funktio [Ekman *et al.*, 1983; Plutchik, 1980]. Kaikki tutkijat eivät ole yksimielisiä perusemootion määritelmästä eikä edes perusemootioiden olemassa olosta [Ortony and Turner, 1990].

Dimensionaalisisessa mallissa emootiot sijoittuvat kaksi- tai useampiulotteiseen jatkuva-asteikolliseen emootioavaruuteen. Yleisimmin ulottuvuuksina ovat

- *valenssi* (valence), joka ulottuu positiivisesta negatiiviseen tai miellyttävästä epämiellyttävään, ja
- *kiihtyneisyys* tai virittyneisyys (arousal), joka ulottuu rauhallisesta kiihtyneeseen [Cacioppo and Gardner, 1999; Lang *et al.*, 1998; Bradley and Lang, 1994].

Cacioppo ja Gardnerin [Cacioppo and Gardner, 1999] mukaan emootioiden taustalla on systeemi, joka kategorisoi merkitykselliset ärsykkeet joko lähestyttäväksi tai vältettäväksi ja ohjaa yksilön toimintaa sen mukaisesti. Tämän systeemin toiminta on niin kriittistä, että tietynlaisten ärsykkeiden kategorisointi sekä ärsykkeen lähestyminen tai välttäminen tapahtuu refleksiinomaisesti. Ihmisen toiminnassa on erityistä kuitenkin se, kuinka oppiminen ja kognitio vaikuttavat kategorisointeihin. Diskreetti ja dimensionaalinen malli eivät ole toisensa poissulkevia vaan diskreetit perusemootiot voidaan sijoittaa tiettyihin kohtiin dimensionaalisisessa mallissa

[Christie and Friedman, in press; Izard and Ackerman, 2000; Mayne and Ramsey, 2001, p. 3].

Diskreetti emootiomalli on tutkimukseni teoreettisena lähtökohtana osaltaan siksi, että diskreettien emootioiden tunnistaminen fysiologisista signaaleista ei ole niin yksikäsitteistä ja varmaa kuin emootioihin liittyvien ulottuvuuksien tunnistaminen [Cacioppo et al, 2000]. Ihmisen ja tietokoneen vuorovaikutuksessa ei myöskään välttämättä tavoitella puhtaiden perusemootioiden ilmentymien tunnistamista kuten psykologisessa tutkimuksessa. Käyttäjän emootiot saattavat olla paljon monimutkaisempia, sekoittuneempia ja epämääräisempiä kuin perusemootiot. Sen sijaan esimerkiksi valenssi-ulottuvuus on tärkeä esimerkiksi käyttäjän mieltymysten ymmärtämisessä ja kiihtyneisyys taas saattaa liittyä vuorovaikutuksen ongelmakohtiin.

3. Tunteet huomioon ottava teknologia

Tässä luvussa kerron tunteet huomioon ottavasta teknologiasta. Selvitän, mitä tunteet huomioon ottava teknologia on ja miksi ymmärrystä käyttäjän emootioista tarvitaan ihmisen ja tietokoneen vuorovaikutuksessa. Olen suomentanut affective computing -käsitteen tunteet huomioon ottavaksi teknologiaksi. Sana computing tarkoittaa suomeksi laskentaa, tietotekniikkaa ja tietokoneen käyttöä. Valitsin kuitenkin käännökseeni sanan teknologia ensiksi siksi, että laskenta ei viittaa suomen kielessä samalla tavalla tietokoneeseen kuin englanniksi ja toiseksi siksi, että teknologia käsittää itsestään selvästi myös muunlaisia laitteita ja tietokoneita kuin perinteisen pöytämikron. Kirjallisuudessa esiintyy affective computing -ilmaisun kanssa rinnakkain ainakin affective human-computer interaction (HCI) -ilmaisu.

Käyttäjän emootioiden ymmärtäminen vuorovaikutuksessa voi parantaa vuorovaikutuksen laatua ja vaikuttaa tehtävän suorittamiseen. Aulan ja Surakan [Aula and Surakka, 2002] tutkimuksessa koehenkilöille annettiin emotionaalista palautetta ongelman ratkaisun aikana. Palaute oli joko positiivista, negatiivista tai neutraalia. Suoritusajat olivat merkitsevästi lyhyempiä positiivisen kuin negatiivisen palautteen jälkeen. Virhemäärään palaute ei vaikuttanut. Tulos todentaa emootioiden vaikutusta ihmisen toimintaan ja niiden huomioimisen tärkeyttä vuorovaikutuksessa teknologian kanssa. Toisena esimerkkinä mainitsen oppimisovellukset. Kapoorin ja kumppaneiden [Kapoor *et al.*, 2001] tutkimuksessa oppijan emootioita seuraamalla säädellään milloin ja miten käyttöliittymäagentti puuttuu oppimisen kulkuun. Käyttöliittymäagentti toimii eri tavoilla esimerkiksi silloin kun oppija on innostunut ja kiinnostunut verrattuna tilanteeseen, jossa oppija

on kyllästynyt tai hermostunut. Tällainen oppijan tunteiden huomioon ottaminen voi parantaa oppimisen tehokkuutta, laatua ja mielekkyyttä.

Tunteet huomioon ottava teknologia on yksi askel kehityksessä, jossa tietokone on vuorovaikutuksessa mukautuva osapuoli käyttäjän sijasta. Tunteiden huomioonottaminen käyttöliittymissä kehittää ihmisen ja tietokoneen vuorovaikutusta myös luonnolliseksi ja intuitiiviseksi, koska emootioilla on keskeinen merkitys inhimillisessä vuorovaikutuksessa [Hudlicka, 2003; Pentland, 2000; Picard, 1997]. Kun ihmisen ja tietokoneen vuorovaikutus perustuu siihen, miten ihmiset ovat vuorovaikutuksessa toistensa ja ympäröivän todellisuuden kanssa, käyttöliittymässä voidaan hyödyntää ihmisillä valmiina olevia taitoja, ihmisten luonnollisia vuorovaikutustapoja ja modaaliteetteja [Turk and Robertson, 2000; Pentland, 2000]. Emootioilla on tärkeä rooli ihmisten välisessä sosiaalisessa vuorovaikutuksessa ja ne ovat usein keskeinen osa viestin sisältöä [Zajonc, 1980, p. 153].

Picardin [1995] määritelmän mukaan tunteet huomioon ottava teknologia on "emootioihin liittyvää, emootioista juontuvaa tai tahallisesti emootioihin vaikuttavaa teknologiaa". Tämä määritelmä pohjautuu suoraan englannin kielen sanakirjan määritelmään sanasta *affective* [Hollnagel, 2003]. Käyttäjän emootioiden huomioon ottamiseen liittyy emotionaalisen informaation kommunikoiminen järjestelmälle, tämän informaation tunnistaminen ja sen pohjalta reagoiminen mukauttamalla järjestelmän toimintaa [Picard, 1995; Carson and Picard, 2001; Hudlicka, 2003]. Tunteet huomioon ottavaa teknologiaa edustaa myös esimerkiksi sosiaalisesti älykäs agentti, joka saattaa käyttäjän emootioiden ymmärtämisen lisäksi tuottaa emotionaalisia ilmaisuja muun muassa dialogin sisällön, eleiden tai puheen akustisten piirteiden kautta [Picard, 1997, p. 50, 56; Johnstone and Scherer, 2000; Hudlicka, 2003].

Tunteet huomioon ottavan teknologian ei tarvitse kokea emootioita, kuten Avaruusseikkailu 2001:n Hal [Kubrick, 1968] tai A.I. Tekoäly -elokuvan [Spielberg, 2001] robottipoika David. Sen sijaan tunteet huomioon ottavalla teknologialla tulisi olla jonkinlainen kyky päätellä käyttäjän emootioihin liittyviä asioita, jotta se voi ottaa käyttäjän emootioita huomioon vuorovaikutuksessa. Tunteet huomioon ottava teknologia havaitsee käyttäjän emootioita jollakin menetelmällä ja mahdollisesti myös tuottaa emotionaalisia ilmaisuja. Kaikkien järjestelmien ei tarvitse eikä kannata olla tunteet huomioon ottavia, ja onkin syytä miettiä, minkälaiset sovellukset hyötyisivät käyttäjän tunteiden huomioimisesta tai jopa edellyttäisivät sitä.

Tunteet huomioon ottava teknologia on siis käyttäjän emootioihin reagoivaa ja käyttäjään mukautuvaa. Jotta käyttöliittymä voisi mukauttaa toimintaansa

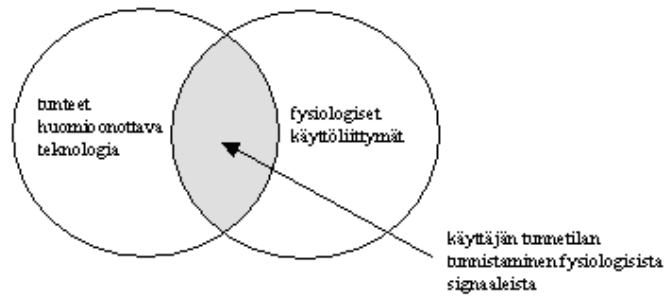
käyttäjän emootioiden perusteella, sillä täytyy olla jonkinlainen emotionaalinen havaintokyky. Yksi mahdollisuus emotionaalisen informaation havaitsemisessa on fysiologisten signaalien käyttö. Seuraavassa luvussa kerron tarkemmin fysiologian hyödyntämisestä käyttöliittymissä ja erityisesti käyttäjän emootioiden tunnistamisesta.

4. Fysiologiset käyttöliittymät

Seuraavaksi selvitän, mitä fysiologisten signaalien käyttö ihmisen ja tietokoneen vuorovaikutuksessa tarkoittaa ja mitä etuja fysiologisilla käyttöliittymillä on. Luvun lopussa käsitelen tarkemmin sydämen sykettä ja sen soveltuvuutta käyttäjän emootioiden tunnistamiseen.

4.1. Mitä fysiologiset käyttöliittymät ovat?

Fysiologiset käyttöliittymät hyödyntävät syötteenään käyttäjän fysiologisia signaaleja [Allanson, 2002; Allanson and Wilson, 2002]. Fysiologisia signaaleja ovat muun muassa sydämen syke, ihon sähkönjohtokyky, kasvolihasten sähköinen aktiviteetti (EMG, elektromyografia) ja pupillin koon muutokset. Allanson [2002] toteaa, että fysiologisten signaalien hyödyntäminen käyttöliittymissä ulottaa ihmisen ja tietokoneen vuorovaikutuksen käyttäjän tarkoituksellisen manipuloinnin ulkopuolelle. Tämä tarkoittaa, että käyttöliittymää voidaan ohjata tai se voi mukautua käyttäjään ilman käyttäjän tarkoituksellista tai tietoista komentoa. Toki fysiologiset käyttöliittymät voivat hyödyntää myös tietoisesti tuotettuja fysiologisia signaaleja esimerkiksi komentojen antamisessa. Tietoisesti tuotettuja fysiologisia signaaleja on käytetty syötteenä esimerkiksi osoitintekniikassa, jossa kursoria ohjataan katseella ja valinta tehdään aktiivomalla kasvolihaksia [Partala *et al.*, 2001] tai aivokäyttö-liittymissä [Kübler *et al.*, 2001; Lusted and Knapp, 1996]. Fysiologisia signaaleja on käytetty ihmisen ja tietokoneen vuorovaikutuksessa pitkään, mutta viime aikoina niitä on alettu hyödyntää myös käyttäjän emootioiden tunnistamisessa. Fysiologiset käyttöliittymät eivät siis aina ole tunteet huomioonottavia järjestelmiä (Kuva 1).



Kuva 1. Tunteet huomioon ottava teknologia ja fysiologiset käyttöliittymät.

Johdannossa kerroin, että hahmottamiskykyisten käyttöliittymien keskeisenä tavoitteena on kehittää ihmisen ja tietokoneen vuorovaikutusta luonnolliseksi ja että ollakseen luonnollista ja intuitiivista ihmisen ja tietokoneen vuorovaikutuksen tulisi perustua inhimilliseen vuorovaikutukseen [Turk and Robertson, 2000]. Tämä tarkoittaa, että tietokoneen tulisi ymmärtää inhimilliseen ja sosiaaliseen käyttäytymiseen kuuluvia piirteitä. Inhimilliseen vuorovaikutukseen ei kuitenkaan kuulu fysiologisten signaalien havaitseminen siinä muodossa, missä kone havaitsee niitä. Voimme havaita fysiologisia muutoksia toisessa ihmisessä etenkin vahvojen emootioiden yhteydessä. Fysiologisten reaktioiden havaitseminen tapahtuu kuitenkin pääasiassa näkö- ja kuuloaistin avulla. Inhimillisessä vuorovaikutuksessa emotionaalinen informaatio välittyy pääosin nonverbaalisti, eli esimerkiksi ilmeiden, eleiden tai puheäänien prosodisten piirteiden kautta. Onko fysiologisten signaalien hyödyntäminen käyttöliittymässä sitten vastoin luonnollisuuden tavoitetta?

Mielestäni näin ei ole. Ihmisten käyttämät havaintomuodot kuten näkö, kuulo, tunto, haju, maku ja tasapaino eivät välttämättä ole tietokoneelle tehokkaimmat tavat tehdä havaintoja käyttäjästä ja ympäristöstä. Monet tunteet huomioon ottavan teknologian sovellukset varmasti hyötyvät esimerkiksi käyttäjän nonverbaalien ilmaisujen havaitsemisesta. Tämän voi saavuttaa konenäön avulla, mutta myös esimerkiksi mallintamalla tiettyihin kehoon osiin sijoitettujen, vaatteisiin upotettujen antureiden liikkeitä. Tavoitteena on *vuorovaikutuksen* luonnollisuus ja intuitiivisuus. Se ei tarkoita, että tietokoneen havaitsemismenetelmien täytyisi imitoida ihmisen aisteja tai rajoittua niihin mahdollistaakseen ihmiselle luonnollisen vuorovaikutuksen. Vuorovaikutuksen luonnollisuutta palvelee tietokoneelle mahdollisimman tehokkaiden ja luotettavien havaintomenetelmien kehittäminen. Havaitsemisen on tietokoneen osalta kuitenkin tapahduttava häiritsemättömästi eli siten, että se on käyttäjälle luonnollista ja ergonomista eikä se keskeytä tai häiritse vuorovaikutuksen kulkua. Tästä johtuen esimerkiksi erilaiset anturit on hyvä

olla upotettuina käyttäjän luonnolliseen ympäristöön. Samoin tietokone voi käyttää ilmaisujen tuottamisessa erilaisia menetelmiä kuin ihmiset, jos lopputulos eli vuorovaikutus käyttäjän näkökulmasta on luonnollinen. Käyttäjän fysiologisten signaalien mittaaminen ei mielestäni myöskään sovellu kaikkiin käyttötilanteisiin. Esimerkiksi ryhmäsovelluksessa kasvoniilmeiden mallintaminen saattaa olla parempi vaihtoehto, koska silloin toisille välittyvä emotionaalinen informaatio on tahdonalaisen kontrollin alaisena.

4.2. Käyttäjän emootioiden tunnistaminen fysiologisista signaaleista

Tunteet huomioon ottavassa teknologiassa yksi keskeinen asia on emotionaalisen informaation havaintokyky ja se voidaan siis toteuttaa fysiologisten signaalien avulla. Käyttäjän fysiologisia signaaleja rekisteröimällä pyritään havaitsemaan ja tunnistamaan käyttäjän emootioita vuorovaikutuksen eri hetkillä sekä reagoimaan niihin. Emootioiden tunnistamisessa fysiologisista signaaleista on oletuksena, että autonomisen hermoston kontrolloimat fysiologiset reaktiot ovat emootiokohtaisia, eli kukin emootio on yksi yhteen suhteessa tiettyyn fysiologisen aktivaation yhdistelmään, josta kyseinen emootio voidaan tunnistaa [Zajonc and McIntosh, 1992; Levenson, 1992; Ekman *et al.*, 1983]. Cacioppo *et al.* [2000] toteavat kirjallisuuskatsauksensa perusteella, että erotettavissa oleva emootiokohtainen fysiologia on saatu tulokseksi useissa tutkimuksissa, mutta ehdottomia päätelmiä autonomisen hermoston aktivaation yhdistelmistä eri emootioille ei kuitenkaan voida tämänhetkisen tutkimuksen perusteella tehdä. Katsaukset [Cacioppo *et al.*, 2000; Zajonc and McIntosh, 1992] olemassa olevaan tutkimukseen osoittavat myös, että fysiologinen aktivaatio erottelee luotettavasti etenkin emootioiden valenssia eli positiivinen–negatiivinen -ulottuvuutta.

Voi myös olla, että yksittäisten tunnetilojen luotettava tunnistaminen pelkästään psykofysiologisten signaalien perusteella ei ole mahdollista. Esimerkiksi Ekman [1992] uskoo, että emootiokohtainen fysiologia on kehittynyt vain sellaisille emootioille, joiden aikana tietynlainen motorinen toiminta on yksilön selviytymisen kannalta välttämätöntä, esimerkiksi pakeneminen pelon yhteydessä. On myös muistettava, että kognitiivisella arvioinnilla ja tilannespesifisyydellä on suuri vaikutus koettuihin emootioihin. Cacioppo *et al.* [2000] ehdottaakin, että on hyödyllisempää pohtia, missä tilanteissa ja millä emootioilla emootiokohtaista fysiologiaa voidaan havaita, kuin tavoitella absoluuttisen emootiokohtaisen fysiologian todistamista.

Perusemootioiden kaltaisia emootiota ei välttämättä tarvitse erotella ihmisen ja tietokoneen vuorovaikutuksessa vaan saattaa riittää esimerkiksi erottaa tila, jossa vuorovaikutus sujuu hyvin, tilasta, jossa käyttäjä turhautuu ja hermostuu vuorovaikutuksen kangerrellessa [Klein *et al.*, 2002]. Emootioiden

päättelyä fysiologisista signaaleista vaikeuttaa se, että yksilöiden välinen vaihtelu fysiologisissa signaaleissa on suurta ja tämän lisäksi vaihtelua on myös yksilöiden sisällä jopa saman päivän aikana. Sydämen syketaison vaihtelua saattavat aiheuttaa muun muassa lämpötila, asento, tupakointi, vuorokauden aika ja edellisestä ateriasta kulunut aika [Siddle and Turpin, 1980, p. 154]. Tätä ongelmaa helpottaa hieman se, että osa sovelluksista voidaan opettaa tunnistamaan yhden tietyn käyttäjän fysiologisia muutoksia ja tyypillisiä piirteitä [Picard *et al.*, 2001]. Tällainen vaihtelu ei myöskään ole yhtä kriittistä mitattaessa perustasoon suhteutettuja sykereaktioita tiettyihin ärsykkeisiin kun mitattaessa absoluuttista syketasoa tietyllä aikavälillä. Psykofysiologisen informaation lisäksi ihmisen ja tietokoneen vuorovaikutuksessa voidaan käyttää hyväksi esimerkiksi järjestelmän tilasta tai vaikkapa käyttäjän hiirikäyttäytymisestä [Scheirer *et al.*, 2002] pääteltävää kontekstittietoa. Silloinkin ongelmana on, miten päätellä, johtuuko käyttäjän tunnistettu emootio vuorovaikutuksesta vai jostakin vuorovaikutuksen ulkopuolisesta tekijästä.

Emootioiden tunnistamisessa fysiologisista signaaleista on etuna se, että fysiologiset reaktiot ovat suurimmaksi osaksi tahdosta riippumattomia. Niiden avulla voidaan siis tavoittaa spontaaneja reaktioita luotettavammin kuin esimerkiksi mallintamalla kasvojen ilmeitä [Surakka and Hietanen, 1998], jotka ovat sosiaalisen kontrollin alaisia [Ekman and Friesen, 1969]. Fysiologisten signaalien avulla voidaan tunnistaa myös käyttäjän emootioihin liittyviä asioita, joista käyttäjä itse ei ole tietoinen tai jotka eivät tulisi esille esimerkiksi subjektiivisia arvioita käyttämällä. Wilson ja Sasse [Wilson and Sasse, 2000] tutkivat videon laadun suhdetta käyttäjältä vaadittuun henkiseen kuormitukseen. Tutkimustulosten perusteella käyttäjät eivät itse huomanneet sellaista laadun heikennystä, jossa fysiologiset signaalit osoittivat merkkejä stressistä.

Lisäksi kuten Wilson ja Eggemeier [Wilson and Eggemeier, 1991] ovat todenneet fysiologisten signaalien käytössä on etuna se, että informaatio on jatkuvasti saatavilla ja että sen hankkiminen ei häiritse ensisijaista tehtävää. Fysiologisia signaaleja monitoroimalla saadaan havaittua informaatiota käyttäjän emootioista häiritsemättä vuorovaikutuksen kulkua ja tehtävän suoritusta. Esimerkiksi kun järjestelmä kysyy käyttäjältä, piditkö tästä toiminnosta, ensisijaisen tehtävän suorittaminen keskeytyy. Informaation jatkuva saatavuus on tärkeää, koska informaatiota halutaan havaita vuorovaikutuksen kaikissa vaiheissa eikä esimerkiksi vain silloin, kun käyttäjältä kysytään subjektiivisia arvioita. Fysiologisten signaalien heikkoutena on niiden herkkyys esimerkiksi puhumisesta ja liikkumisesta

aiheutuville häiriöille. Tämän vuoksi on kehitettävä ohjelmia, jotka havaitsevat ja korjaavat tällaisia artefaktoja reaaliajassa [Wilson and Eggemeier, 1991].

4.3. Syke käyttäjän emootioiden tunnistamisessa

Sydämen aktiviteetti on yksi yleisimmistä psykofysiologisista autonomisen hermoston aktivaation mittareista ja se on ollut tutkimuskäytössä jo kauan [Siddle and Turpin, 1980]. Sydämen aktiviteetin mittareita on useita ja yksi niistä on sydämen syke. Sydämen aktiviteetti on yleisesti liitetty vireystilan muutoksiin, mutta niitä on käytetty myös emootiotutkimuksessa. Sydämen sykkeen on todettu erottelevan erityisesti emootioiden valenssi-ulottuvuutta [Cacioppo *et al.*, 2000]. Aikaisemmat tutkimukset autonomisen hermoston kyvystä erotella positiivisia ja negatiivisia emootioita osoittavat, että sykkeessä tapahtuu merkitsevästi suurempaa aktivaatiota negatiivisten kuin positiivisten perusemootioiden aikana [Cacioppo *et al.*, 2000]. Käyttäjän emootioiden havaitsemisessa ja tunnistamisessa on käytettävä useampaa fysiologista signaalia, koska tunnistaminen on luotettavampaa silloin kun käytetään useampaa eri signaalia verrattuna vain yhden signaalin käyttöön [Wilson and Eggemeier, 1991].

Koska sydämen aktiviteetti liittyy vireystilan muutoksiin, sitä on käytetty monitoroimaan käyttäjän kognitiivista kuormitusta, stressiä ja ahdistusta [Hilbert and Redmiles, 1999, p. 390; Rowe *et al.*, 1998; Wastell 1990]. Joissakin viimeaikaisissa tutkimuksissa on alettu käyttää sydämen aktiviteettia nimenomaan emootioihin liittyen erilaisissa sovelluksissa. Sykettä tai muita sydämen aktiviteetin mittareita, usein yhdessä muiden fysiologisten mittareiden kanssa, on käytetty muun muassa riippuvana muuttujana käytettävyydestänsä [Ward and Marsden, 2003] ja käyttäjän emootioiden tunnistamisessa [esimerkiksi käyttäjän turhautumisen tunnistamisessa, Scheirer *et al.*, 2002].

Sydämen aktiviteettia voidaan mitata muun muassa elektrokardiografialla (EKG), joka perustuu sydänlihaksen sähköisen aktiviteetin mittaamiseen kehoon kiinnitettävien elektrodien avulla. Ballistokardiografia (BKG) perustuu sydämen toiminnasta aiheutuvan kehoon leviävän rekyylin mittaamiseen [Smith, 1974]. Kehittyneemmät ja tarkemmat sydämen aktiviteetin mittaamenetelmät, kuten esimerkiksi elektrokardiografia, ovat syrjäyttäneet ballistokardiografian kliinisessä käytössä. Ihmisen ja tietokoneen vuorovaikutuksessa tarkkuusvaatimukset eivät kuitenkaan ole yhtä suuret kuin kliinisessä käytössä vaan siinä BKG:n tarkkuus on riittävä. Tarkemmat mittaamenetelmät ovat ballistokardiografiaan verrattuna kalliita ja lisäksi ne eivät tunkeilevuutensa takia sovellu ihmisen ja tietokoneen vuorovaikutukseen.

Edellisen pohjalta voin nyt tarkentaa tutkimustehtävääni. BKG-tuoli² on ballistokardiografiseen mittaamenetelmään perustuva sykettä mittaava tuoli. Sen istuinosaan, selkänojaan ja käsinojiin on upotettu sähkömekaanisesta kalvosta valmistettuja antureita, joiden avulla tuolissa istujan syke voidaan laskea. BKG-tuolin etu käyttäjän emootioiden tunnistamisessa on sen häiritsemättömyys. Sykeinformaation havaitseminen tuolilla ei häiritse eikä keskeytä vuorovaikutuksen kulkua. BKG-tuoli on yleensä myös jatkuvassa kontaktissa käyttäjään, toisin kuin esimerkiksi näppäimistöön tai hiireen upotetut anturit. Antureiden jatkuva kontakti käyttäjään mahdollistaa emootioihin liittyvän informaation jatkuvan saatavuuden. Tulen selvittämään kokeellisella tutkimuksella, onnistuuko BKG-tuoli erottelemaan positiivisiin, negatiivisiin ja neutraaleihin ärsykkeisiin liittyviä sykereaktioita. Vertaan myös BKG-tuolin mittaamia sykearvoja perinteisellä sykkeen mittaamenetelmällä rekisteröityihin sykearvoihin.

Kokeessa sykettä mitataan myös korvolehden fotopletysmografialla (PPG), jonka rekisteröimiin sykearvoihin vertaamalla BKG-tuolin mittaamat sykearvot verifioidaan. Fotopletysmografia mittaa sydämen sykettä ääreisverenkierrossa havaittavien sykäysten avulla. PPG-anturi koostuu infrapunavaloa lähettävän diodin ja fototransistorin parista. Diodin lähettämä infrapunavalon läpäisee korvolehden ja fototransistori havaitsee valon. Infrapunavalon läpäisee korvolehden heikommin veripulssin kulkiessa mittauskohdan hiussuonien läpi kuin pulssien välissä [Stern, 1974]. Kontrolloidun kokeen tavoitteena on tutkia ensiksi, mittaako BKG-tuoli sykettä yhtä hyvin kuin fotopletysmografinen sykemittaus korvalehdestä. BKG-tuolin saamia sykearvoja verrataan fotopletysmografialla mitattuihin sykearvoihin ja lisäksi verrataan näiden kahden metodin eroa mittaushäiriöiden määrässä. Toiseksi kokeessa tutkitaan, voidaanko BKG-tuolilla rekisteröityjen syketietojen perusteella erottaa käyttäjän emootioiden valenssia eli onko positiivisiin, negatiivisiin ja neutraaleihin ärsykkeisiin liittyvissä sykereaktioissa eroja.

5. Metodi

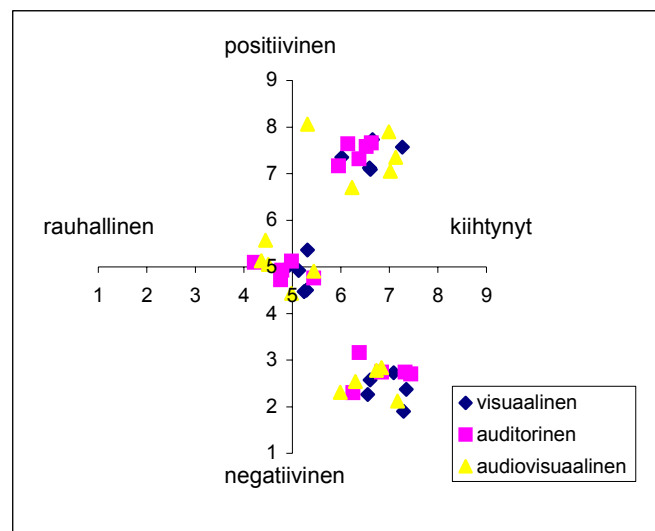
5.1. Koehenkilöt

Kokeeseen tullaan keräämään aineistoa 24 onnistuneen mittauksen verran vapaaehtoisilta koehenkilöiltä.

² BKG-tuoli on Tampereen Teknillisen yliopiston signaalinkäsittelyn laitoksella kehitetty prototyyppi. BKG-tuoli kuuluu Tekesin rahoittamaan "Kalvotekniikkaan perustuvat uudet käyttöliittymäsovellukset" -hankkeeseen.

5.2. Ärsykkeet

Koe koostuu visuaalisista, auditiivisista ja audiovisuaalisista ärsykkeistä. Tällä tavoin saadut tulokset eivät riipu yhdestä tietystä ärsykytyypistä. Kokeessa käytettävät ärsykkeet on valittu IAPS-kuvaärsykkeiden kokoelmasta [Lang *et al.*, 1999] ja IADS-ääniärsykkeiden kokoelmasta [Bradley and Lang, 1999]. Kokeessa esitettävät ärsykkeet jakautuvat IAPS- ja IADS-luokitusten perusteella kolmeen valenssiluokkaan: positiiviset, negatiiviset ja neutraalit. Kokeessa käytettyjen positiivisten ja negatiivisten ärsykkeiden kiihdyttävyytensä on kontrolloitu siten, että kokeeseen valitut positiiviset ja negatiiviset ärsykkeet ovat molemmat melko kiihdyttäviä (Kuva 2). Audiovisuaaliset ärsykkeet ovat samanaikaisesti esitettäviä kuva- ja ääniärsyke pareja. Ärsykeparit ovat sisällöltään samankaltaisia, esimerkiksi kuva nauravasta vauvasta ja nauravan vauvan ääni. Stimulaatiota ohjataan kokeen aikana PC-tietokoneessa toimivalla E-Prime-ohjelmalla. Ärsykkeet esitetään koehenkilöille nestekidenäytöllä, joka on säädetty noin 100 senttimetrin etäisyydelle silmistä. Auditiiviset ärsykkeet esitetään kaiuttimien kautta vakioidulla äänenvoimakkuudella. Kokeessa käytetyt ärsykkeet on lueteltu liitteessä 1.



Kuva 2. Kokeessa käytettyjen ärsykkeiden sijoittuminen valenssi-kiihdyttävyyss koordinaatistoon.

IAPS- ja IADS-ärsykekokoelmat ovat kokeellisessa psykologiassa yleisesti käytettyjä tunnepitoisten ärsykkeiden kokoelmia. Niiden tarkoituksena on mahdollistaa systemaattisesti tutkittujen ja luokiteltujen ärsykkeiden käyttö kontrolloiduissa kokeissa ja edistää tulosten toistettavuutta.

5.3. Sydämen sykkeen mittaaminen

Ärsykkeiden aikaansaamia sykkeen muutoksia rekisteröidään ärsykkeiden esittämisen aikana BKG-tuolilla ja korvalehdestä mitattavalla fotopletysmografialla (PPG).

Fysiologinen signaali sisältää aina muun muassa liikkumisesta johtuvia artefaktoja eli menetelmästä aiheutuvia virheellisiä mittaustuloksia. Tällaiset artefaktat on poistettava signaalista jollakin menetelmällä. Tässä tutkimuksessa ne poistettiin jälkikäteen tutkimalla sykkeen muutoksia. Jos sykkeen muutos ylittää tietyn raja-arvon, kyseessä on artefakta, joka poistetaan. Tämä menetelmä soveltuu myös reaaliaikaiseen artefaktojen poistoon, joka on välttämätöntä todellisessa käyttötilanteessa. Yhden sekunnin aikana saattaa rekisteröityä useampi kuin yksi sykearvo. Jos arvoja on useampia, niistä lasketaan kyseisen sekunnin aikainen keskiarvo.

5.4. Subjektiiiset arviot

Lopuksi koehenkilöt arvioivat kokeessa käytettyjen ärsykkeiden heissä herättämät tunnekokemukset kolmella asteikolla: valenssi, kiihtyneisyys ja lähestymis-välttämistuntemus. Asteikot esitetään näytöllä ja arvioinnit annetaan näppäimistöllä 1-9 numeronäppäimillä. Kaikki asteikot ovat yhdeksänkohtaisia. Ärsykkeen herättämän tunnekokemuksen valenssi arvioidaan asteikolla positiivinen-negatiivinen, kiihtyneisyys asteikolla rauhallinen-kiihtynyt ja ärsykkeen herättämä lähestymis-välttämiskäyttäytyminen asteikolla 'lähestyisin mielelläni' - 'välttäisin ehdottomasti'.

Tunnepitoisten ärsykkeiden oletetaan virittävän koehenkilössä vastaavia emootioita. Oletus verifioidaan keräämällä koehenkilöiden subjektiiviset arviot. Arvioiden avulla voidaan myös tutkia, vastaako koehenkilöpopulaation arviot IAPS- ja IADS-kokoelmissa dokumentoituja (amerikkalaisia) valenssin ja kiihtyneisyyden keskiarvoja. Tässä kokeessa subjektiivisten arvioiden kerääminen on välttämätöntä, koska kokeessa käytetään audiovisuaalisia yhdistelmäpareja. Yhdistelmille ei ole dokumentoituja keskiarvoja, eikä niitä voida päätellä erillisistä IAPS- ja IADS-arvoista. En myöskään tiedä toista tutkimusta, jossa tällaisia yhdistelmiä olisi käytetty.

5.5. Proseduuri

Ensin laboratorio esitellään koehenkilölle. Hänelle kerrotaan, että kokeessa mitataan ihon lämpötilan muutoksia kuvien katselun ja äänien kuuntelun aikana. Sitten koehenkilöä pyydetään istumaan BKG-tuoliin ja korva-anturi asetetaan hänen oikeaan korvalehteensä. Häntä neuvotaan istumaan rennosti ja seuraamaan kuvia ja ääniä. Häntä pyydetään välttämään liikkumista, ja hänelle

selitetään, että se on tärkeää, koska liikkuminen aiheuttaa häiriöitä mittaukseen. Koehenkilölle kerrotaan myös, että hän saa missä tahansa vaiheessa ja mistä tahansa syystä keskeyttää kokeen. Kun kokeenjohtaja poistuu huoneesta, koehenkilö istuu puoli minuuttia paikallaan ennen kuin ärsykkeiden esitys aloitetaan.

Koe koostuu kolmesta neljän minuutin kestoisesta osuudesta: visuaalisesta, auditiivisesta ja audiovisuaalisesta. Osiot esitetään peräkkäin ja niiden välissä on pieni lepotauko, jolloin saa liikkua vapaasti. Jokainen osio koostuu viidestä positiivisesta, viidestä negatiivisesta ja viidestä neutraalista ärsykkeestä. Kokeessa on siis yhteensä 45 ärsykettä. Ärsykkeiden esitysjärjestys vaihtelee satunnaisesti koehenkilöittäin ja koe on tasapainotettu siten, että visuaalisen, audiovisuaalisen ja auditiivisen osuuden järjestys vaihtelee. Ärsykkeiden kesto on kuusi sekuntia ja niiden välinen intervalli kestää 10 sekuntia. Näyttö on intervallien aikana musta.

Kokeen jälkeen koehenkilöä pyydetään antamaan subjektiiviset arviot jokaisesta ärsykkeestä edellä mainituilla kolmella arviointiasteikolla. Koehenkilölle annetaan ohjeet jokaisen asteikon käytöstä ja hänen kanssaan tehdään aluksi kolme (yksi ärsyke jokaisesta kategoriasta) harjoitusarviointia ärsykkeillä, joita ei käytetä kokeessa. Häntä neuvotaan vastaamaan spontaanisti oman subjektiivisen kokemuksensa mukaisesti. Jokainen ärsyke arvioidaan ensin valenssi-, sitten kiihtyneisyys- ja viimeiseksi lähestymisvälttämistuntemus-asteikolla. Ärsyke esitetään ennen jokaista asteikkoa eli yhteensä kolme kertaa. Asteikko ilmestyy ruudulle ärsykkeen esittämisen jälkeen. Kun koehenkilö antaa arvionsa, asteikko katoaa, ja piippaus ilmaisee, että arviointi on rekisteröity. Visuaaliset, auditiiviset ja audiovisuaaliset ärsykkeet arvioidaan omissa osioissaan. Sekä osioiden järjestys että ärsykkeiden esitysjärjestys osion sisällä vaihtelee satunnaisesti. Koehenkilö tekee arvioinnit omaan tahtiin.

Kokeen kokonaiskesto on noin tunti (ärsykkeiden esitys 3*4 minuuttia ja arviointi noin puoli tuntia). Kokeessa käytetään peitetarinaa: koehenkilölle kerrotaan, että korva-anturi mittaa ihon lämpötilan muutoksia eikä tuolin erityislaatua esitellä koehenkilölle. Varsinaisen kokeen ja arviointien jälkeen koehenkilölle kerrotaan, mitä kokeessa todella on mitattu. Hänelle kerrotaan, että peitetarinan käyttö on välttämätöntä, koska tietoisuus sykkeen mittaamisesta voisi vaikuttaa hänen sykkeeseensä.

5.6. Tilastollinen analyysi

Aineiston tilastollisessa analyysissä käytetään tyypillisiä fysiologien aineiston analysointimenetelmiä. Syke on melko hitaasti reagoiva fysiologinen signaali, joten ärsykkeen aiheuttama reaktio lasketaan ärsykkeen ajalta (6 sekuntia) sekä

kymmenen sekuntia ärsyksen päättymisen jälkeen eli yhteensä kuudentoista sekunnin kestoiselta jaksolta. Kuhunkin ärsykeeseen liittyville sykearvoille tehdään perustasokorjaus. Perustasokorjauksessa määritetään ensin kutakin ärsykettä edeltävä sykkeen perustaso laskemalla keskiarvo ärsykettä edeltäviltä sekunneilta. Sen jälkeen kaikista kuudestatoista sykearvosta vähennetään saatu perustaso, jolloin saadaan sykkeen suhteellinen muutos eri ajanhetkillä. Perustasokorjaus on tehtävä fysiologiselle aineistolle aina, koska fysiologisissa signaaleissa on niin suurta variaatiota, ettei absoluuttisia arvoja voida käyttää.

Tutkiakseni, onko positiivisiin, negatiivisiin ja neutraaleihin ärsykkeisiin liittyvissä sykereaktioissa eroja lasken sykkeiden keskiarvot jokaiselle ärsykkeelle kaikkien koehenkilöiden yli perustasokorjatuista sykearvoista. Tutkin tilastollisin menetelmin, onko positiiviseen, negatiiviseen ja neutraaliin kategoriaan sijoittuvien ärsykkeiden välillä tilastollisesti merkitseviä eroja. Tilastollinen analyysi tehdään toistuvien mittausten varianssianalyysillä, jolla tutkitaan onko eri ajanhetkien (1. - 16. sekunti ärsyksen alkamisesta) sykkeissä merkitseviä eroja. Tilastollisia merkitsevyyksiä testataan lisäksi myös t-testeillä.

BKG-tuolin luotettavuutta analysoidaan vertaamalla BKG-tuolin ja PPG:n eroja mittaushäiriöiden määrässä. Artefaktoja sisältävien sykearvojen määrää verrataan sykearvojen kokonaismäärään molemmissa menetelmissä.

6. Pohdintaa

BKG-tuolin käyttöönottoon todellisessa tietokoneen käyttötilanteessa on vielä matkaa eikä laboratoriotuloksia voi suoraan yleistää todelliseen käyttötilanteeseen. Yksi tällainen ongelma on ballistokardiografisen mittauksen herkkyyksistä aiheutuville häiriöille [Siddle and Turpin, 1980, p. 153]. Laboratoriossa mitattaessa koehenkilöitä pyydetään olemaan kokeen aikana mahdollisimman liikkumatta. Tätä ei kuitenkaan voida vaatia käyttäjältä normaalissa käyttötilanteessa. BKG-tuolin käyttämisen mittausmenetelmän ominaisuutena on lisäksi se, että tuoli ei kykene mittaamaan sykettä asennon vaihtamisen aikana. Signaalinkäsittelystä vastaavien algoritmien tulisi kyetä poistamaan artefaktoja reaaliajassa. Tällöin saatetaan tosin menettää kriittistä informaatiota, jos käyttäjän emootioiden kannalta relevantti tapahtuma ajoittuu liikkumisen kanssa samaan aikaan. Jotta käyttöliittymässä voitaisiin hyödyntää fysiologisia signaaleja reaaliajassa, on kehitettävä algoritmeja, jotka analysoivat useaa yhtäaikaista fysiologista signaalia. Parhaaseen tunnistamistulokseen päästäkseen tunnistamisessa voidaan huomioida myös yksittäisen käyttäjän tyypillisiä fysiologisten reaktioiden malleja [Wilson and Eggemeier, 1991, p. 352]. Jatkossa BKG-tuolia olisikin kehitettävä siten, että sitä voidaan käyttää käyttöliittymässä reaaliaikaisesti.

Jos BKG-tuoli osoittautuu validiksi sykkeen mittausmenetelmäksi, sitä voidaan käyttää tietokoneen aistinelimenä käyttäjän emootioiden tunnistamisessa yhdessä muiden fysiologisten signaalien kanssa. Myös muita emootioiden tunnistamiseen tarvittavia signaaleja tulisi voida mitata vuorovaikutusta häiritsemättä esimerkiksi käyttäjän luonnolliseen ympäristöön upotetuilla antureilla. BKG-tuolia voidaan käyttää myös aivan toisenlaisissa käyttötarkoituksissa kuin käyttäjän emootioiden tunnistamisessa. Tästä voisi olla esimerkkinä vanhusten tai muiden sydänsairauksien riskiryhmään kuuluvien ihmisten sykkeen seuranta.

7. Lopuksi

Käyttäjän emootioiden tunnistamisessa fysiologisista signaaleista tarvitaan vielä paljon perustutkimusta. Tutkimusta tarvitaan sekä mittausmenetelmien kehittämisessä että emootioiden tunnistamisessa mitatuista signaaleista. Tämä tutkimus palvelee kumpaakin tarkoitusta. Lisäksi jatkossa on tärkeää pohtia, millaiset emotionaalisen informaation havaitsemismenetelmät soveltuvat mihinkin tarkoitukseen sekä vertailla eri havaitsemismenetelmiä esimerkiksi niiden emootioiden tunnistamistarkkuuden suhteen.

Viiteluettelo

- [Allanson, 2002] Jennifer Allanson, Electrophysiologically interactive computer systems. *IEEE Computers*, **35** (3), 2002, 60-65.
- [Allanson and Wilson, 2002] Jennifer Allanson and Gillian M. Wilson (eds.), Physiological Computing. *Workshop on Physiological Computing, CHI '02 Conference Proceedings*, 2002, 63.
- [Aula and Surakka, 2002] Anne Aula and Veikko Surakka, Auditory emotional feedback facilitates human-computer interaction, *HCI2002, Conference Proceedings*, 2002, 337-349, Springer, London.
- [Bradley and Lang, 1994] Margaret M. Bradley and Peter J. Lang, Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, **25** (1), 1994, 49-59.
- [Bradley and Lang, 1999] Margaret M. Bradley and Peter J. Lang, The international affective digitised sounds (IADS): Stimuli, instruction manual and affective ratings. University of Florida, The Center for Research in Psychophysiology, Technical Report B-2, 1999.
- [Cacioppo et al., 2000] John T. Cacioppo, Gary G. Berntson, Jeff T. Larsen, Kirsten M. Poehlmann and Tiffany A. Ito, The psychophysiology of emotion. In: Michael Lewis and Jeannette M. Haviland (eds.), *Handbook of Emotions*. The Guilford Press, New York, 2000, 2nd Edition, 173-191.

- [Cacioppo and Gardner, 1999] John T. Cacioppo and Wendi L. Gardner, Emotion. *Annual Review of Psychology*, **50**, 1999, 191-214.
- [Christie and Friedman, in press] Israel C. Christie and Bruce H. Friedman, Autonomic specificity of discrete emotion and dimensions of affective space: a multivariate approach. *International Journal of Psychophysiology*, in press.
- [Davidson, 2003] Richard J. Davidson, Seven sins in the study of emotion: Correctives from affective neuroscience. *Brain and Cognition*, **52**, 2003, 129-132.
- [Davidson and Cacioppo, 1992] Richard J. Davidson and John T. Cacioppo, New developments in the scientific study of emotion: An introduction to the Special Section. *Psychological Science*, **3** (1), 1992, 21-22.
- [Dimberg, 1997] U. Dimberg, Facial reactions: rapidly evoked emotional responses. *Journal of Psychophysiology*, **11** (2), 1997, 115-123.
- [Ekman, 1992] Paul Ekman, Are there basic emotions? *Psychological Review*, **99**, 1992, 550-553.
- [Ekman, 1999] Paul Ekman, Basic emotions. In: T. Dalgleish and M. Power (eds.), *Handbook of Cognition and Emotion*, John Wiley & Sons Ltd., Sussex, U.K, 1999, 45-60.
- [Ekman and Friesen, 1969] P. Ekman and W. V. Friesen, The repertoire of nonverbal behavior - categories, origins, usage and coding. *Semiotica*, **1**, 1969, 49-98.
- [Ekman *et al.*, 1983] P. Ekman, R. W. Levenson and W. V. Friesen, Autonomic nervous system activity distinguishes among emotions. *Science*, **221**, 1983, 1208-1209.
- [Hilbert and Redmiles, 1999] David M. Hilbert and David F. Redmiles, Extracting usability information from user interface events. *ACM Computing Surveys*, **32** (4), 1999, 384-421.
- [Hollnagel, 2003] Erik Hollnagel, Is affective computing an oxymoron? *International Journal of Human-Computer Studies*, **59**, 2003, 65-70.
- [Hudlicka, 2003] Eva Hudlicka, To feel or not to feel: The role of affect in human-computer interaction. *International Journal of Human-Computer Studies*, **59**, 2003, 1-32.
- [Izard and Ackerman, 2000] Izard and Ackerman, Motivational, organizational, and regulatory functions of discrete emotions. In: Michael Lewis and Jeannette M. Haviland (eds.), *Handbook of Emotions*. The Guilford Press, New York, 2nd Edition, 2000, 253-264.
- [Johnstone and Scherer, 2000] Tom Johnstone and Klaus R. Scherer, Vocal communication of emotion. In: Michael Lewis and Jeannette M. Haviland

- (eds.), *Handbook of Emotions*. The Guilford Press, New York, 2000, 2nd Edition, 220-235.
- [Kapoor *et al.*, 2001] Ashish Kapoor, Selene Mota and Rosalind W. Picard, Towards a learning companion that recognizes affect. *Emotional and Intelligent II: The Tangled Knot of Social Cognition, AAAI Fall Symposium 2001 Conference Proceedings, 2001*.
- [Klein *et al.*, 2002] J. Klein, Y. Moon and R.W. Picard, This computer responds to user frustration: Theory, design and results. *Interacting with computers*, **14**, 2002, 119-140.
- [Kübler *et al.*, 2001] Andrea Kübler, Boris Kotchoubey, Jochen Kaiser, Jonathan R. Wolpaw and Niels Birbaumer, Brain-computer communication: Unlocking the locked-in. *Psychological Bulletin*, **127**, 2001, 358-375.
- [Kubrick, 1968] Stanley Kubrick (director), 2001: a space odyssey, Warner Bros., 1968. See <http://kubrickfilms.warnerbros.com/>. Suomeksi 2001 Avaruusseikkailu.
- [Lang *et al.*, 1998] P. J. Lang, M. M. Bradley and B. N. Cuthbert, Emotion, motivation, and anxiety: Brain mechanisms and psychophysiology. *Biological Psychiatry*, **44**, 1998, 1248-1263.
- [Lang *et al.*, 1999] P. J. Lang, M. M. Bradley and B. N. Cuthbert, International Affective Picture System (IAPS): Instruction manual and affective ratings. University of Florida, The Center for Research in Psychophysiology, Technical Report A-4, 1999.
- [Levenson, 1992] Robert Levenson, Autonomic nervous system differences among emotions. *Psychological science*, **3**, 1992, 23-27.
- [Lusted and Knapp, 1996] Hugh S. Lusted and R. Benjamin Knapp, Controlling computers with neural signals. *Scientific American*, **275** (4), 1996, 82-87.
- [Mayne and Ramsey, 2001] T. J. Mayne and J. Ramsey, The structure of emotions: a nonlinear dynamic systems approach. In: Tracy J. Mayne and George A. Bonanno (eds.), *Emotion – Current issues and future directions*. The Guilford Press, New York, 2001, 1-37.
- [Nesse, 1990] Randolph M. Nesse, Evolutionary explanations of emotions. *Human Nature*, **1** (3), 1990, 261-289.
- [Oatley and Jenkins, 1996] Keith Oatley and Jennifer M. Jenkins, *Understanding Emotions*, Blackwell, Cambridge, Massachusetts, 1996.
- [Ortony and Turner, 1990] Andrew Ortony and Terence J. Turner, What's basic about basic emotions? *Psychological Review*, **97**, 315-331.
- [Partala *et al.*, 2001] Timo Partala, Anne Aula and Veikko Surakka, Combined voluntary gaze direction and facial muscle activity as a new pointing

- technique. *Human-Computer Interaction - Interact '01, Conference Proceedings*, 2001, IOS Press.
- [Pentland, 2000] Alex Pentland, Perceptual intelligence. *Communications of the ACM*, **43** (3), 2000, 35-44.
- [Picard, 1995] Rosalind Picard, Affective Computing. MIT Media Laboratory, Perceptual Computing Section, Technical Report 321, 1995.
- [Picard, 1997] Rosalind Picard, *Affective Computing*, MIT Press, Cambridge, 1997.
- [Picard *et al.*, 2001] Rosalind W. Picard, Elias Vyzas and Jennifer Healey, Towards machine emotional intelligence: Analysis of affective physiological state. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23** (10), 2001, 1175-1190.
- [Reynolds and Picard, 2001] Carson Reynolds and Rosalind W. Picard, Designing for affective interactions. *Proceedings from the 9th International Conference on Human-Computer Interaction, 2001*.
- [Rowe *et al.*, 1998] Dennis W. Rowe, John Sibert and Don Irwin, Heart rate variability: Indicator of user state as an aid to human-computer interaction. *Human Factors in Computing Systems, CHI '98 Conference Proceedings*, 1, 1998, 480-487, CAN Press.
- [Scheirer *et al.*, 2002] Jocelyn Scheirer, Raul Fernandez, Jonathan Klein and Rosalind W. Picard, Frustrating the user on purpose: A step toward building an affective computer. *Interacting with computers*, **14**, 2002, 93-118.
- [Schulkin *et al.*, 2003] Jay Schulkin, Barbara L. Thompson and Jeffrey B. Rosen, Demythologizing the emotions: Adaptation, cognition, and visceral representations of emotion in the nervous system. *Brain and Cognition*, **52**, 2003, 15-23.
- [Siddle and Turpin, 1980] David A. Siddle and Graham Turpin (eds.), Measurement, quantification, and analysis of cardiac activity. In: I. Martin and P. H. Venables (eds.), *Techniques in psychophysiology*, John Wiley & Sons Ltd, 1980, 139-246.
- [Smith, 1974] N. Ty Smith, Ballistocardiography. In: A. M. Weissler (ed.), *Noninvasive Cardiology*, Grune & Stratton, New York, 1974, 39-148.
- [Spielberg, 2001] Steven Spielberg (director), A.I. Artificial Intelligence, Warner Bros., 2001. See <http://aimovie.warnerbros.com/>. Suomeksi A.I. Tekoöly.
- [Stern, 1974] Robert M. Stern, Ear lobe photoplethysmography. *Psychophysiology*, **11** (1), 1974, 73-75.
- [Surakka and Hietanen, 1998] Veikko Surakka and Jari K. Hietanen, Facial and emotional reactions to Duchenne and non-Duchenne smiles. *International Journal of Psychophysiology*, **29**, 1998, 23-33.

- [Tomkins, 1982] S. S. Tomkins, Affect theory. In: P. Ekman (ed.), *Emotion in the Human Face*, Cambridge University Press, Cambridge, 1982, 353-395.
- [Turk and Robertson, 2000] Matthew Turk and George Robertson, Perceptual user interfaces. *Communications of the ACM*, **43** (3), 2000, 33-34.
- [Ward and Marsden, 2003] R. D. Ward and P. H. Marsden, Physiological responses to different WEB page designs. *International Journal of Human-Computer Studies*, **59**, 2003, 199-212.
- [Wastell, 1990] D. Wastell, Mental effort and task performance: towards a psychophysiology of human-computer interaction foundations: cognitive ergonomics. *IFIP INTERACT'90: Human-Computer Interaction, Conference Proceedings*, 1990, 107-112.
- [Wiederhold *et al.*, 2001] Brenda K. Wiederhold, Dong P. Jang, Mayumi Kaneda, Irene Cabral, Yair Lurie, Todd May, In Y. Kim, Mark D. Wiederhold and Sun I. Kim, An investigation into physiological responses in virtual environments: An objective measurement of presence. In: Giuseppe Riva and Carlo Galimberti (eds.), *Towards CyberPsychology: Mind, Cognitions and Society in the Internet Age*, IOS Press, Amsterdam, 2001, 175-184.
- [Wilson and Eggemeier, 1991] Glenn F. Wilson and F. Thomas Eggemeier, Physiological assessment of workload in multi-task environments. In: D. Damos (ed.), *Multiple-task performance*, Taylor and Francis, London, 1991, 329-360.
- [Wilson and Sasse, 2000] Gillian M. Wilson and M. Angela Sasse, Do users always know what's good for them? Utilizing physiological responses to assess media quality. *HCI 2000, Conference Proceedings*, 2000, 327-339.
- [Zajonc and McIntosh, 1992] R. B. Zajonc and D. N. McIntosh, Emotions research: Some promising questions and some questionable promises. *Psychological Science*, **3**, 1992, 70-74.

Liite 1. Kokeessa käytettävät ärsykkeet

VISUAALINEN		IAPS nro	IAPS Nimi	Valenssi	Kiihdyttävyys
Valenssi-luokka	Positiivinen	8080	Sailing	7,73	6,65
		8185	Skydivers	7,57	7,27
		5470	Astronaut	7,35	6,02
		8180	CliffDivers	7,12	6,59
		8400	Rafters	7,09	6,61
	Keskiarvo			7,372	6,628
	Negatiivinen	6230	AimedGun	2,37	7,35
		3150	Mutilation	2,26	6,55
		6550	Attack	2,73	7,09
		6350	Attack	1,9	7,29
		9250	WarVictim	2,57	6,6
	Keskiarvo			2,366	6,976
	Neutraali	2220	MaleFace	5,03	4,93
		3550,2	Coach	4,92	5,13
		7560	Freeway	4,47	5,24
8060		Boxer	5,36	5,31	
1390		Bees	4,5	5,29	
Keskiarvo			3,498	6,160	

AUDITIIVINEN		IADS nro	IADS Nimi	Valenssi	Kiihdyttävyys
Valenssi-luokka	Positiivinen	220	BoyLaugh	7,64	6,14
		226	Laughing	7,32	6,37
		353	Baseball	7,66	6,63
		601	ColnialMusic	7,17	5,95
		820	FunkMusic	7,58	6,52
	Keskiarvo			7,474	6,322
	Negatiivinen	626	Bombs	2,7	7,44
		116	Wasp	2,74	6,84
		627	HowlingRain	2,74	7,32
		501	PlaneCrash	3,16	6,38
		502	EngineFails	2,3	6,25
	Keskiarvo			2,728	6,846
	Neutraali	101	Cat	4,72	4,76
		105	PuppyCry	4,76	5,44
		325	Traffic	5,12	4,98
425		Train	4,93	4,78	
704		Touchtone	5,1	4,22	
Keskiarvo			3,727	5,932	

AUDIOVISUAALINEN		IAPS nro	IAPS Nimi	Valenssi	Kiihdyttävyys	IADS nro	IADS Nimi	Valenssi	Kiihdyttävyys
Valenssi-luokka	Positiivinen	2050	Baby	8,2	4,57	110	BabyLaugh	7,92	6,04
		8370	Rafting	7,77	6,73	815	RockNRoll	8,02	7,25
		8490	RollerCoaster	7,2	6,68	360	RollerCoaster	6,9	7,36
		8030	Skier	7,33	7,35	352	SportsCrowd	7,36	6,91
		5450	Liftoff	7,01	5,84	415	Countdown	6,38	6,62
	Keskiarvo			7,502	6,234			7,316	6,836
	Negatiivinen	3500	Attack	2,21	6,99	286	Victim	2,03	7,35
		2800	SadChild	1,78	5,49	261	BabyCry	2,84	6,49
		6312	Abduction	2,48	6,37	290	Fight	2,59	6,22
		3250	OpenChest	3,78	6,29	287	CardiacArr	1,9	7,38
		6510	Attack	2,46	6,96	730	GlassBreak	3,1	6,49
	Keskiarvo			2,542	6,42			2,492	6,786
	Neutraali	7550	Office	5,27	3,95	320	Office1	4,83	5,04
		7211	Clock	4,81	4,2	311	Crowd2	5,44	4,52
		9411	Boy	4,63	5,37	410	Helicopter2	5,19	5,5
1313		Frog	5,65	4,79	171	CountryNight	5,49	4,51	
6930		Missiles	4,39	4,88	699	Bomb	4,46	5,07	
Keskiarvo			4,95	4,558			5,082	4,928	

Tuntopalautteen hyödyntäminen näkövammaisten esikoululaisten opetuksessa

Taina Arjanmaa

Tiivistelmä

Tässä tutkimuksessa pyrin kartoittamaan tutkimustuloksia ja teorioita tuntopalautteesta ja sen hyödyntämisestä esikouluikäisille lapsille suunnatuissa tietokonesovelluksissa. Tarkoitukseni on koota yhteen tietoa, joka voisi hyödyttää sovellusten suunnittelijoita. Pyrin lähestymään aihetta myös vähemmän tekniseltä kannalta määrittäen myös näkövammaisuuden käsitettä sekä kartoittamalla esikouluikäisten lasten kognitiivista tasoa. Lisäksi pohdin tuntopalautteen hyviä ja huonoja puolia näkövammaisille lapsille suunnatuissa sovelluksissa.

Tuntopalautetta välittävissä laitteissa on vielä mielestäni paljon kehittämistä. Nykyisin on käytössä lähinnä yhdestä pisteestä tietoa välittäviä laitteita, jotka eivät tue reaaliaikaisen tunnistelutapoja. Tavallisestihan sokeita kannustetaan käyttämään molempia käsiä ja useita sormia yhtäaikaisesti. Tuntolaitteiden vajavuus selittää mielestäni osan tutkimustuloksista, joiden mukaan tuntopalautteen tulkinta on hitaampaa kuin visuaalisen palautteen tulkinta.

Kartoituksessani tuli myös selkeästi esille kuinka tuntopalautteella voidaan hyvin korvata ja tehostaa visuaalista palautetta. Tuntopalautteesta on hyötyä sekä näkeville että näkövammaisille. Tuntopalaute sopivasti yhdistettynä auditiiviseen viestintään asettaa sokeat samalle viivalle näkevien kanssa käytettäessä graafisia käyttöliittymiä. Lapsille suunnatut sovellukset antavat näkövammaiselle lapselle väsymättömän harjoitusvastustajan peleissä, joiden avulla hän harjaannuttaa tuntoaistiaan ja oppii käyttämään tietokoneita.

Avainsanat ja -sanonnat: Tuntopalaute, näkövammaisuus, esikouluikäiset
CR-luokat: H.1.2, H.5.2, K.3.1

1. Johdanto

Suomessa syntyy vuosittain noin 100 näkövammaista lasta [Näkövammaisten lasten tuki ry., 2003]. Useimmat heistä käyvät täysin tavallista tarhaa ja koulua. Näkövammaiset tarvitsevat apuvälineitä toimiessaan arjessa näkevien kanssa. Apuvälineistä ja oppimista auttavista sovelluksista on heille hyötyä heidän hahmottaessaan esim. aikaa, paikkaa, tilaa ja syy - seuraus-suhteita.

Kartoitukseni aihepiiri liittyy psykologiaan, kasvatustieteeseen sekä tietojenkäsittelytieteeseen. Psykologian tutkimus on enemmän keskittynyt audio-visuaalisiin kanaviin ja sivuuttaa usein tuntoon perustuvat havainnot. Kasvatustieteeseen aihe liittyy, koska luotavien sovellusten tulee tukea näkövammaisten oivaltavaa oppimista. Tietojenkäsittelytieteen tekniikoilla voidaan luoda ohjelmat olemassa olevaa tietoa käyttäen.

Tutkimuksen tarkoituksena on tuottaa jo olemassa oleva tieto näkövammaisten lasten tuntoaistista yhdeksi kokonaisuudeksi ja soveltaa sitä mahdollisuuksien mukaan oivaltavan oppimisen näkökulmasta. Tutkimuksen tulisi hahmottaa miten näkövammaisen lapsi kokee maailman. Lisäksi pyrin löytämään tapoja parhaiten hyödyntää näkövammaisen lapsen tuntoaistia opetuksessa ja leikissä tapahtumien sekä asioiden havainnollistajana.

Tutkimuskysymyksiä ovat mm. "Eroaako näkövammaisen tuntoaisti näkevien tuntoaistista?", "Miten tuntoaisti voi tukea oppimista?", "Miten näkövammaisen lapsen oppimista voidaan tehdä mielekkäämmäksi ja kiehtovammaksi tuntoaistia hyödyntäen?" "Millainen tulisi olla tuntopalautetta antavan laitteen, jotta sen palaute tulkittaisiin oikein?"

Tutkimuksen tuottamaa tietoa voidaan käyttää koottuna tietopakettina omalta kapealta sovellusalueeltaan. Tällöin esimerkiksi varhaiskasvatuksessa näkövammaisia lapsia kohtaava voi paremmin ymmärtää näkövammaisen lapsen maailmaa ja mahdollisuuksia.

Luvussa kaksi tehdään katsaus näkövammaisuuden määrittelyyn. Lisäksi tarkastellaan miten näkövammaisuus vaikuttaa esimerkiksi oppimiseen verrattuna näkeviin. Kyseessä on pikakatsaus näkövammaisuuden käsitteeseen ja käsitteen huomattavaan laajuuteen.

Kolmas luku käsittelee näkövammaisilla jo käytössä olevia tuntopalautelaitteita ja tutkimusten perusteella luotuja sovellusten mallintamistekniikoita. Pyrin huomioimaan käytössä olevien laitteiden hyvät sekä huonot puolet. Mallintamistekniikoissa on uutena trendinä huomattu, kuinka tärkeää on luoda myös vaikutusmahdollisuuksia sovelluksen sisällön viestimisen lisäksi.

Esikouluikäisen lapsen kognitiivista kehitystä käsitellään neljännessä luvussa. Esitys on Piaget'n [1977] kehitysteorian mukainen ja siten hieman suppea. Tälle kappaleelle oman lisämausteensa antaa keskusteluni Arto Hippulan [2003] kanssa.

Viides luku perehtyy tuntopalautteen jaottelemisen vaikeuteen. Olemme jatkuvasti tuntopalautteen ympäröimänä, sillä saamme jatkuvasti tietoa kehomme asennosta ja eri kehonosiimme kohdistuvasta paineesta. Tuntoaisti

on vähintäänkin yhtä laaja käsite kuin näköaisti. Yhtä vaikeaa on jakaa tätä aistia erillisiin osiin.

Kuudes ja seitsemäs luku käsittelevät tuntopalautteen käyttämisen etuja ja rajoitteita. Nykyiset laitteet luovat ehkä suurimman rajoitteen tarjoten mahdollisuuden useimmiten vain yhdestä pisteestä saatavaan informaatioon. Reaalimaailmassa sokeita rohkaistaan ja opastetaan käyttämään mahdollisuuksien mukaan molempia käsiään ja useita sormia yhtäaikaaisesti. Tuntopalaute voi silti nostaa suorituksen tasoa näkeville ja tehdä monesta tehtävästä mahdollisen näkövammaiselle käyttäjälle.

2. Näkövammaisuus

Näkövammaisuutta on hankala määrittää tarkoin arvoin, koska moni näkemisen osatekijä vaikuttaa näön heikkenemiseen eritavoin [Poussu-Olli ja Keto, 1999]. Näkökyky voidaan esittää janana, jonka toisessa päässä on täydellinen näkö ja toisessa päässä täydellinen sokeus. Näkövammarekisterin vuosikirjan [2000] mukaan näkövammaiset ja ei-näkövammaiset erottavan viivan vetäminen tälle janalle on sopimuksen varaista. Kuitenkin näkövammaiseksi voidaan määritellä henkilö, jolle näkökyvyn heikentyminen aiheuttaa huomattavaa haittaa jokapäiväisissä toiminnoissa [Ojamo, 2000].

Näköaistin puuttuminen vaikeuttaa ja hidastaa sellaisten asioiden oppimista, jotka normaalit lapset oppivat ympäristön visuaalisten virikkeiden avulla kuin itsestään [Poussu-Olli ja Keto, 1999]. Esimerkkinä voisin käyttää tilannetta, jossa sokea lapsi ja näkevä saavat katsella ja koskea täytettyä huuhekajaa biologian tunnilla. Näkeville lapselle on jo täysin selvää miten linnut lentävät (liikuttamalla siipiään), mutta syntymästään asti sokea lapsi ei ole koskaan nähnyt linnun lentävän ja hänelle tämä tulee selittää tarkemmin. Näkevät lapset oppivat paljon tällaisia 'itsestäänselvyksiä' vain tarkkailemalla ympäristöään jokapäiväisissä toiminnoissaan.

Suurin osa näkövammaiseksi luokitelluista ihmisistä ei ole täysin sokeita, vaan he pystyvät aistimaan esimerkiksi valon, varjon ja liikkuvia esineitä [Scholl, 1986, s.23-33]. Tämän varioivuuden vuoksi on tärkeää Schollin [1986, s.23-33] mukaan arvioida jokaisen oppijan tarvitsemat aputoimet erikseen. Yksilöllinen näkövammaisen oppijan arviointi nousi useammassakin lähteessäni esille [Baloian *et al.*, 2002; Christian, 2000; Poussu-Olli ja Keto, 1999; Scholl, 1986]. Mielestäni on tärkeää, että näkövammaisen saa tarvittavat apulaitteet ja tarvittavan avustuksen ilman holhotuksi tulemisen tunnetta. Liialla huolehtimisella voidaan sammuttaa lapsen tai nuoren luontainen halu tutkia maailmaa itsenäisesti [Vilkko-Riihelä, 1999].

Poussu-Olli ja Keto [1999] kumoavat yleisen uskomuksen, jonka mukaan näkövammaisen muut aistit olisivat lähtökohtaisesti tarkempia ja paremmin toimivia kuin näkevillä, vedoten useisiin tutkimustuloksiin. Uskomuksen pohjalla on mahdollisesti muiden aistien suurempi käyttö korvaavina informaatiokanavina, jolloin aistien teho harjoituksen myötä herkistyy. Nimenomaan harjoittelun myötä ihmiskeho harjaantuu laajemmin käyttämään muita kuin näköaistia informaation saamiseen. Ei ole siis kyse jostakin salaperäisestä tuntemisen ja kuulemisen lahjasta, joka sokeilla on syntyessään. Korvaavien aistien harjaannuttamisen tulisi alkaa jo varhain, jolloin näkövammaisen oppiminen varmimmin seuraisi näkevien vauhtia. Kuten täysin vammattomilla lapsilla, myös sokeilla, vanhempien, opettajien sekä sosiaalisen verkoston rohkaisu aktiiviseen ympäristön kartoittamiseen on suotavaa ja kehitystä edistävää [Hippula, 2003; Poussu-Olli ja Keto, 1999].

Näkövammaisten oppivelvollisuus alkaa Suomessa jo 6-vuotiaana sisältäen kaksi vuotta esikoulua ja 9 vuotta peruskoulua [Poussu-Olli ja Keto, 1999]. Näkövammaisten keskusliiton kokoaman oppimateriaalin mukaan [Kurvinen-Tikkanen, 1999] esiopetuksessa ”lapsi saa valmiuksia mm. pistekirjoituksen oppimiseen, tutustuu opiskelua helpottaviin apuvälineisiin sekä harjoittelee itsenäistä liikkumista”.

3. Apuvälineet ja mallinnustekniikat

Sokeille on olemassa monia apuvälineitä, jotka toimivat huomioimatta mitään erityistä esim. www-sivujen tai käyttöliittymien asiasisällöstä. Apuvälineet siis vain toistavat käyttöliittymässä olevan tekstuaalisen tiedon. Näiden tueksi on kehitelty asiasisällön mallinnustekniikoita, jotka pyrkivät antamaan sokealle käyttäjälle informaatiota pelkän tekstin lisäksi myös graafisen käyttöliittymän visuaalisesta sisällöstä. On tärkeää näiden tekniikoiden sisällä ottaa huomioon myös sokealle käyttäjälle tarjotut mahdollisuudet vaikuttaa käyttöliittymään. Käyttöliittymän suunnittelussa vaikuttavat siis sekä mahdollisuus antaa syötettä ja saada informaatiota [Mynatt and Weber, 1994].

3.1. Apuvälineet

Sokeainkirjoitus eli pistekirjoitus (braille-kirjoitus) on Louis Brailleen kehittelemä järjestelmä. Järjestelmässä yhden kirjaimen tai merkin muodostaa 6 pisteen ryhmä, jossa poisjätetyt pisteet merkitsevät kirjaimia tai merkkejä [Poussu-Olli ja Keto, 1999]. Tietokoneen apuvälineinä on jo kauan ollut käytössä braille-näyttöjä, joissa tietokoneen näytöllä oleva teksti-info on käännetty sokeainkirjoitukselle. Tämä oli riittävä tekniikka aikana, jolloin suurin osa tietokoneista käytti tekstipohjaista käyttöliittymää [Boyd *et al.*, 1990].

Esikouluikäisille lapsille ei kirjoitetusta tekstistä ole suurempaa hyötyä, koska suurin osa ei osaa vielä lukea. Poussu-Olli ja Keto [1999] korostavat kuitenkin, että sormenpäiden harjaannuttamisen aloittaminen ajoissa on tärkeää. Lukutaidon puuttumisesta aiheutuvat ongelmat ovat samanlaisia niin näkövammaisten kuin näkevien lasten kanssa.

PHANToM-laite [Brehm and Salisbury, 1998] antaa tuntopalautetta yhden pisteen kautta. PHANToM-laitetta on mahdollista käyttää myös näköärsykkeiden kanssa rinnan, mutta tämän esityksen rajoissa keskityn lapsiin, jotka käyttävät PHANToM-laitetta ilman visuaalisia ärsykeitä eli täysin tuntopalautteen ja äänien varassa. PHANToM pystyy antamaan tektuuriaistimuksen ja voimapalautteen lisäksi myös lämpöaistimuksia [Brehm and Salisbury, 1998]. PHANToM tarjoaa mahdollisuuden käyttää kahta erilaista tuntotyökalua. Janssonin ja Billbergerin [1998] tutkimuksessa nämä on nimetty "stylus" ja "thimble". Stylusta eli vapaasti suomentaen piirrintä, käytettäessä informaatiota saadaan sekä sormenpäistä, kämmenestä että käden asennosta, kun taas sormustimen (*thimble*) käytössä informaatio tulee vain yhden sormenpään tuntoaistimusten kautta. Tutkimuksen [Jansson and Billberg, 1998] mukaan tunnusteltavien kappaleiden tunnistaminen ja määrittely ei ole riippuvainen käytetystä tuntotyökalusta.

Colwell *et al.* [1998] tutkivat Impulse Engine 3000TM -laitteen antamaa tuntopalautetta. Kyseessä on laite, jota PHANToM:in tavoin käytetään paksun kynän oloisella ohjaimella (probe). Probe voi liikkua kolmessa suunnassa eli: ylös-alas, sivuille sekä eteen ja taakse. Laite luo vastusta tunnusteltavan kappaleen mukaan soveltaen tietoa, jonka se saa monitoroimalla käyttäjän käden asentoa. Kuten muissakin voimapalautetta antavissa laitteissa, tuotettavan voiman määrä on rajallinen. Tällöin on mahdollista luonnottomasti läpäistä jokin luonnollisesti vastaavalla voimalla läpäisemätön pinta virtuaalisesti.

Tunto- ja auditiivista palautetta antavilla näytöillä voidaan käyttää graafisia käyttöliittymiä, jotka perustuvat melkein kokonaan ikoneihin ja niiden valitsemiseen hiirellä ohjelmien käynnistämiseksi ja kansioden avaamiseksi. Sjöström [2001] on tutkinut tällaisten näyttöjen sovelluksia lapsille esimerkiksi muistipelin muodossa.

Tuntopalautetta käyttävät hiiret, joystikit ja muut ohjaimet toimivat useimmiten vastustamalla liikettä ikonien ja muiden kohteiden läheisyydessä sekä värisemällä eri tavoin eri ikonien kohdalla [Hinckley and Sinclair, 1999]. Hinckley ja Sinclair [1999] tutkivat ohjaimia, jotka aistivat käyttäjän kosketuksen. Näin saadaan aikaan vuorovaikutusta myös toiseen suuntaan

kuin tavallisesti, eli käyttäjältä laitteelle. Käyttäjän kosketuksesta voidaan aktivoida tai disاپلoida valikkoja ja toimintoja.

Erilaisia tekniikoita voidaan yhdistellä keskenään sopivin tavoin, jotta saataisiin kompensoitua puuttuvaa aistia mahdollisimman hyvin [Baloian *et al.*, 2002]. Ääntä ei tule unohtaa yhtenä mahdollisena viestintätapana sokeille. Erilaiset äänet voivat kertoa mm. tilassa sijaitsevista kohteista kuten *AudioDoom* sovelluksessa, jota Baloian *et al.* [2002] tutkivat.

Mielestäni yksinkertaisinta on käyttää lasten sovelluksissa puhetta. Esimerkkinä puhetta hyödyntävästä käyttöliittymästä käytän ylen lastenohjelma Pikku Kakkosen sivuja, joissa kuvilla ilmaistut linkit puhutaan ääneen kun kursori on niiden päällä <http://www.yle.fi/pikku2/>. Tällainen käyttöliittymä ei tietäkään sellaisenaan sopisi täysin sokealle lapselle, mutta esimerkkinä lukutaidottomuuden aiheuttaman ongelman poistamisesta sivut ovat hyvä esimerkki. Huonona puolena puheessa on sen paljon hitaampi kyky välittää informaatiota kuin luettulla (sekä tavallinen, että sokeainkirjoitus) tekstillä.

3.2. Mallinnustekniikat

Usein on tarkoituksenmukaista, että näkövammaisten käyttämät sovellukset ovat sellaisia, joita myös näkevät voivat käyttää. Tällöin pyritään muodostamaan sovellus, josta muodostuva mentaalimalli on samanlainen näkeville ja sokealle käyttäjälle [Baloian *et al.*, 2002].

Mynatt ja Weber [1994] nostavat esille mielestäni tärkeän seikan mallinnuksen kahtiajakoisuudesta. Tulee ottaa huomioon mahdollisuudet sekä vaikuttaa tapahtumien etenemiseen sovelluksessa että saada tarvittava informaatio sovelluksesta tapahtumien ymmärtämiseksi. Kun tavallinen esikouluikäinen lapsi pelaa esimerkiksi lukemaanoppimista tukevaa peliä hän saa informaation katsomalla näytölle ja vaikuttaa tapahtumiin hiiren tai näppäimistön avulla. Sokeilla käyttäjillä voidaan usein yhdistää nämä kaksi vuorovaikutustapaa. Tällaisista kokeiluista kertovat esim. PHANToM-laitteen kehitys [Brehm and Salisbury, 1998]; Ramsteinin [1996] tutkimus tuntopalautteen ja braille-näytön yhdistämisestä; GUIB projekti [Petrie *et al.*, 1995], jossa pyrittiin yhdistämään synteettistä- ja digitoitua puhetta, merkityksettömät äänet, braille-näyttö sekä kosketustabletti (touchpad) palautteenanto ja -saantivälineinä.

Martin Kurze [1996; 1997; 1998] on useammassa tutkimuksessa tutkinut sokeiden tapaa hahmottaa ympäristöään kuvina. Hän esittää idean, jonka mukaan kappaleiden esitystavan tulisi lähteä liikkeelle sokeiden omasta tavasta saattaa kolmiulotteisen maailman kohteet kaksikulotteiseen muotoon [Kurze, 1996]. Kurze [1997] on luonut tutkimustensa pohjalta mallinnustekniikan, jonka

perusteella pystytään luomaan tarvittavan esitysvoiman omaavia kuvia sokeille (rendering drawings). Itse tekniikka muodostuu seitsemästä vaiheesta, joiden kautta saadaan muodostettua kuvaus kolmiulotteisesta kohteesta tai tilanteesta kaksiulotteiseksi kuvaksi, jota voidaan tunnustella.

Baloian *et al.* [2002] esittävät yleisen mallin (common model), jonka mukaan mikä tahansa näytön kuva tai oikean maailman tilanne voidaan muuttaa sokeille ymmärrettävään muotoon. Aluksi tulee kartoittaa näkövammaisen kognitiiviset taidot ja niiden mahdolliset puutteet. Kohteet, joista kuva muodostetaan, määritellään geometrisesti ja annetaan ominaisuuksille mahdolliset akustiset ja tuntopalaute arvot. Seuraavaksi luodaan tarvittava tietokone-esitys käyttäen juuri käyttäjälle sopivia tilääniä (spatial sound). Käyttäjä kokeilee ja tutkii muodostettua esitystä. Lopuksi tutkitaan mahdolliset vastaamattomuudet luodun mallin ja sokealle muodostuneen sisäisen mallin välillä ja korjataan sovelluksen luomaa esitystä vastaamaan käyttäjän näkemystä. Kun malli vastaa käyttäjän mentaalimallia sitä voidaan soveltaa aina uudelleen uusien kuvien ja kohteiden esittämiseen.

4. Lapsen kognitiivinen kehitys ja sen taso esikouluikäisillä

Tätä aihetta varten kävin lävitse muutamia yleisesti psykologian piirissä hyväksytyjä kehitysteorioita. Valitsin tässä käsiteltäväksi Piaget'n kehitysteorian, koska se selkeimmin mielestäni käsittelee lapsen kognitiivisen kehityksen etenemistä. Piaget'n teoria perustuu hyvin pienestä populaatiosta tehtyihin havaintoihin. Piaget tutki ja havainnoi omia lapsiaan näiden kasvaessa. Teoria tai osia siitä on jälkepäin useasti asetettu tutkimuksen kohteeksi, jonka perusteella Piaget'n havainnot on todettu paikkansa pitäviksi.

Piaget'n kehitysteorian [Piaget, 1977] mukaan esikouluikäinen lapsi elää esioperationaalisen vaiheen loppua eli intuitiivisen ajattelun kautta ja konkreettisten operaatioiden vaiheen alkua. Esioperationaalisten vaiheen aikana lapsen ajattelu on hyvinkin konkreettista ja liittyy suoraan johonkin koettuun asiaan. Esimerkiksi käsite mummu voi koskea vain omaa mummua, eikä lapsi täysin ymmärrä miten hänestäkin voisi joskus tulla mummu. Lisäksi lapsi elää nykyhetkessä hyvin vahvasti. Tyypillistä lapselle on tässä kehitysvaiheessa myöskin uskoa se minkä näkee. Piaget [Vilkko-Riihelä, 1999] teki vesilasikokeen osoittaakseen tämän seikan. Kokeessa kaadettiin kahdesta samankokoisesta lasista, joissa oli lapsienkin mielestä saman verran nestettä, toisen sisältö kapeampaan lasiin. Koska veden pinta oli kapeammassa lasissa korkeammalla kuin toisessa lasissa, päättelivät lapset suoraan, että kapeammassa lasissa oli enemmän nestettä.

Keskustellessani Arto Hippulan kanssa [2003] pohdimme mahdollisuutta, että sokea lapsi voi ymmärtää monta asiaa näkeviä ikätovereitaan aiemmin käsitteellisellä tasolla. Sokealla lapsella on jo varhain tarve kehittää käsitteellisempää ajattelua kuin näkevällä lapsella, koska hän ei voi pohjata päätelmiään näkemäänsä maailmaan. Itse uskoisin lisäksi sokean lapsen välttävän tämän käsitteellisen ajattelun aikaisen heräämisen vuoksi eräänlaisilta ajattelun virheiltä, josta vesilasikoe on esimerkkinä. Toinen tunnettu esimerkki liittyy perunoiden keittämiseen. Näkevä lapsi voi olettaa, että perunat kypsyvät myös sellaisessa vedessä johon puhalletaan pillillä kuplia. Näkövammaisen lapsi luultavasti tietäisi, ettei kiehuvaan veteen voi laittaa sormiaan, mutta kuplivaan juomaveteen voi. Tällöin hän tietää myös perunoiden kypsyvän vain kuumassa vedessä.

Piaget'n teorian mukaisessa konkreettisten operaatioiden alkuvaiheessa lapsi on kuuden tai seitsemän vuoden ikäinen [Vilkko-Riihelä, 1999]. Tällöin lapsi pystyy jo eläytymään muiden asemaan paremmin kuin ennen ja kykenee matemaattisiin sekä loogisiin päätelmiin. Yhä silti sanonnat ja käsitteet liittyvät enemmän konkreettisiin kohteisiinsa kuin symbolisiin merkityksiinsä. Esimerkiksi sanonta 'katsoa sormien läpi' ei välttämättä tuo näkövammaisen lapsen mieleen yhtään mitään. Syntymästään asti sokea lapsi ei ole koskaan katsellut mitään. Näkevä lapsi voi ymmärtää sanonnan sen konkreettisessa merkityksessä. Hän ei näe sormiansa lävitse, koska sormet eivät ole läpinäkyvät. Hippulan [2003] kanssa käydyn keskustelun perusteella uskallan esittää väitteen, jonka mukaan sokealla lapsella on tarve kehittyä tällekin tasolle jo näkevää ennemmin.

Haluan korostaa sokeiden lasten olevan yksilöitä siinä missä näkevätkin lapset ovat [Poussu-Olli, 1999; Scholl, 1986]. Kodin kannustava ilmapiiri ja asioiden tarkka selittäminen lapsen kysyessä 'miten' tai 'miksi' - kysymyksiä, nopeuttavat tietenkin lapsen kehitystä. Aikuisten on helppo suojella näkövammaista lasta liikaakin, jolloin hänelle ei tarjoudu tilaisuuksia tutustua maailmaan itsenäisesti [Scholl, 1986]. Tämä voi herättää näkövammaisessa pelkoja tuntematonta ulkomaailmaa kohtaan, jolloin hän kieltäytyy edes yrittämästä tutkia sitä aktiivisesti.

5. Tuntopalautteen osa-alueet

Tuntopalautetta on monenlaista, aivan kuten näkemisessäkin on monta erilaista osa-aluetta. Tuntopalaute jaetaan useassa lähdeoteoksessani taktiiliin ja haptiikkaan [Schiff and Foulke, 1982; Hippula, 2003]. Poussu-Olli ja Keto [1999, s. 71] käyttävät termiä taktiiliset taidot, jolla tarkoitetaan sekä taktiilia että

haptista aistimista. Rajanveto näiden kahden välille on hieman hankalaa, koska useasti näitä kahta aistin osa-aluetta käytetään yhtäaikaaisesti.

Toisen jaottelun mukaan tuntopalautteen yläkäsitteenä voidaan pitää haptiikkaa [Schiff and Foulke, 1982, s.263-304]. Tällöin alakäsitteinä ovat mm. tekstuuri, lämpö, muoto, pehmeys, paino, muokattavuus, kimmoisuus, kosteus, kappaleen liikkuminen, tuntoaistimuksen vaihtelevuus sekä vaihtelevuuden taajuus. Thayer [Schiff and Foulke, 1982, s.263-304] ottaa lisäksi esille myös tuntemuksen keston.

Jokainen meistä saa jatkuvaa tuntopalautetta toimiessaan normaaleissa oloissa. Sensorisen deprivaation kokeissa [Vilkko-Riihelä, 1999] on todettu, ettei ihminen pysy kauaa normaalilla tajunnan tasolla kun kaikki tunto- ja ääniärsykkeet poistetaan. Kaikista ärsykkeistä eristetty ihminen kuuntelee paremman puutteessa mielellään vaikka puhelinluettelon ääneen lukemista säilyttääkseen järkensä.

Suunniteltaessa tuntopalautetta käyttävää järjestelmää on mielestäni tärkeää tehdä palautteesta soveltuvien osien realistista, mutta muistettava hyödyntää myös virtuaaliodellisuuden mahdollistamat luonnottomat tuntoaistimukset. Miksi jättäisimme käyttämättä virtuaalimaailman rikkaudet antaa erilaisia aistimuksia?

5.1. Taktiili

Taktiilia informaatiota saadaan tunnustelemalla kohdetta. Sormenpäiden, tai muun tunnustelussa käytetyn objektin, kautta välittyy tietoa kohteen pinnan tekstuurista. Taktiilista aistia sokeat käyttävät muunmuassa sokeainkirjoituksen lukemiseen ja erilaisten tuntomerkkien perusteella tavaroiden erotteluun. Jokaisessa eurosetelissä on esimerkiksi erilainen vesileima, joka on samalla kohokuviona tunnettavissa.

Aiemmin mainitussa Colwell *et al.* [1998] tekemässä tutkimuksessa selvitettiin virtuaalisen pinnan tekstuurin hahmottamista 'tuntotikun' välityksellä, objektin koon ja kulman hahmottamista sekä monimuotoisten virtuaalisten kohteiden (kuten tuoli ja sohva) tunnistamista. Erilaisten virtuaalitekstuurien erottelutehtävissä näkevät koehenkilöt eivät arvioineet samoin virtuaalisia ja 'aitoja' vertailutekstuureja. Sen sijaan sokeiden koehenkilöiden arviot virtuaalisista tekstuureista korreloivat 'aidoista' tehtyjen arvioiden kanssa. Sokeiden havaintokyky oli tässä tehtävässä erottelevampi kuin näkevien. Suurimmaksi syyksi Colwell *et al* [1998] epäilivät sokeiden harjoituksen myötä herkistynyttä kykyä tunnustella erilaisia pintoja ja esineitä.

Sokeiden taktiilista aistia kehitetään ja harjaannutetaan jo aivan pienestä lapsesta alkaen. Tärkeää on Poussu-Ollin ja Kedon [1999] mukaan, että alusta asti opetetaan sokealle oikea tekniikka tuntoaistin käyttämiseen. Kun

mahdollisimman varhaisessa vaiheessa aloitetaan erilaisten pintojen ja esineiden koskettelu paraneen sormien tuntoherkkyys [Poussu-Olli ja Keto, 1999]. Uskoisin lapsien oppivan yhtäläillä virtuaalisten maailmojen pinnoista ja merkityksistä mikäli heidät totutettaisiin niihin varhain. Virtuaalisesti voidaan antaa myös luonnotonta palautetta tehden esimerkiksi nuppineulanpään kokoisesta kappaleesta todella painava. Tällainen mahdollisuus tulisi mielestäni käyttää hyväksi. Lapsien totuttamisessa erilaisiin välineisiin voidaan käyttää pelejä ja leikkejä. Oppiminen olisi hauskaa, ja myöhemmin opittua voisi soveltaa tietokoneen hyödyntämiseen oppimisen välineenä tai työvälineenä.

Taktiiliksi aistimukseksi voisi lukea myös kosteuden- ja lämmön-aistimukset. Saadakseen aistimuksen tällaisesta käsitteestä ei tarvitse liikuttaa tunnusteluun käytettyä kehonosaa, vaan aistimus on saatavissa yhdestäkin pisteestä. Toisaalta mahdollisuus tunnustella pintaa tai kohdetta laajemmin auttaa tarkemman kuvan muodostumisessa. Esimerkiksi hieman kastunut paidanhiha saattaa ensisijaisesti tuntua iholla kylmältä. Laajempi tunnustelu muuttaa kuitenkin hihasta saatua kuvaa niin, että vaate tuntuu kostealta ja siksi se on myös kylmä päällä.

5.2. Haptiikka

Haptiikka liittyy tuntopalautteeseen, jossa informaatio saadaan kehon asennon muutoksien ja lihasten jännittyneisyyden mukaan. Haptinen havaitseminen voidaan määritellä myös ”aktiiviseksi käsien ja sormien käytöksi” havaitsemisessa, kuten David H. Warren sen tekee [Schiff and Foulke, 1982, s.82-83].

Poussu-Olli ja Keto [1999] korostavat myös haptisten taitojen harjaannuttamisen varhaisen aloittamisen merkitystä, jotta aistista myöhemmin saataisiin kaikki mahdollinen hyöty. Oikeaan tekniikkaan heidän mukaansa kuuluu muunmuassa molempien käsien yhtäaikaista käyttöä tunnustelussa. Tällöin kolmiulotteisuus ja kappaleiden koko on helpompaa havaita.

Colwell *et al.* [1998] käsittelee tutkimuksessaan haptista havaitsemista. Kappaleen koon ja kulman määrittelyä tehtävässä, sekä sokeat että näkevät olivat samalla tasolla. Molemmat pystyivät tarkempiin arvioihin kun kappaleen koko kasvoi. Kun arvioitiin monimutkaisempia kappaleita, kuten keittiötuolia ja sohvia, törmättiin ongelmaan. Huonekaluja oli hankala erottaa vaikka koehenkilölle kerrottiin etukäteen minkä nimistä kohdetta tämä tunnusteli. Colwell *et al.* [1998] arvioivat loppupuheenvuorossaan tämän johtuvan tunnusteluvälineen heikkouksista. Yksi tunnustelutikku ei anna mahdollisuutta käyttää molempia käsiä yhtäaikaaisesti eikä tuntoaistimus tule suoraan pinnasta vaan tikun välittämänä. Lisäksi tunnustelutikun luoma vastustava voima on

rajallinen, jolloin tarpeeksi painamalla on mahdollisuus läpäistä pintoja eritavoin kuin se reaali maailmassa on mahdollista.

6. Tuntopalautteen hyödyntämismahdollisuudet lasten tietokonesovelluksissa

Suurin osa 5-7-vuotiaista lapsista ei osaa lukea. Sokeat lapset oppivat lukemaan sokeainkirjoitusta ja käyttämään braille-näyttöjä vasta paljon harjoittelun jälkeen. Taidon oppimiseen tarvitaan puhumaan oppimisen lisäksi myös riittävä herkkyys sormenpäissä [Poussu-Olli ja Keto, 1999]. Tämän vuoksi puhekäyttöliittymät ja tuntopalautetta hyödyntävät käyttöliittymät ovat esikouluikäisille ainoita mahdollisia.

Tietokonesovellukset ovat mielestäni lapsille hyviä leikkiympäristöjä, kunhan pelien laatuun ja tarkoitukseen on käytetty tarpeeksi suunnittelu-aikaa. Tietokone on väsymätön harjoittaja ja pelivastus. Lisäksi yhteiskäytön mahdollistavat sovellukset ovat omiaan ohjaamaan lapsia yhteistoiminnallisuuteen ja sosiaalisuuteen, varsinkin silloin kun peli ei etene elleivät lapset toimi yhdessä. Huonokin viihdyke tietokone voi olla, mikäli pelataan merkityksettömiä pelejä, jotka mahdollisesti vahvistavat vääriä käyttäytymismalleja tai sekoittavat jo opittuja taitoja.

Tuntopalautteen ei havaittu lisäävän sosiaalisen läsnäolon tuntua Salnässin *et al.* [2000] tutkimuksessa. Virtuaalisen läsnäolon tuntua tuntopalautteen käyttö sen sijaan lisäsi. Lisäksi tutkimuksessa tehdyt yhteistoiminnalliset tehtävät sujuivat, sekä koehenkilöiden mielestä että mitattujen arvojen mukaan, paremmin ja nopeammin. Harjoiteltaessa tuntopalautteen vastaanottamista näkövammaisten lasten kanssa uskoisin tällaisten yhteistoiminnallisten pelien tekemisen oppimisesta hauskaa. Lisäksi ystävä voi tuoda rohkeutta tutkia ympäristöä enemmän kuin yksinään. Ystävän kanssa voi pohtia mahdollisesti herääviä kysymyksiä yhdessä.

Sjöström [2001] tutki näkövammaisilla ja näkevillä lapsilla tunto- ja auditiivistapalautetta antavaa muistipeliä verraten sen käyttöä vastaavaan visuaalisauditiivistapalautetta hyödyntävään muistipeliin. Hiirtä, näyttöä ja ääniä käyttäneet testihenkilöt saivat pelin päätökseen nopeammin kuin PHANToM-laitetta ja ääniä käyttäneet. Tämä saattaa johtua näköaistin nopeudesta informaation välittäjänä. Muutamaa poikkeusta lukuunottamatta kaikki sokkoina muistipeliä pelanneet saivat pelin kuitenkin päätökseen ja jokainen onnistui löytämään muutaman parin.

Sjöströmin [2001] tarkoituksena oli muistipelitutkimuksessa tutkia onko Windowsin kaltaista graafista käyttöjärjestelmää mahdollista käyttää pelkän tunto- ja äänipalautteen perusteella. Hänen tutkimustuloksensa osoittavat

samalla yhden mahdollisen pelin, jonka kautta lapsi harjoittelee erilaisten ikoneiden tunnistamista ja valintaa muistinsa kehittämisen lisäksi. Samaisessa Sjöströmin tutkimuksessa [2001] on muitakin testattuja ohjelmia. Mielestäni hieno oivallus on sokeille suunnattu piirto-ohjelma, jossa eri väreillä on tunnettava ja kuultava ilmiasu. Piirtämisen lisäksi sokea voi näin tunnustella ja hahmottaa tarkoin muiden piirtämiä kuvia.

Matematiikan opettaminen on varsinkin geometriaan ja tilastotieteeseen liittyen osittain visuaalista. Kuvaajien tunnusteleminen ja geometrinen kuvioiden manipulointi voidaan kuitenkin tehdä mahdolliseksi myös sokeille, jolloin asian ymmärtäminen mahdollisesti helpottuu. Warren [Schiff and Foulke, 1982] kertoo tutkimuksesta, jossa 3- ja 4-vuotiaille lapsille opetettiin kuusikulmio terminä ja muotona. Opetusmetodeina mahdollisuus kohteen manipulointiin ja koskettamiseen nostivat kuusikulmio-termin ja -muodon oppimistuloksia verrattuna pelkkään visuaaliseen opetustapaan. Tuntopalautteella oli täten parantava vaikutus lopputulokseen. Kyseisessä kokeessa käytettiin reaali maailman kohteita, mutta uskoisin saatavan samanlaisia tuloksia vaikka koe suoritettaisiin virtuaalisilla kappaleilla.

7. Tuntopalautteen rajoitukset

Tuntopalautteella on nykyisin laittein montakin rajoitusta. Useat laitteet antavat mahdollisuuden vain yhdellä kädellä tapahtuvaan tunnusteluun ja informaatio saadaan tavallaan "toisen käden kautta" kun pinnan ja muotojen välittäjänä käytetään jotakin tunnustelutyökalua [Colwell *et al.*, 1998]. Tällöin tunnustelemalla muodostunut kuva on paljon vajavaisempi, kuin se voisi oikealla tekniikalla [Poussu-Olli ja Keto, 1999] tunnusteltuna olla.

Tutkimuksessa, jossa pyrittiin tunnistamaan virtuaalisia huonekaluja [Colwell *et al.*, 1998] oli huonekalut muodostettu erikokoisista geometrisista kappaleista. Tunnusteltaessa oli mahdollista, että tunnustelijan tuntotikku 'putosi' kappaleiden väliin jääneisiin rakoihin. Tämä vaikeutti tunnistusta, koska raoilla ei ollut mitään merkitystä reaali maailmassa ja tunnustelijan oli ymmärrettävä nostaa tikku pois raosta ennenkuin tunnustelua voitiin jatkaa.

Tekniikka on verrattain uutta, ja sen käyttö vaatii harjoittelua. Useassa tutkimuksessa arveltiin harjoittelun parantavan tuloksia [Jansson and Billberger, 1998; Kurze, 1997; Ramstein, 1996]. Tämä on tuettavissa oleva oletus, sillä kaikki uusi tekniikka vaatii harjoittelua ja totuttelua. Täysin uuteen lähestymistapaan en silti lähtisi, sillä jo muodostuneita mentaalimalleja olemassaolevista sovelluksista täytyy pyrkiä tukemaan. Lisäksi yhteiskäytössä sokeiden ja näkevien välillä tulee molemmilla olla mahdollisuus puhua samoin mielikuvien käytettävästä sovelluksesta [Mynatt and Weber, 1994]. Tätä

näkökantaa kannattivat myös Sjöströmin tutkimuksen [2001] sokeat koehenkilöt.

Tunnustelemalla tapahtuva virtuaalisten kappaleiden muodon tunnistaminen oli Janssonin ja Billbergerin [1998] tutkimuksen mukaan aitojen kappaleiden tunnustelua hitaampaa ja epätarkempaa. Arviot virtuaalikappaleista tarkentuivat kuitenkin kappaleiden koon kasvaessa [Jansson and Billberger, 1998]. Huomioimalla tämä seikka käyttöliittymän suunnittelussa saadaan sen vaikutus pienenemään tai jopa katoamaan.

”Tardis efekti” tulee ottaa huomioon suunniteltaessa tunnusteltavia virtuaalikappaleita [Colwell *et al.*, 1998]. Tardis efekti tarkoittaa ilmiötä, jonka mukaan tunnusteltava kappale arvioidaan usein suuremmaksi sisältä tunnusteltaessa kuin ulkoa tunnusteltaessa. Efekti ei ole välttämättä huono puoli, mutta tärkeää on huomioida sen vaikutukset suunnittelussa, jotta kappaleista muodostuva kuva on mahdollisimman realistinen. Mikäli halutaan tarkoituksella luoda vain virtuaalimaailmassa mahdollisia kappaleita ja muotoja voidaan tietoa tardis efektistä hyödyntää myös.

8. Lopuksi

Tutkimus on suppea verrattuna kaikkeen tutkimukseen ja lähdemateriaaliin, josta olisi ollut mahdollista kirjoittaa. Tarkoitus ei kuitenkaan ollut tehdä tästä kartoituksesta tämän pidempää. Uskoakseni tarvittava tieto on nyt koottu yhteen, ja jokainen jota asia lähemmin kiinnostaa, voi perehtyä viitteisiini ja muuhun aiheeseen liittyvään tutkimukseen tarkemmin itsenäisesti.

Itselleni selvisi moni asia näkövammaisuudesta kartoituksen tekemisen yhteydessä. Toivon näiden oivallusten välittyneen myös lukijalle. Tärkeää mielestäni on ymmärtää, etteivät kaikki näkövammaiset suinkaan ole täysin sokeita. Lisäksi ymmärrys sokeuden vaikutuksista on tärkeää. Näkövammainen ei ole tyhmempi kuin näkevä, vaan vamman myötä hän on menettänyt yhden kanavan saada tietoa. Kaikki näkövammaisten koulutus ja erityisvälineet pyrkivät kompensoimaan tämän yhden aistin puutteita, jotta näkövammainen pääsisi samalle viivalle näkevien kanssa. Vammasta voi olla myös hyötyä, jos se johtaa nopeampaan kognitiiviseen kehitykseen ja syvempään käsitteiden ymmärtämiseen.

Lähteitä kartoittaessani huomasin kuinka paljon psykologiassa on tutkittu kuuloa ja sen puuttumisen vaikutuksia verrattuna sokeuden vaikutuksiin. Hahmottamista ja näköaistia on tutkittu laajasti, muttei tällaisten kykyjen täydellistä tai osittaista puuttumista. Tuntoaistia käsitellään usein sosiaalisen koskemisen näkökannalta.

Tietojenkäsittelyn ja vuorovaikutteisten teknologioiden tutkimuksessa on tähänasti keskitytty aikuisille suunnattuihin sovelluksiin. Hyvin vähän löysin tutkimuksia nimenomaan alle kouluikäisten lasten sovelluksista. Selkeästi oli havaittavissa, että lapsille suunnattujen sovellusten kehittäminen ja tutkimus on kuitenkin nouseva trendi. Ehkä aikuisten näkövammaisten työskentely näkevien rinnalla on nykyään mahdollistettu tutkimuksen tulosten ansiosta ja mielenkiinto kohdistuu seuraavaan kohderyhmään.

Viiteluettelo

- [Baloian *et al.*, 2002] Nelson Baloian, Wolfram Luther and Jaime Sánchez. Modeling Educational Software for People with Disabilities: Theory and Practise. *Proceedings of the fifth international ACM conference on Assistive Technologies*, ACM Press, 2002, 111-118.
- [Brehm and Salisbury, 1998] Denise Brehm, Kenneth Salisbury. *Device turns computer into tactile tool, lets user touch virtual objects*.
Saatavilla: <http://web.mit.edu/newsoffice/nr/1998/phantom.html>. Linkki tarkistettu 14.12.2003.
- [Boyd et al., 1990] Boyd L.H., Boyd W.L. and Vanderheiden G.C. The Graphical User interface: crisis, danger and opportunity. *A Journal of Visual Impairment and Blindness*, **84** (December), 1990, 496-502.
- [Christian, 2000] Kevin Christian. Design of Haptic and Tactile Interfaces for Blind Users. Unpublished manuscript, April 19, 2000.
Saatavilla: <http://www.otal.umd.edu/UUGuide/kevin/>. Linkki tarkistettu 14.12.2003.
- [Colwell *et al.*, 1998] Chetz Colwell, Helen Petrie, Diana Kornbrot, Andrew Hardwick and Stephen Furner. Haptic Virtual Reality for Blind Computer Users. *Proceedings of the third international ACM conference on Assistive technologies*, ACM Press, 1998, 92-99.
- [Hinckley and Sinclair, 1999] Ken Hinckley and Mike Sinclair. Touch-sensing input devices. *Proceedings of the SIGCHI conference on Human factors in computing systems: The CHI is the limit*, 1999, 223-230.
- [Hippula, 2003] Arto Hippula, keskustelu 25.11.2003.
- [Jansson and Billberger, 1998] Gunnar Jansson and Katarina Billberger. The PHANToM Used without Visual Guidance. *The First PHANToM Users Research Symposium*, 1998.
Saatavilla: <http://mbi.dkfz-heidelberg.de/purs99/proceedings/Jansson.pdf>. Linkki tarkistettu 14.12.2003.

- [Kurvinen-Tikkanen, 1999] Anna-Maija Kurvinen-Tikkanen. *Tietoa näöstä, näkemisen vaikeuksista ja näkövammaisuudesta*, Näkövammaisten keskusliitto ry., 1999
- [Kurze, 1996] Martin Kurze. TDraw: A Computer-Based Tactile Drawing Tool for Blind People. *Proceedings of the second annual ACM conference on Assistive Technologies*, ACM Press, 1996, 131-138.
- [Kurze, 1997] Martin Kurze. Rendering drawings for interactive haptic perception. *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, 1997, 423-430.
- [Kurze, 1998] Martin Kurze. TGuide: A Guidance System for Tactile Image Exploration. *Proceedings of the third annual ACM conference on Assistive Technologies*, ACM Press, 1998, 85-91.
- [Mynatt and Weber, 1994] Elizabeth D. Mynatt and Gerhard Weber. Nonvisual Presentation of Graphical User Interfaces: Contrasting Two Approaches. *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, 1994, 1, 166-172.
- [Näkövammaisten Lasten Tuki ry., 2003] Näkövammaisten Lasten Tuki ry. 4.9.2003.
Saatavilla: <http://www.sci.fi/~nlt/index.htm>. Linkki tarkistettu 14.12.2003.
- [Ojamo, 2000] Ojamo Matti. Sosiaali- ja terveystieteiden tutkimus- ja kehittämiskeskus ja Näkövammaisten keskusliitto ry., Helsinki, 2001.
Saatavilla: <http://www.nkl.fi/nvrek/vuosikirja2000/>. Linkki tarkistettu 14.12.2003.
- [Petrie et al., 1995] Helen Petrie, Sarah Morley and Gerhard Weber. Tactile-Based Direct Manipulation in GUIs for Blind Users. *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1995, 2, 428-429.
- [Piaget, 1977] Piaget Jean. *Lapsen psykologia*. Gummerrus, 1977.
- [Poussu-Olli ja Keto, 1999] Hanna-Sofia Poussu-Olli ja Leena Keto. *Näkövammaisuus*. Turun yliopiston kasvatustieteiden laitos, 1999.
- [Ramstein, 1996] Christophe Ramstein. Combining haptic and braille technologies: design issues and pilot study. *Proceedings of the second annual ACM conference on Assistive technologies*, 1996, 37-44.
- [Sallnes et al., 2000] Eva-Lotta Sallnes, Kirsten Rasmus-Gröhn and Calle Sjöström. Supporting presence in collaborative environments by haptic force feedback. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2000, 461-476.
- [Schiff and Foulke, 1982] William Schiff and Emerson Foulke, *Tactual Perception: a Sourcebook*. Cambridge University Press, 1982.

- [Scholl, 1986] Geraldine T. Scholl. *Foundations of Education for Blind and Visually Handicapped Children and Youth*. American Foundation for the Blind, 1986.
- [Sjöström, 2001] Calle Sjöström. *Designing Haptic Computer Interfaces for Blind People*. ISSPA, 2001.
- Saatavilla: <http://www.certec.lth.se/doc/designinghaptic/designinghaptic.pdf>.
Linkki tarkistettu 14.12.2003.
- [Vilkko-Riihelä, 1999] Anneli Vilkko-Riihelä. *Psyhyke Psykologian Käsikirja*. WSOY, 1999.

Puhekäyttöliittymä osana virtuaalitodellisuusympäristöjä

Mark Claydon

Tiivistelmä.

Tutkimuksessa tarkastellaan puhekäyttöliittymän lisäämistä virtuaalitodellisuussovelluksiin vuorovaikutuskeinona. Tarkastelun kohteena on myös puheentunnistukselle virtuaalitodellisuudessa asetetut vaatimukset ja puheohjauksen käytettävyysskysymykset virtuaalitodellisuusympäristöissä. Tutkimuksen on tarkoitus toimia analyysina tehokkaista tavoista toteuttaa puhekäyttöliittymä osana virtuaalitodellisuusympäristöä jossa voi olla käytössä samanaikaisesti myös muita vuorovaikutuskeinoja. Lisäksi arvioidaan onko tällainen multimodaalinen virtuaalitodellisuusmaailma käyttäjälle mielekäs ympäristö sekä onko sen käyttö helppoa tai kannattavaa, myös toteutuksen kannalta oleellisia asioita käydään läpi tutkimuksessa.

Avainsanat ja -sanonnat: Virtuaalitodellisuus, puhekäyttöliittymät, puheohjaus, 3d-näyttölaitteet, multimodaalinen vuorovaikutus, ihmisen ja tietokoneen välinen vuorovaikutus.

CR-luokat: H.5.1, H.5.2.

1. Johdanto

Virtuaalitodellisuusjärjestelmät ovat nykyään yleistymässä koska niihin tarvittava tekniikka kehittyy ja on myös edullisempaa kuin aikaisemmin. Virtuaalitodellisuutta hyödyntäviä sovelluksia kehitetään erilaisia käyttötarkoituksia ja -kohteita varten. Erityisesti kolmiulotteista suunnittelua ja mallintamista vaativilla aloilla virtuaalitodellisuustekniikoista on paljon hyötyä, mutta myös esimerkiksi niin lääketieteellisiä ratkaisuja kuin viihdesovelluksiakin kehitetään erilaisiin virtuaalitodellisuusympäristöihin. Kolmiulotteisen syvyysvaikutelman aikaansaava virtuaalitodellisuus mahdollistaa erilaisten vuorovaikutustekniikoiden käytön paremmin kuin esimerkiksi pelkällä kaksiulotteisella näyttölaitteella esitetty käyttöliittymä. Pääasiassa siksi koska todellista, eli kolmiulotteista maailmaa, vastaavassa ympäristössä on luontevampaa käyttää tuntopalautetta, kolmiulotteista äänimaisemaa tai erityyppisiä virtuaalitodellisuusjärjestelmille ominaisia ohjausmekanismeja, kuten datahanskaa, kuin ympäristössä jossa visuaalinen palaute annetaan vain kaksiulotteisen näyttölaitteen kautta.

Toki vastaavia tekniikoita on kehitetty käytettäväksi myös kaksiulotteiseen visuaaliseen palautteeseen yhdistettynä; näissä sovelluksissa virtuaalitodellisuuden vaikutus aikaansaadaan muilla menetelmillä, kuten käyttämällä käyttöliittymässä mahdollisen realistisia ohjausmekanismeja tai antamalla käyttäjälle mahdollisimman todentuntuista palautetta eri menetelmillä.

Virtuaalitodellisuusympäristön ja käyttäjän välillä kuitenkin on usein vahva visuaalinen vuorovaikutus ja yleensä mukana on myös joitakin edellä mainituista tilavaikutelman aikaansaavista vuorovaikutusmenetelmistä. Tämän vuoksi virtuaalitodellisuuteen on luontevaa lisätä myös luonnollinen kieli, eli puheohjaus ja puhepalautte, osaksi käyttöjärjestelmää. Puhe on kuitenkin ihmisen luonnollisin viestintätapa ja siksi se sopiikin hyvin käytettäväksi sovellukseen, jonka tarkoitus on esittää käyttäjälle mahdollisimman realistinen ja luonnollinen ympäristö. [Grasso et al., 1998; McGlashan, 1995]

Tässä tutkimuksessa tarkastellaan mahdollisuuksia yhdistää puhe käyttöliittymä virtuaalitodellisuusympäristöön, sekä millaisia vaatimuksia puhe käyttöliittymän tulee täyttää jotta se soveltuu käytettäväksi osana multimodaalista virtuaalitodellisuusjärjestelmää.

2. Virtuaalitodellisuus

2.1. Määritelmiä

Pääasiassa eri viestintävälineissä esiintyneiden aihetta koskevien esitysten vuoksi virtuaalitodellisuudella on nykyään hyvin monta erilaista määritelmää, ja virtuaalitodellisuudella voidaankin tarkoittaa lähes mitä tahansa datakypäristä ja muista kolmiulotteisista näyttölaitteista aina yksinkertaisiin tietokonepeleihin asti. Lisäksi varsinainen termi "virtuaalitodellisuus" tarkoittaa käytännössä jotakin, mitä ei vielä nykytekniikalla pystytä toteuttamaan, ja usein puhutaankin joko virtuaalimaailmoista tai virtuaaliympäristöistä [Feiner and MacIntyre, 1996]. Tässä tutkimuksessa käytän kuitenkin termiä virtuaalitodellisuus johtuen sen yleisyydestä niin suomenkielisessä tieteellisessä tekstissä kuin arkipäiväisessäkin viestinnässä. Feiner ja MacIntyre [1996] määrittelevät virtuaalitodellisuuden järjestelmäksi joka esittää ympäristön tai käyttöjärjestelmän kolmiulotteisena, tarjoaa interaktiivisen ympäristön ja käyttää jotakin menetelmää tilavaikutelman luomiseen. Nämä menetelmät voivat olla visuaalisia, kolmiulotteista ääntä hyödyntäviä, tai kosketukseen, tuntopalautteeseen ja kehon liikkeisiin perustuvia. Samassa artikkelissa esitellään myös määritelmiä erilaisille virtuaalitodellisuusympäristöille:

Keinotodellisuus (artificial reality) tarkoittaa yleisimmin järjestelmää, joka ei vaadi käyttäjälle teknisiä laitteita, esimerkiksi datahanskoja tai -kypäriä, virtuaalitodellisuuden kokemiseksi tai hallitsemiseksi. Tällaisia sovelluksia ovat mm. erilaiset ympäristöt, missä käyttäjälle luodaan tilavaikutelma kolmiulotteisena kuvana erillisille näyttölaitteille sekä kolmiulotteisen äänimaailman avulla.

Lisätty todellisuus (augmented reality) tarkoittaa järjestelmää jossa käyttäjälle esitetään oikea maailma jollain tavoin tietokoneen avulla vahvistettuna. Esimerkiksi käyttämällä datakypäriä, jonka näyttölaitteen läpi käyttäjä näkee oikean maailman näytölle piirrettävän kolmiulotteisen tietokonekuvan vahvistamana.

Etäläsnäolo (telepresence) tarkoittaa järjestelmää jossa käyttäjälle esitetään kuva jossakin muualla olevasta ympäristöstä kameroiden ja näyttölaitteen avulla. Käytettäessä erityistä kypäränäyttöä käyttäjän päähän liikkeet voidaan yhdistää liikuttamaan kameroita, jolloin vaikutelma läsnäolosta jossakin muualla vahvistuu. Tällaisia järjestelmiä voisivat olla esim. kehittyneet etäneuvottelujärjestelmät.

Etätoiminta (teleoperation) tarkoittaa järjestelmää jossa käyttäjä ohjaa jotakin etäällä olevaa laitetta käyttäen apuna tarkoitukseen suunniteltua ohjainta ja esim. laitteen lähettämää kuvaa. Tällaisia järjestelmiä ollaan suunnittelemassa mm. lääketieteessä etäkirurgian käyttöön ja muihin vastaaviin sovelluksiin.

Upottavat järjestelmät (immersive systems) ovat usein multimodaalisia, hyvin kehittyneitä virtuaalitodellisuusympäristöjä, joissa katsojan näkökulma sijoittuu keskelle näennäismaailmaa. Virtuaalitodellisuuden tunne saadaan aikaan käyttämällä HMD-näyttölaitteita (Head Mounted Display, datakypäri), tai erityistä heijastusnäyttötekniikkaa. Lisäksi usein käyttäjälle tarjotaan myös kolmiulotteinen äänimaisema. Näissä järjestelmissä käyttäjä ”upotetaan” sisään virtuaalimaailmaan.

2.2. Virtuaalitodellisuuden toteutuksesta

Tässä tarkastellaan multimodaalisen virtuaalitodellisuusjärjestelmän toteuttamiseen vaadittavia tekijöitä, kuten järjestelmän toteuttamiseen käytettäviä laitteita ja niille asetettavia vaatimuksia, sekä järjestelmän suunnittelussa huomioitavia asioita. Kolmiulotteisen äänimaailman luomista ei tarkastella erikseen koska siihen käytettävät menetelmät ovat melko yksinkertaisia ja kotiteatterijärjestelmistäkin tuttuja.

2.2.1. 3D-näyttölaitteet

Käyttämällä kolmiulotteisen näkymän tarjoavia näyttölaitteita saadaan virtuaalitodellisuusympäristöön visuaalinen syvyysvaikutelma ja tunne käyttäjää ympäröivästä maailmasta. Stereokuva luodaan erityisellä projektorilla, joka heijastaa näyttölaitteelle kuvan molemmille silmille. Käyttäjällä on päässään lasit, joissa käyttämällä elektronisia sulkimia peitetään molemmat silmät vuoronperään siten että kumpikin silmä näkee juuri sille tarkoitetun kuvan [Vince, 1995]. Tätä tekniikkaa voidaan käyttää näyttämällä käyttäjälle kuva heijastusnäyttötekniikalla (esim. seinä tai kangas jolle stereokuva heijastetaan), jolloin käyttäjä pitää päässään tarkoitukseen suunniteltuja sulkimilla varustettuja laseja. Tekniikkaa voidaan käyttää myös HMD-näyttölaitteilla, jotka sisältävät tarpeellisen tekniikan stereokuvan tuottamiseen.

Jos virtuaalitodellisuussovellus ei käytä stereokuvaa, on syvyysvaikutelma luotava muilla tavoilla. Syvyyden vaikutelmaa voidaankin tukea erilaisilla syvyysvihjeillä joita tarjoavat stereokuvan lisäksi mm. perspektiivi, liike, varjot ja värit [Turunen, 1998]. Etenkin yhdessä käytettäessä näillä voidaan tuottaa erittäin realistinen syvyysvaikutelma.

2.2.2. Käyttäjän liikkeen ja paikan seuraaminen

Käyttäjän liikkeitä voidaan virtuaalitodellisuusympäristöissä seurata mm. sähkömagneettisilla, akustisilla sekä optisilla paikannusmenetelmillä.

Sähkömagneettiset paikannusmenetelmät ovat tällä hetkellä laajimmin käytössä. Niissä käytetään kolmen toisiaan vastaan kohtisuorassa olevan käämien ryhmää, joiden muodostamaa magneettikenttää vastaanotetaan samanlaisella käämien ryhmällä. Haittapuolena tässä tekniikassa on magneettikentän rajattu koko: kentänvoimakkuus laskee jos etäisyys lähettimestä kasvaa liikaa. Magneettikentän alueella ei saa olla mitään ferromagneettisia esineitä. [Feiner and MacIntyre, 1996; Kalawsky, 1993]

Akustiset paikannusmenetelmät perustuvat äänen kulkuaikaan ja äänen kulkemasta matkasta johtuviin vaihe-eroihin, käytetty äänisignaali on ultraäänialueella. Tämä tekniikka on melko halpaa ja sopii siksi käytettäväksi esimerkiksi viihdesovelluksissa. [Feiner and MacIntyre, 1996]

Optisissa paikannusmenetelmissä käytetään yleensä mm. valoa rekisteröiviä kameroita. Kameran voivat olla kiinni joko itse virtuaalitodellisuussovelluksen käyttöympäristössä tai sitten käyttäjällä olevassa varustuksessa. Kameran seuraavat joko ympäristössä tai käyttäjän varustuksessa olevien yleensä pistemäisten valonlähteiden liikkeitä, ja seuraavat näin käyttäjän liikkumista ja eleitä. Valonlähteiden ja kameroiden

väliin ei saa kuitenkaan asettaa mitään esteitä, jotta paikannus toimisi kunnolla. [Feiner and MacIntyre, 1996; Kalawsky, 1993]

2.2.3. Tunto- ja voimapalaute

Käyttämällä datahanskaa ohjaimena virtuaalitodellisuusympäristössä voidaan käyttäjälle antaa myös jonkinlaista tunto- ja voimapalautetta. Datahanska voi estää käyttäjää mm. koukistamasta sormiaan jos käyttäjän sormet ovat virtuaalitodellisuudessa jonkin esineen ympärillä. Tämä toteutetaan mittaamalla käyttäjän käden paikka virtuaalimaailmassa datahanskan antureiden avulla, ja sen jälkeen käyttämällä hanskassa olevia laitteita vastustamaan käyttäjän sormien liikkeitä voimapalautteen avulla. [Kalawsky, 1993] Toistaiseksi käden paikantamiseen käytetyt menetelmät eivät vielä ole kuitenkaan riittävän tarkkoja tarjotakseen käyttäjälle aidon tunteen siitä että hän pitäisi jotakin esinettä kädessään. Myös erityyppisissä virtuaalitodellisuutta hyödyntävissä simulaattoreissa voidaan käyttää voimapalautetta samalla periaatteella luomaan vaikutelma luonnollisesta ympäristöstä.

Datahanskojen lisäksi ollaan kehittämässä myös erilaisia datapukuja, joilla käyttäjän liikkeitä voidaan seurata tarkemmin, ja joilla voidaan antaa myös monipuolisempaa voimapalautetta.

2.2.4. Laitteisto- ja ohjelmistovaatimuksista

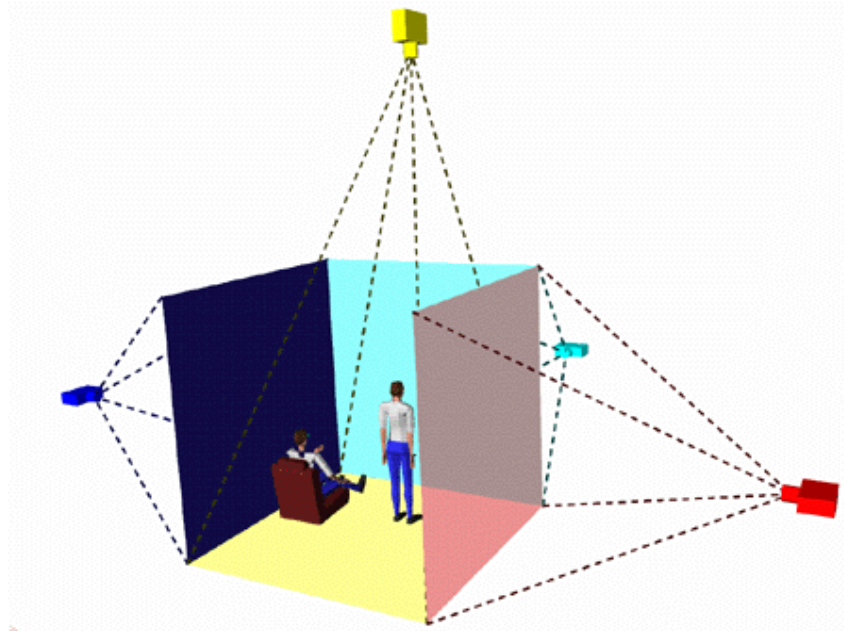
Raskain toimenpide virtuaalitodellisuutta luotaessa on 3D-kuvien generointi, joka luonnollisesti vaatii käytettävältä laitteistolta paljon tehoa. Nykyään tietotekniikka on jo niin kehittynyttä, että monimutkaistakin grafiikkaa voidaan tuottaa reaaliaikaisesti, mutta usein virtuaalitodellisuusympäristöjen toteuttamiseen käytetään tavallista tehokkaampia työasematietokoneita. Virtuaalitodellisuuden realistisuutta mitataan usein polygonien määrällä, ja sillä kuinka monta kuvaa sekunnissa tällä määrällä voidaan esittää. Nykyiset laitteet pystyvät tuottamaan jopa satojatuhansia polygoneja sisältäviä kohteita riittävän nopeasti. Todellisuuden simulointiin tarvittaisiin kuitenkin todennäköisesti kymmeniä miljoonia polygoneja. [Lappeenrannan teknillisen korkeakoulun verkkosivut, 2003] Virtuaalitodellisuusympäristöihin grafiikka tuotetaan usein samoilla menetelmillä kuin normaalille tietokoneellekin.

Ohjaimina virtuaalitodellisuudessa voidaan käyttää edellä mainittujen datahanskojen ja simulaattoreiden omien ohjaimien lisäksi myös tarkoitukseen suunniteltuja hiiriä, joiden hallinnassa käytetään hyväksi kuutta vapausastetta, eli siis kolme paikka- ja kolme asentokoordinaattia. [Feiner and MacIntyre, 1996]

2.3. CAVE - esimerkki nykyaikaisesta virtuaalitodellisuusympäristöstä

CAVE (Cave automatic virtual environment) on tyypillinen upottava järjestelmä, eli siinä käyttäjällä on hyvin vahva tunne läsnäolosta virtuaalitodellisuudessa. CAVEssa käyttäjä on kuution mallisessa huoneessa, jossa kolmelle seinälle, lattialle sekä mahdollisesti myös kattoon heijastetaan stereokuvaa heijastusnäyttötekniikkaa ja virtuaalitodellisuuslaseja käyttäen. CAVE ympäristössä käyttäjän liikkeitä seurataan, ja tämän avulla ympäristöön heijastetaan jatkuvasti uutta kuvaa joka vastaa käyttäjän senhetkistä paikkaa. Käyttäjä voi vaikuttaa ympäristöön mm. datahanskan tai erikoishiiren avulla. [Vince, 1995] CAVE on nykytekniikalla toteutetuista virtuaalitodellisuusympäristöistä kaikkein lähimpänä virtuaalitodellisuuden perusajatusta, jossa käyttäjä on virtuaalimaailman ympäröimä ja jossa hänelle tarjotaan mahdollisimman luonnollisia vuorovaikutusmetodeja ympäristön kanssa, etenkin jos käytettävään järjestelmään on lisätty mm. puhekäyttöliittymä ja voima- tai tuntopalautelaitteet.

CAVE-ympäristöön voidaan mallintaa erilaisia kolmiulotteisia maailmoja, joita voidaan hyödyntää suunnittelu, tutkimus ja koulutustehtäviin mm. käyttämällä järjestelmää simuloimaan erilaisia työympäristöjä. Joskus tällaisiin simulaattorisovelluksiin liitetään myös kyseiselle sovellukselle ominaisia ohjaimia. Esimerkiksi jonkin simuloidun koneen todelliset hallintalaitteet voivat olla käyttäjän ja sovelluksen vuorovaikutusvälineenä.



Kuva 1: CAVE-ympäristö

[University of Michigan virtual reality lab, 2003]

3. Puhekäyttöliittymät

3.1. Puhekäyttöliittymän rajoitukset, hyödyt ja käytettävyys

Puhekäyttöliittymässä ohjataan käyttöliittymän toimintoja puheen avulla ja palautetta saadaan usein puhesynteesin avulla muodostetulla puheella. Puhekäyttöliittymässä komentojen tai vaikutusmahdollisuuksien määrä on usein rajallinen johtuen pääasiassa niissä käytettävän tekniikan rajoituksista. Mitä laajempi sanasto puhekäyttöliittymällä on tunnistettavia sanoja, sitä heikompi sanantunnistustarkkuus järjestelmällä on. Sanastoa voidaan nykyään kasvattaa melko suureksikin, mutta tällöin ohjelma osaa tunnistaa tarkasti vain yleensä yhden ihmisen äänen. Usean käyttäjän järjestelmissä sanasto on tästä syystä pienempi. Lisäksi puhekäyttöliittymät käyttävät vain tiettyjä niiden sanastosta löytyviä komentoja, joita käyttäjä ei välttämättä tunne tai hän ei osaa yhdistää niitä tilanteeseen. Tämän takia puhekäyttöliittymän käyttäjille on tarjottava riittävä opastus ja ohjelman on myös selviydyttävä virheistä käyttäjälle selkeällä tavalla. Tässä on kuitenkin otettava huomioon myös kokeneet käyttäjät joita liiallinen ohjeistus voi häiritä. [Barker and Weinschenk, 2000] Erityisesti virtuaalitodellisuusympäristöissä tarvittavia puhekäyttöliittymien erityisvaatimuksia käsitellään tutkimuksessa myöhemmin.

Puhe ei kuitenkaan korvaa graafisen käyttöliittymän elementtejä, sillä puhuminen saattaa, tarvittavasta komennosta riippuen, kestää kauemmin kuin hiirellä osoittaminen, ja lisäksi kuuntelu tapahtuu hitaammin kuin tekstin lukeminen tai muun visuaalisen palautteen rekisteröinti. Toisaalta puhuminen on huomattavasti nopeampaa kuin komentojen kirjoittaminen. Muita rajoituksia puhekäyttöliittymissä on mm. puheen hetkellisyys, eli jo kuullun puheen palauttaminen takaisin mieleen ei ole yhtä helppoa kuin vain vilkaista jo luettua tekstiä näytöltä. [Barker and Weinschenk, 2000]

Suurimmat puhekäyttöliittymän avulla saavutettavat hyödyt ovatkin muiden aistien ja vuorovaikutuskanavien vapautuminen muuhun käyttöön ja puhekomennolla saavutettava nopeus järjestelmän käytössä. Parhaimmillaan puhekäyttöliittymä onkin yhdistettynä muihin vuorovaikutusmenetelmiin siten, että puheen käytöstä aiheutuvat haitat saadaan minimoitua: esimerkiksi nopeuttamalla toimintaa antamalla kaikki tilanteessa mahdollinen palaute visuaalisesti, ja antamalla puhuttua palautetta riittävän lyhyissä jaksoissa jotta käyttäjän on helpompi hahmottaa ja muistaa kuulemansa.

3.2. Puhekäyttöliittymän toiminta

Puhekäyttöliittymän toiminta muodostuu pääasiassa kahdesta osatekijästä: puhesynteesistä ja puheentunnistuksesta. Joissain tapauksissa, esimerkiksi usean käyttäjän järjestelmissä, myös puhujantunnistus on oleellista.

3.2.1. Keskustelu koneen kanssa

Keskustelu puhekäyttöliittymän ja käyttäjän välillä voidaan toteuttaa eri tavoilla, jotka kaikki asettavat erilaisia vaatimuksia sekä suunnittelulle että käyttäjän toiminnalle [Olsen et al., 2001].

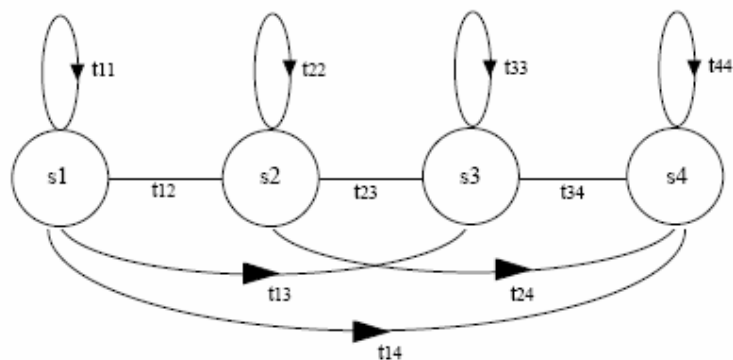
Järjestelmä voidaan suunnitella käyttämään lähes rajatonta luonnollista kieltä komentokeinona. Tällöin puheentunnistuksen suunnittelu ja toteutus on erittäin vaikeaa, mutta toisaalta käyttäjän ei tarvitse miettiä mitä komentoja sovellus hyväksyy. Toinen tapa toteuttaa keskustelu on tarjota käyttäjälle eri vaihtoehtoja dialogin edetessä. Tällaisessa tapauksessa käyttäjä vastaa järjestelmän kysymykseen tai antaa järjestelmälle jonkin komennon, jonka jälkeen järjestelmä mahdollisesti varmistaa edellisen, ja siirtyy sitten dialogissa eteenpäin. On myös mahdollista toteuttaa järjestelmä joka sallii vain tietyt komennot tietyissä tilanteissa. Tällaisen järjestelmän käyttö vaatii käyttäjältä opettelua ja toistuvaa käyttöä jotta sen hallinta muuttuisi sujuvaksi, etuna on puheentunnistuksen huomattava helpottuminen.

3.2.2. Puheentunnistus

Hyvin toimiva puheentunnistus on puhekäyttöliittymissä usein vaikein toteutettava osa-alue, pääasiassa puheen monimuotoisuuden vuoksi. Puhutun kielen lauserakenteet ja sanamuodot eroavat huomattavasti kirjoitetun kielen vastaavista, ja puheentunnistusjärjestelmän täytyy tämän takia osata tunnistaa erilaisia muotoja eri sanoista, mm. slangisanontoja, lyhenteitä ja mahdollisesti myös kielen murteista johtuvia eroja. Lisäksi ihminen voi tulkita saman äänten eri yhteyksissä eri äänneiksi kuin myös eri äänneetkin samaksi äänneeksi tilanteesta riippuen [McTear, 2002]. Ihmisen puhe on myös epätäydellistä, sillä varsinkin pitkiä lauseita korjataan puhuttaessa, lauseissa on usein mukana turhia täytesanoja tai niistä voi puuttua sanoja jotka puhuja kokee tarpeettomiksi. Puheentunnistus vaikeutuu huomattavasti kun järjestelmä joutuu käsittelemään tällaisia epätäydellisiä lauseita, ja siksi pitkäkestoisen luonnollisen puheen tunnistus on lähes mahdotonta [McTear, 2002].

Puheentunnistus toteutetaan analysoimalla foneemien, eli yksittäisten äänneiden, piirteitä. Foneemeista huomioidaan eri taajuudet ja näiden taajuuksien voimakkuus, eli kuinka paljon kutakin taajuutta on äänneessä mukana. Tärkeitä äänneiden tunnistuksen kannalta ovat voimakkaimpina esiintyvät taajuudet, eli formantit. Foneemien tunnistusta vaikeuttaa mahdollinen taustahäly tai epäselvä puhe. Näitä äänneitä yhdistelemällä voi järjestelmä tunnistaa puheesta sanoja jotka kuuluvat sen ennalta määrättyyn sanastoon ja kielioppiin. Puheentunnistuksessa käytetään apuna myös tilastollisia malleja joilla voidaan tutkia mitä järjestelmän sanaston kielimallia

lähimpänä jokin käyttäjän puhuma sanonta on. Tällainen tilastollinen mallintaminen perustuu yleensä *Markovin piilomalleihin* (Hidden Markov Models). Markovin piilomalli on äärellinen joukko tiloja joista jokaiseen liittyy tietty todennäköisyysjakauma. Jotta äänneiden yhdistelmä voitaisiin tunnistaa, ohjelma liikkuu eri tilojen välillä. Näiden tilojen välillä siirtyminen tapahtuu *siirtotodennäköisyyksien* perusteella, joiden avulla ilmaistaan kyseisen siirtymän todennäköisyys. Jokaisesta tilasta voidaan tuottaa tietty havainto siihen liittyvän todennäköisyysjakauman perusteella, ja vain tämä havainto on näkyvissä mallin ulkopuolelle, kun taas kyseinen tila pysyy ”piilotettuna”. Tämä perustuu siis todennäköisyyksien laskentaan analysoidun puheen perusteella. [McTear, 2002]



Kuva 2: yksinkertainen Markovin piilomalli jossa on näkyvissä eri tilat ja niiden välillä siirtyminen. Mallissa voidaan käydä sama tila läpi useasti (silmukat) tai jättää jokin tila kokonaan käymättä läpi. [McGlashan, 1995]

Jos järjestelmä on epävarma jostakin käyttäjän antamasta syötteestä, sen pitää kysyä käyttäjältä varmistus ennen toiminnon toteuttamista. Epävarmuus voi ilmetä esim. jos käyttäjä syöttää tuntemattoman sanan, eikä sen sovittaminen järjestelmän kielioppiin onnistu riittävällä todennäköisyydellä.

Suurimpia ongelmia puheentunnistuksessa ovat erilaiset puhesignaaleissa esiintyvät vaihtelut, kuten akustinen vaihtelu, lingvistinen vaihtelu sekä henkilöön liittyvä vaihtelu [Turunen, 1998]. Myös jo edellä mainittu puheen epätäydellisyys tuottaa tunnistukselle ongelmia, kuten myös puheen spontaanisuuskin, koska se sisällyttää puheeseen katkoja, korjauksia, epäröintiä ja muita ominaisuuksia, jotka eivät kuulu järjestelmän normaaliin kielioppiin tai sanastoon.

3.2.3. Puhesynteesi

Puhesynteesi tarkoittaa käytännössä tekstin muuttamista puheeksi. Tähän käytetään nykyään pääasiassa kolmea eri menetelmää: konkatenaatiosynteesiä, artikulaatiosynteesiä ja formanttisynteesiä [Lemmetty, 1999]. Konkatenaatiosynteesissä puhe muodostetaan yhdistelemällä elementtejä äänitietokannasta jotka ovat usein äänten kokoisia osia nauhoitettua ihmisen puhetta. Artikulaatiosynteesissä yritetään mallintaa ihmisen äänentuottoelimistöä mahdollisimman tarkasti, jotta saataisiin aikaan mahdollisimman luonnollista puhetta. Tämä on kuitenkin erittäin hankala toteuttaa nykyisillä menetelmillä. Formanttisynteesi sen sijaan on kokonaan koneella tuotettua puhetta, jossa pyritään jäljittelemään puheen akustisia signaaleja, ja tämä onkin yleisimmin käytetty tekniikka syntetisoitua puhetta tuottaessa. Formanttisynteesillä tuotettu puhe on kuitenkin kaikkein vähiten luonnollisen kuuloista, mutta sillä pystytään parhaiten mallintamaan puheessa esiintyviä prosodisia piirteitä, kuten sävelkorkeuden vaihtelua sekä puheen nousua ja laskua.

Konkatenaatiosynteesissä yhdistettävät äänteet ovat yleensä foneemeja tai difoneja (kahden foneemin yhdistelmiä), koska näitä on kielessä huomattavasti vähemmän kuin sanoja tai tavuja, ja näiden avulla voidaan myös hallita puheen prosodiaa, esimerkiksi puheen korkeuden vaihtelua, jonkin verran. Monissa sanoissa ilmenee myös kahden äänten yhdistelmiä, ja siksi luonnollista puhetta ei pystyttäisi tuottamaan konkatenaatiosynteesillä jos apuna ei käytettäisi foneemien lisäksi difoneja. [Lemmetty, 1999]

Syntetisoidun puheen laatua arvioidaan pääasiassa kahdella kriteerillä, ymmärrettävyydellä ja luonnollisuudella. Nykytekniikalla tuotettu puhe on usein riittävän ymmärrettävää, mutta ei aina kovin luonnollista. Luonnollisuuden puute onkin usein yksi puhekäyttöliittymien käytön suurimpia haittapuolia, sillä nykyään juuri luonnollisuus on yhtenä pääasiallisena tavoitteena puhekäyttöliittymäsuunnittelussa [Turunen, 1998].

Hyvin suunniteltu puhesyntetisaattori osaa myös käsitellä erilaiset lyhenteet, erikoismerkit sekä erityyppiset luvut ja numerot, kuten päivämäärät, oikein. Tällaiset tekstin ominaisuudet pitää esittää käyttäjälle ymmärrettävässä ja arkipäiväisessä muodossa. [Lemmetty, 1999]

3.2.4. Puhujantunnistus

Puhujantunnistus toteutetaan analysoimalla puhujan äänen piirteitä, jotka ovat kaikilla yksilöllisiä. Toimiakseen puhujantunnistin täytyy opettaa tunnistamaan tietty ihminen käyttämällä opetuksessa apuna näytteitä kyseisen ihmisen puheesta. Jos halutaan päästä parhaaseen mahdolliseen tulokseen, täytyy nämä näytteet nauhoittaa samassa ympäristössä jossa varsinainen tunnistuskin

aiotaan suorittaa, jotta mm. taustahälyn vaikutukset voidaan ennakoida ja välttää.

3.3. Vaatimuksia puhekäyttöliittymälle ja sen suunnittelulle

Tärkeimpiä kohtia puhekäyttöliittymän suunnittelussa on kehittää sovellukselle riittävän laajat sanasto ja kielioppi jotta ohjelma suoriutuisi sille annetuista komennoista. Kuten aiemmin jo todettiin, sanasto ei kuitenkaan saa olla liian laaja koska puheentunnistuksen taso laskisi sen takia. Sanastoon on kuitenkin hyvä liittää kaikkien tärkeimpien sanontojen ja niiden synonyymien lisäksi tarpeeksi muitakin termejä, joita käyttäjä saattaa tarvita voidakseen tehokkaasti ohjata sovellusta. Eri tilanteissa voidaan myös käyttää eri kieliopeja, jotta tunnistustehokkuus säilyy ja järjestelmästä saadaan silti riittävän monipuolinen. [Barker and Weinschenk, 2000] Lisäksi järjestelmän on osattava käsitellä virheet oikein, ja tarvittaessa varmistaa käyttäjältä mitä tämä tarkoitti. Varmistaminen on kuitenkin hyvä tehdä vain silloin, kun järjestelmä on epävarma käyttäjän antamasta syötteestä, tai jos kyseessä on erittäin kriittinen tai oleellinen toiminto. Jatkuva kysely todennäköisesti hermostuttaisi käyttäjän ennen pitkää.

Keskustelu järjestelmän kanssa voidaan toteuttaa siten, että aloite on käyttäjällä tai aloite on järjestelmällä [McTear, 2002]. Käytettäessä järjestelmää jossa aloite on käyttäjällä, täytyy käyttäjän tietää mitä järjestelmällä voi tehdä. Käyttäjän ei välttämättä tarvitse tietää järjestelmän komentoja tarkasti, mutta hänellä on oltava hyvä kuva siitä mitä hän haluaa ja voi tehdä järjestelmällä. Jos aloite on järjestelmällä, täytyy käyttäjän ainoastaan vastata järjestelmän esittämiin kysymyksiin. Tällainen tapa saattaa toisinaan olla hidas, mutta sopii hyvin sovelluksiin joissa tapahtumat etenevät helposti vaiheittain. [McTear, 2002; Barker and Weinschenk, 2000] Toisinaan näitä kahta keskustelutapaa voidaan myös yhdistellä, esimerkiksi sovelluksessa jossa käyttäjällä on aloite voi järjestelmä tarjota käyttäjälle apua jos tämä on hiljaa pitkään.

Liian pitkien lauseiden sanomista käyttäjälle pitää myös välttää, sillä ihminen ei kykene muistamaan kovin pitkiä puhuttuja lauseita. Varsinkin jos käyttäjälle annetaan mahdollisuus valita useista eri toiminnoista, on järjestelmän hyväksyttävä valinta heti kun kyseinen toiminto on käyttäjälle sanottu, eikä vasta valintalistan loputtua. Puhesovellusten haittana olevaa käyttäjälle aiheutuvaa odotusta on myös pyrittävä minimoimaan mm. tarjoamalla edistyneimmille käyttäjille lyhyempiä kehoitteita, ja lyhentämällä usein toistuvia lauseita käyttäjän tottuessa niihin [McTear, 2002].

4. Virtuaalitodellisuusympäristössä toimivan puhekäyttöliittymän suunnittelu

Tässä luvussa tarkastellaan virtuaalitodellisuuden puhekäyttöliittymille asettamia vaatimuksia sekä myös mitä erilaisia käyttömahdollisuuksia puheella on virtuaalitodellisuusympäristöissä.

4.1. Puheen käyttö vuorovaikutuskeinona virtuaalitodellisuudessa

Tavanomaisin hallintalaite virtuaalitodellisuusympäristössä on jokin virtuaalitodellisuudelle ominainen laite, kuten datahanska tai jokin muu kuuden vapausasteen ohjainlaite. Jos virtuaalitodellisuussovellukseen liitetään puheohjaus vuorovaikutusmenetelmäksi, niin käyttäjät voivat vaikuttaa sovelluksen toimintaan entistä monipuolisemmin ja tehokkaammin. Käyttäjät voivat esimerkiksi suorittaa kaksi eri tehtävää samanaikaisesti, kuten virtuaalitodellisuudessa liikkumisen ja jonkin toiminnon suorittamisen. Virtuaalitodellisuudessa liikkuminen tapahtuu yleensä saman ohjainlaitteen avulla, jolla muitakin eri toimintoja ohjataan, eikä tämän laitteen käyttö useampaan kuin yhteen toimintoon kerrallaan ole kovin helppoa. Lisäämällä ympäristöön puhekäyttöliittymän voi käyttäjä toteuttaa eri toimintoja puhumalla samalla kun hän navigoi ympäristössä. [Axling and McGlashan, 1996]

Puhekomennoilla voi myös nopeuttaa toimintaa virtuaalitodellisuudessa huomattavasti, sillä useimmat komennot virtuaalitodellisuusjärjestelmissä löytyvät valikoista jotka saa esiin ympäristön ohjauslaitteella. Valikoiden selaaminen kolmiulotteisessa ympäristössä voi olla vaikeaa, varsinkin jos selattavia vaihtoehtoja on useita. Tämä johtuu joidenkin hallintalaitteiden kömpelyydestä ja käytön epäselvyydestä, laitteissa on usein monta eri painiketta joiden toiminnasta on vaikea ottaa selvää ennen kuin vasta usean käyttökerran jälkeen, lisäksi usean vapausasteen omaavia hallintalaitteita ei ole helppo käyttää kaksiulotteisissa valikoissa. Jos käyttäjä tietää jo jonkin komennon, hän voi sanoa sen järjestelmälle avaamatta yhtäkään valikkoa, ja käyttäjä voi myös nopeuttaa toimintaa lukemalla komennon jo avatusta valikosta ääneen sen sijaan että siirtäisi hallintalaitteen osoittimen komennon päälle valitakseen sen. [Grasso et al., 1998]

Virtuaalitodellisuudessa eri kohteiden käsittely puhekomennoilla on myös paljon nopeampaa kuin niiden muokkaaminen erillisellä hallintalaitteella [McGlashan, 1995]. Käyttäjä voi valita kohteen, jos kohteita on useita, kuvailemalla sen jotakin ominaisuutta (esimerkiksi väriä: "sininen laatikko") ja antamalla samalla haluamansa komennon (esimerkiksi "siirrä sininen laatikko vasemmalle pöydälle"). Näin hänen ei tarvitse ohjata kohdetta hallintalaitteella eikä edes liikkua eri paikkaan ympäristössä. Myös kohteen ominaisuuksien,

kuten värin, koon ja jossain määrin myös muodon muokkaaminen puheohjauksella on helpompaa ja nopeampaa kuin hallintalaitteella.

Virtuaalitodellisuusympäristössä voi palautteen puhekomennoista myös antaa nopeasti visuaalisella tavalla, eikä käyttäjän tarvitse odottaa järjestelmän puhuttua vahvistusta tapahtumalle. Tietysti puhesynteesin yhdistäminen visuaaliseen palautteeseen tai pelkän puhepalautteen antaminen voi myös joissakin tilanteissa olla hyödyllistä. Esimerkiksi silloin, jos käyttäjä on antanut puhekomennon joka koskee jotakin sillä hetkellä näkymätöntä kohdetta, on järjestelmän luonnollisesti annettava komennon suorittamisesta puhuttu vahvistus. Virtuaalitodellisuusympäristöissä järjestelmän puhepalautteen aikana voi kuitenkin suorittaa muitakin tehtäviä muilla hallintalaitteilla, joten käyttäjä ei tunne odottavansa niin paljon kuin pelkkää puhetta hyödyntävän käyttöliittymän kanssa.

4.2. Puheentunnistuksen toteuttaminen virtuaalitodellisuudessa toimivaksi

Puheentunnistuksen vaikeutena virtuaalitodellisuusympäristöissä tavallisten ongelmien lisäksi on virtuaalitodellisuudessa oleva tila ja kohteiden määrä: järjestelmä ei välttämättä tiedä mitä kohdetta käyttäjä tarkoittaa tai mitä käyttäjä oikeastaan haluaa tehdä, vaikka se ymmärtäisikin komennon muuten oikein. [McGlashan, 1995] Jos ympäristössä on vain vähän kohteita, niin tämä ei ole erityinen ongelma. Mutta jos mahdollisia kohteita on paljon ja jos ne ovat vielä lähestulkoon identtisiä, niin järjestelmä ei mahdollisesti onnistu erottamaan käyttäjän tarkoittamaa kohdetta muiden joukosta.

Tällaiset tilanteet voidaan kuitenkin välttää käyttämällä useaa eri vuorovaikutusmenetelmää. Käyttäjä voi esimerkiksi valita haluamansa kohteen hallintalaitteella, ja sanoa komennon vasta sen jälkeen, tai siirtää haluamansa kohteen erilleen muista jolloin järjestelmä ei enää sekoita sitä niin helposti muihin vastaaviin kohteisiin.

Toinen vaihtoehto kohteen valitsemisen sijaan on pelkkä kohteen osoittaminen järjestelmälle käyttäjän eleellä. Tästä kerrotaan lisää kappaleessa 4.3.

Puheentunnistusjärjestelmän on sisällettävä myös riittävän laaja sanasto jotta se pystyisi toimimaan tehokkaasti virtuaalitodellisuusympäristössä, jossa on normaalien komentojen lisäksi huomioitava myös ympäröivä tila, suunnat ja ulottuvuudet.

4.3. Kehonliikkeiden tunnistaminen ja yhdistäminen puhekomentoihin

Tärkein syy yhdistää eleet ja puhekomennot toisiinsa ohjauskeinona virtuaalitodellisuusympäristöissä on tämän aikaansaama käytännöllisyys, käyttäjä voi ilmaista itseään helposti ja luonnollisesti hyödyntämällä

luonnollista puhetta sekä eleitä, asioita jotka ovat tärkeimmät kommunikaatiokeinomme normaalissa vuorovaikutuksessa [Billinghurst, 1998]. Puhe ja eleet täydentävät tehokkaasti toisiaan, koska niissä yhdistyy puheen kuvaileva ominaisuus sekä eleiden tapa vaikuttaa suoraan kohteeseen [Cohen, 1992]. Puheella voi myös paikata hyvin ele- tai kosketuskäyttöliittymässä olevia aukkoja, sillä puheella voi käsitellä myös piilossa olevia kohteita tai vaihtoehtoja toisin kuin pelkkiä eleitä tai kosketusta hyödyntämällä. Yhdistämällä nämä kaksi vuorovaikutustapaa voidaan puheohjausta myös nopeuttaa, koska käyttäjä voi korvata osan komentoon tarvittavista sanoista eleillä tai kosketuksella. Suurin osa käyttäjistä käyttääkin mieluummin näitä molempia vuorovaikutuskeinoja yhdistettynä jos se vain on mahdollista [Billinghurst, 1998].

Eleiden tunnistus virtuaalitodellisuusympäristössä tapahtuu kappaleessa 2.2.2 kuvatulla käyttäjän seurannalla. Käyttäjällä voi olla ohjainvälineenä esimerkiksi datahanska, jonka liikettä ja paikkaa järjestelmä seuraa. Käyttäjän antaessa esimerkiksi seuraavanlaisen komennon: "poista tuo pallo", järjestelmä tunnistaa komennon "poista" sekä kohteen "pallo", mutta ei pysty pelkän puhekomennon perusteella päättämään mikä pallo on kyseessä, olettaen että ympäristössä on vastaavia kohteita useampikin. Tällöin järjestelmän täytyy ottaa huomioon myös pronomini "tuo", ja yrittää sen perusteella arvioida mitä käyttäjä tarkoittaa. Järjestelmä voi nyt laskea käyttäjän hallintalaitteen osoittaman suunnan ja laskea onko siellä käyttäjän mainitsemaa kohdetta. Jos kohde löytyi, komento voidaan toteuttaa ilman mitään tarkentavaa käskyä. [Billinghurst, 1998]

4.4. Puheohjattavat agentit virtuaalitodellisuussovelluksissa

Tässä tarkastellaan mahdollisuutta käyttää puheohjattuja agenttiohjelmaa erilaisiin tehtäviin virtuaalitodellisuussovelluksissa. Agenttiohjelma on ohjelma joka suorittaa käyttäjän puolesta erilaisia toimintoja, joita toistetaan usein tai jotka suoritetaan yleensä tiettyjen vaiheiden mukaan. Agentteja voi myös käyttää etsimään tietoa tai apua käyttäjälle, tai suorittamaan muita yksinkertaisia käyttäjän haluamia toimintoja.

Axling ja McGlashan [1996] esittelevät sovelluksen, jossa käyttäjä pystyy vaikuttamaan virtuaalitodellisuusympäristöön puheohjattavien agenttien avulla. Tässä ympäristössä agenttihahmo seuraa käyttäjää odottaen jotakin komentoa, ja komennon saadessaan vahvistaa sen järjestelmän toteuttaessa annetun komennon. Käyttäjän saatavilla voi olla useampikin agentti, jotka kaikki suorittavat eri toimenpiteitä, ja joilla näin ollen on jokaisella oma sanastonsa puheentunnistuksen helpottamiseksi.

Kyseisessä sovelluksessa käyttäjä puhuu ainoastaan agenteille halutessaan suorittaa jonkin tehtävän, ja muulloin sovelluksessa ei hyödynnetä puhekäyttöliittymää. Mielestäni järjestelmä, jossa käyttäjä voi vaikuttaa suoraan sovellukseen, on käyttäjälle mielekkäämpi ja myös ehkä luonnollisempi käyttää kuin järjestelmä, jossa kaikki puheohjaus tapahtuu agenttien välityksellä. Puheohjatun agentin tai agenttien lisääminen tällaiseen järjestelmään olisi myös hyvä ajatus. Käyttäjä voisi käyttää tällaisia agenteja esimerkiksi tiedonhakuun tai automatisoimaan tehtäviä joihin käyttäjältä kuluisi itseltään useita komentoja ja paljon aikaa.

5. Lopuksi

Tässä tutkimuksessa käsittelin puhekäyttöliittymien soveltamista erityyppisiin käyttötarkoituksiin virtuaalitodellisuusympäristöissä keskittyen mm. tällaisten sovellusten käyttömahdollisuuksiin ja helppokäyttöisyyteen. Tarkastelun kohteena oli myös puhekäyttöliittymän toteutuksen kannalta oleellisia asioita sekä käytettävyyden näkökohtia ja virtuaalitodellisuustekniikoiden perustekijöitä.

Puhutun kielen käyttö vuorovaikutuskeinona virtuaalitodellisuusjärjestelmissä on tehokas ja käyttäjälle luonnollinen tapa hallita ympäristöään. Pelkkä puheohjaus ei kuitenkaan riitä virtuaalitodellisuussovelluksien käyttöön, vaan mukana pitää olla myös muitakin vuorovaikutustapoja, etenkin liikkumista ja kohteiden tarkkaa siirtelyä ja muokkaamista varten. Parhaiten puhekäyttöliittymä sopiikin virtuaalitodellisuusympäristöön toimintaa tehostavana ja nopeuttavana välineenä, jota käyttäjä voi käyttää yhdessä muiden järjestelmän sallimien vuorovaikutuskeinojen ohella silloin kun haluaa.

Virtuaalitodellisuusjärjestelmien yleistyessä niistä todennäköisesti tullaan kehittämään käyttäjille mahdollisimman luonnollisia ympäristöjä, jolloin puhekäyttöliittymät tulevat olemaan olennainen osa näitä järjestelmiä.

Viiteluettelo

- [Axling and McGlashan, 1996] Tomas Axling and Scott McGlashan, Talking to agents in virtual worlds, Swedish Institute of Computer Science, Kista, Sweden, 1996. <http://www.sics.se/~scott/papers/ukvrsiq96.ps>
- [Barker and Weinschenk, 2000] Dean T. Barker and Susan Weinschenk, *Designing effective speech interfaces*. Wiley computer publishing, 2000.
- [Billinghurst, 1998] Mark Billinghurst, Put that where? Voice and gesture at the graphics interface, *ACM SIGGRAPH Computer graphics*, **32(4)** 1998, 60-63.
- [Cohen, 1992] Philip R. Cohen, The role of natural language in a multimodal interface, *Proceedings of the 5th annual ACM symposium on user interface software and technology*, ACM Press, 1992, 143-149.

- [Feiner and MacIntyre, 1996] Steven Feiner and Blair MacIntyre, Future multimedia user interfaces, *Multimedia Systems*, 4(5), 1996, 250-268.
- [Grasso et al., 1998] Michael A. Grasso, David S. Ebert and Timothy W. Finin, The integrality of speech in multimodal interfaces, *ACM transactions on computer-human interaction (TOCHI)*, 5(4) 1998, 303-325.
- [Kalawsky, 1993] Roy S. Kalawsky, *Science of virtual reality and virtual environments, a technical, scientific, and engineering reference on virtual environments*. Addison-Wesley, 1993.
- [Lemmetty, 1999] Sami Lemmetty, Review of speech synthesis technology, Helsinki university of technology, department of electrical and communications engineering, Master's thesis, 1999. Also available at <http://www.acoustics.hut.fi/~slemmett/dippa/thesis.pdf>
- [McGlashan, 1995] Scott McGlashan, Speech interfaces to virtual reality, Swedish Institute of Computer Science, Kista, Sweden, 1995. Also available at <http://www.sics.se/~scott/papers/specom96.ps>
- [McTear, 2002] Micheal F. McTear, Spoken dialogue technology: enabling the conversational user interface, ACM Computing Surveys (CSUR) march 2002, ACM Press, New York, USA, 90-169, 2002.
- [Lappeenrannan teknillisen korkeakoulun verkkosivut, 2003] Tietotekniikan seminaari 3D-tietokonegrafiikan tulevaisuudennäkymät; virtuaali-todellisuus <http://www-user.lut.fi/~leve/1591.htm> (5.12.2003)
- [Olsen et al., 2001] Dan Olsen, Ronald Rosenfeld, Alex Rudnicky, Universal speech interfaces, *Interactions*, 8(6) 2001, 34-44.
- [Turunen, 1998] Markku Turunen, Puheohjaus 3D-käyttöliittymissä, Tampereen yliopisto, Tietojenkäsittelytieteen laitos, Pro Gradu-tutkielma, 1998. Saatavana elektronisessa muodossa <http://www.cs.uta.fi/~mturunen/Gradu/>
- [University of Michigan virtual reality lab, 2003] University of Michigan virtual reality lab web page, Virtual Reality: A Short Introduction, <http://www-vrl.umich.edu/intro/> (7.12.2003)
- [Vince, 1995] John Vince, *Virtual reality systems*, Addison-Wesley, 1995.

Muutosvastarinta ilmiönä

Anne Jokinen

Tiivistelmä.

Muutosvastarinta on ilmiö, jota ilmenee organisaatioiden muutosprosessien yhteydessä. Se syntyy pääosin työntekijöiden emotionaalisista lähtökohdista, jotka johtuvat siitä, että ei ymmärretä muutoksen syitä tai koetaan muutos uhkana omalle asemalleen. Tätä voidaan pyrkiä ennaltaehkäisemään ottamalla työntekijät osaksi uuden tietojärjestelmän suunnittelua ja näinollen antamalla heille mahdollisuuden vaikuttaa tulevien muutosten luonteeseen ja paremmin ymmärtämään sen tarpeellisuutta. Avoin tiedottaminen ja yhtenäisen organisaatiokulttuurin rakentaminen ovat tärkeimmät työkalut niin muutosvastarinnan ennaltaehkäisyssä kuin siinäkin tilanteessa kun ilmiötä jo esiintyy.

Avainsanat ja -sanonnat: Muutosvastarinta, Organisaation muutos, Organisaatiokulttuuri

CR-luokat: 61.2

1. Johdanto

Hieman yksinkertaistaen voidaan sanoa, että organisaatiot koostuvat ihmisistä, tehtävistä, joita tarvitsee suorittaa organisaation olemassaolon (toiminta-ajatuksen) toteutumiseksi sekä näiden tehtävien suorittamiselle tarkoitetuista (teknisistä) apuvälineistä. Kun nämä osat kohtaavat toisensa jokapäiväisessä toiminnassa, muodostuu usein rutiininomaisia tapoja tai käytäntöjä, joiden mukaan toimitaan. Rutiinit ovat ihmisten tapa suorittaa usein toistuvia tehtäviä kuormittamatta liikaa omaa kapasiteettiaan. Kuitenkin, joskus näissä organisaation osissa saattaa tapahtua muutoksia: henkilöstön koostumus muuttuu, tehtävien laatu muuttuu tai jonkin tehtävän suorittamista varten on kehitetty uudenlaisia apuvälineitä, joita otetaan käyttöön tehtävän entistä tehokkaampaa suorittamista varten. Näissä tilanteissa organisaatiossa toimivien ihmisten keskuudessa saattaa syntyä muutosvastarintaa, joka usein vaikeuttaa uuteen tilanteeseen totuttautumista ja uudenlaisten rutiinien kehittymistä. Ihmiset eivät välttämättä ole valmiita luopumaan totutuista käytännöistä ja haluavat pitää niistä kiinni, jolloin uutta tapaa hyljeksitään ja tulos saattaa tämän myötä olla organisaation tuloksen kannalta varsin epätehokas.

Muutosvastarinta on jokaisessa organisaatiossa tapahtuvissa muutoksissa mukana ja erityisesti se tuntuu korostuvan uuden tekniikan käyttöönotossa. Siksi on tärkeää tietää mitä muutosvastarinta kokonaisuudessaan merkitsee, mistä se johtuu ja miten siihen voidaan vaikuttaa. Sen merkitys heijastuu koko yritykseen, sillä se vaikuttaa välillisesti myös yrityskulttuuriin ja työilmapiiriin, jossa työntekijöiden tulisi kokea viihtyvää ja työtehtävien tekeminen olisi paitsi miellyttävää, myös sujuvaa.

Ilmiön yleisyydestä johtuen siihen tulisi kiinnittää riittävästi huomiota siihen, että organisaatiossa tapahtuvien muutosten avulla pystyttäisiin lisäämään organisaation toimintatehokkuutta ja kyettäisiin säilyttämään tai kehittämään mahdollisimman toimiva työilmapiiri. Tämä lienee tärkeää jokaiselle organisaatiolle, joka tavoittelee hyvää menestystä toiminnalleen. Edellä mainituista syistä johtuen aihepiiriä ja aihetta voidaan pitää merkittävänä tutkimuskohteena, jotta siihen osattaisiin paremmin varautua ja kehittää organisaation keinoja toimia sekä ennaltaehkäisevästi että niissä tilanteissa, joissa muutosvastarintaa jo esiintyy.

Tässä tutkimuksessa tarkastelen luvussa 2 mitä varhaisemmissa tutkimuksissa on määritetty muutosvastarinnan keskeisiksi piirteiksi ja millaiset tekijät vaikuttavat muutosvastarinnan syntyyn. Luku 3 keskittyy käsittelemään millaisia keinoja on kehitetty torjumaan muutosvastarintaa ja minimoimaan sen negatiivisia vaikutuksia ennaltaehkäisevästi sekä 4 luku millaisia keinoja on kehitetty muutosvastarinnan aiheuttamien ongelmien ratkaisemiseksi.

2. Muutosvastarinnan syntytekijät ja keskeiset piirteet

Tässä luvussa tarkastellaan muutosvastarinnan syntytekijöitä ja mitkä ovat kyseisen ilmiön tunnusomaisia piirteitä. Pyrin tuomaan esille ensisijaisesti teknologisten muutosten aiheuttaman vastarinnan näkökulman sen historiallisuudessaan.

2.1. Tietotekniikkapohjaisen muutosvastarinnan tutkimuksen lähtökohtia

Muutosvastarinnan tutkimukseen liittyy usein teknologiset muutokset, ja ilmiötä on paljon tutkittu nimenomaan teknologisten uudistusten näkökulmasta. Sonquist [1976] tutki asiaa yleisestä organisaationaalista muutoksesta ja organisatorisesta suunnittelusta erityisesti tietojenkäsittelyyn liittyvien asioiden ja ongelmien näkökulmasta käyttäen tutkimuskohteena yliopistojen tiedekuntia. Hän esittää artikkelissaan muutosvastarinnan syntymiseen vaikuttaviksi tekijöiksi organisaation monimutkaisuuden, vallan, keskittymisen, organisaation muodollisuuden ja kerrostumisen asteen.

Organisaation monimutkaisuudella tarkoitetaan sitä, että se koostuu monista eri toimista, ja kun toimijoita on monia, muodostuu varsin monimutkainen suhdeverkosto. Tässä verkostossa voidaan toimia keskitetysti tai hajautetummin, samoin valta jakautuu eri toimijoiden kesken. Muodollisuus puolestaan tarkoittaa sitä, miten rajattu kukin toimi on: Mitä rajatumpia, sen muodollisempi organisaatio. Muodolliset organisaatiot usein kapinoivat herkemmin muutosta vastaan, sillä tehtävät on totuttu suorittamaan tietyn rutiininomaisen kaavan mukaan ja siitä ei välttämättä haluta luopua. Kerrostumisasteet viittaavat siihen, mikä on esimerkiksi ikä- ja asemajakauma organisaatiossa. Ne voivat olla selkeitä, mikä johtaa eri kerrostumien toiminnan omaksumisen palkitsemisiin herkemmin, kun taas niiden ollessa vähemmän selkeitä, tällaista ei ole määriteltynä.

Sonquist toteaa myös, että tärkeä tekijä muutoksen hyväksymisessä on, että sitä pidetään toteuttamiskelpoisena ratkaisuna jollekin ongelmalle. Uuden tietotekniikan kanssa on aina hyvä kysyä onko se todella ratkaisu olemassaolevalle ongelmalle, vai onko se jotakin, joka halutaan ottaa käyttöön tekniikan itsensä takia.

Sosiaalisen inertiaa eli alkuvastustusta tutkinut Keen [1981] määrittelee sille neljä syytä: 1) Informaatio on vain pieni komponentti organisaation päätöksentekoprosessissa, 2) Ihmisten informaationkäsittely on kokemuksellista ja perustuu yksinkertaistamiseen, 3) Organisaatiot ovat monimutkaisia ja muutos on kasvavaa ja evolutiivista; suuria askelia vältetään ja jopa vastustetaan ja 4) Data ei ole vain älyllinen tuote vaan poliittinen resurssi, jonka uudelleenkäyttö informaatiotieteiden kautta vaikuttaa tiettyjen ryhmien intresseihin. Tämä osittain tukee Sonquistin aiemmin esittämiä näkemyksiä ja tuo siihen hiukan uusia piirteitä.

Keen argumentoi, että vaikka tietokoneasiantuntijat usein pitävät tietojärjestelmiä keskeisessä asemassa päätöksenteossa, todellisuudessa näin ei ole. Lukuisat tutkimukset osoittavat, että ihmiset välttävät monimutkaisia tiedonkeruumenetelmiä ja pyrkivät käyttämään mahdollisimman yksinkertaistettua dataa, vaikka se joskus olisi jopa asian kannalta irrelevanttia. Organisaation monimutkaisuus johtaa siihen, että suuren muutoksen läpi saaminen on hyvin harvinaista. Yksi muutosvastarinnan keskeisimmistä syntytekijöistä on omasta valta-asemastaan kiinni pitäminen. Uuden tietojärjestelmän koetaan monesti olevan uhkana omalle valtapotentiaalille. Omaa tietämystä pidetään siis poliittisena resurssina, jonka avulla omasta vaikutusvallasta organisaatiossa pidetään huolta.

2.2. Oliosuuntautuneen ja graafisen käyttöliittymän aikakauden muutosvastarinnan riskitekijöitä

Ihmisyhteisön, joka voi olla esimerkiksi organisaatio, mielipiteiden ja toimintatapojen muutoksen hitaudesta eli psykologisesta inertiaasta puhuva Hartikka [1990] tukee aiemmin esitetyn sosiaalisen inertian ajatusta. Voidaan sanoa hänen tarkoittavan samaa ilmiötä sanoessaan psykologisen inertian säätelevän ja jarruttavan muutosvastarinnan voimalla lähes kaiken uuden käyttöönottoa piirissään riippumatta siitä, onko uudistuksesta hyötyä kyseiselle yhteisölle vai ei. Hän määrittää ilmiön keskeisiksi piirteiksi passiivisuuden, vastustuksen ja asiantuntemattomuuden, jotka jokainen omassa vaiheessaan jarruttavat uuden käyttöönottoa. Ihmiset ovat usein passiivisia uuden vastaanotolle, "ilmankin on pärjätty ihan hyvin" on yleinen asenne. Vastustus ilmenee erilaisten negatiivisten argumenttien etsintänä, yhteisön ihmiset pyrkivät usein perustelemaan miksi uusi innovaatio on tarpeeton ennemmin kuin miksi siitä voisi olla jotakin hyötyä. Kolmanneksi, mikäli päätetään ottaa uusi teknologia käyttöön, sen käyttö yhteisön keskuudessa saattaa johtaa epäpätevään tulokseen, sillä he ovat asiantuntemattomia ja luonnollisesti tulkitsevat sitä oman kokemusmaailmansa pohjalta. Niinpä jotkut projektit kaatuvat siihen, että niiden käyttö ei ole oikein hallittua. Usein uuden tekniikan tuominen organisaatioon merkitsee sitoutumista pitkäjänteiseen kehitysprosessiin tai projektin epäonnistumista, mikäli sitoutumista ei tapahdu.

Dray [1994] puolestaan tuo esille 1990-luvun puoliväliä hallinneen teknologisten odotusten muospaineiden ajalta eräitä tärkeitä huomioita mahdollisista muutosta estävistä tekijöistä. Osa niistä on ilmiselviä, osa vaikeammin havaittavissa. Vaikeammin havaittavissa olevien riskitekijöiden huomaamatta jääminen tarkoittaa usein sitä, että projekti ei varsinaisessa mielessä epäonnistu, se vain ei täytä niitä tuottavuustavoitteita, jotka sille asetettiin. Tällöin voi olla vaikeaa löytää erityistä syytä epäonnistumiselle.

Dray on määrittänyt yleisiksi kompastuskiviksi seuraavat: liiketoiminnan ja teknologian käsittely erillisinä kokonaisuuksina, liian vähäisen ajan tai muiden resurssien käytön organisationaalisessa suunnittelussa, teknisen muutoksen käyttäminen organisationaalisesta muutoksesta aikaansaamiseksi, keskittyminen toiminnallisuuteen ennemmin kuin käytettävyyteen, teknosentristen menetelmien käyttäminen käyttäjäkeskeisten menetelmien sijaan, projektin tilan seurannan puutteellisuus tai sen ymmärtämättömyys, muutokseen kohdistuvien emotionaalisten reaktioiden käsittelemättä jättäminen sekä taitojen kehittämisen tarpeen huomiotta jättäminen kehitystiimin ja/tai käyttäjien keskuudessa. Tämä tukee varhaisempia tutkimuksia [Sonquist 1976;

Keen 1981], mm. emotionaalisten reaktioiden merkitys ja huomion kiinnittäminen käyttäjien kykyyn omaksua uusi järjestelmä toistuvat tutkimuksesta toiseen.

Siirryttäessä 2000-luvulle muutosvastarinnan keskeisiksi syiksi ovat vakiintuneet työntekijöiden uudenlaisten vastuiden vastaanottohaluttomuus, prosessiajatuksen vierastaminen, haluttomuus riskinottoon ja siksi pitäytymien vanhassa ja turvalliseksi koetussa tavassa, haluttomuus itsensä jatkuvaan kouluttamiseen, pelko lisääntyvästä työmäärästä sekä omassa asemassa tai vallassa tapahtuvista muutoksista sekä keskitason johdon haluttomuus tukea muutosta. Usein perusteet ovat varsin emotionaalisia, vastustetaan enemmän periaatteesta kuin varsinaisesti perustelluista syistä. [Qualitas Fennica Oy, 2000]

Turun yliopiston täydennyskoulutuskeskuksen [2001] määritelmässä puolestaan sanotaan muutosvastarinnan tavoittelevan tasapainon säilymistä ja henkisen tilan hallintaa. Siitä johtuen ihminen säännöstelee muuttumistaan ja arvioi uutta ainesta ja tilanteita tarkasti ennen kuin lähtee niihin mukaan. Joillekin ihmisille muuttuminen on erityisen hankalaa tai sitoutuminen uuteen on hidasta, sillä ihminen kaipaa tulevaisuudennäkymät muutaman vuoden päähän ja ennakoi tulevaa, muutos taas järkyttää ennakkoinnin perusteita. Muutosvastarinta on tässä yhteydessä jaettu kahteen vaiheeseen, uhankokemisvaiheeseen ja varsinaiseen vastustusvaiheeseen. Ensimmäisessä esiintyy myös välitön lamaantumisvaihe ja toisinaan myös uhkaavan tilanteen pitkittymisvaihe. Uhankokemisvaiheessa työntekijä kohtaa psyykkisen lamaantumisen, jonka aikana hän prosessoi mieleen vyörynyttä aineistoa. Tätä säännöstellään salpaamalla ajatuksia ja osittain kieltämällä todellisuutta. Näkemysala kaventuu ja uusajattelu jähmettyy, jotta uuden tiedon ja tunnekokemusten käsittelylle on tilaa. Kun muutoksen tarkka sisältö ja merkitys selviää työntekijälle, hän alkaa valita toimintavaihtoehtoja. Uhan kokemus muuttuu aktiiviseksi toiminnaksi ja tiedon myötä uteliaisuudeksi sitä ilmiötä kohtaan, joka aiheutti muutoksen.

3. Muutosvastarinnan riskien minimoiminen ja torjuminen ennaltaehkäisevästi

Hodgson ja Aiken [1998] tutkivat informaatiotietojärjestelmien (IS) ja ennen kaikkea niiden suunnittelusta ja toteutuksesta vastaavien henkilöiden roolia muutoksen kohtaamisessa. Muutoksen hallintaan vaikuttavat paitsi itse konkreettiset muutokset järjestelmässä, myös yleinen asenne muutoksia kohtaan ja erityisesti asenne kyseistä muutosta kohtaan. He toteavat aikaisempien tutkimusten luovan käsitystä siitä, että muutokseen vaikuttaa olennaisesti suhde teknisen

systemin ja ihmisten muodostaman systemin (yksilöiden ja ryhmän käyttäytymistottumukset) välillä.

Tästä johtuen Hodgson ja Aiken uskovat uuden IS:n toteutuksesta vastaavien henkilöiden roolin muutosagenttina olevan varsin merkittävä, sillä heillä on mahdollisuus vaikuttaa siihen, millaisia asenteita muutosta kohtaan organisaation työntekijöille muodostuu. Heidän tulisikin siis ajatella itseään organisationaalisen muutoksen agentteina, ei pelkästään informaation systemin muutosta edistävänä agenttina.

Hodgsonia ja Aikenia aiemmin mm. Salmela [1993] on tutkinut informaation systemien suunnittelun roolia organisaation muutoksissa. Hänen mukaansa lukuisat eri teoriat tukevat käsitystä siitä, että muutokset informaation systemissä tarkoittavat muutosta myös organisaatioiden muilla tasoilla. Tätä muutosta tulee siis tarkastella organisaation kokonaisvaltaisen muutoksen tavoitteiden näkökulmasta, ei ainoastaan IS -näkökulmasta.

Hän esittelee artikkelissaan muutossuuntautuneen kehittämisen (change oriented development, Chord), jota hyödynnetään käyttäjien vaatimusten analysoimisvaiheessa. Tällöin käyttäjät ja käyttäjien johto otetaan osaksi koko systemin suunnittelua, jolloin voidaan minimoida riskiä, että käyttäjät kapoinoisivat uudistuksen käyttöönottoa vastaan kun he voivat itse olla mukana määrittelemässä, mitkä muutokset nähdään väistämättöminä ja haluttavina. Tätä tietoa käytetään myöhemmin analysoimaan ja valitsemaan sopiva vaihtoehto kehitettäväksi systemiksi.

Chordin perustana ovat ulkoisten trendien ja sisäisten muutosten sekä olemassaolevien ongelmien suhteen tehtävät selvitykset. Ulkoisten trendien suhteen olennainen kysymys on mitkä ovat suuret muutokset, joita ilmenee tai oletetaan ilmenevän organisationaalissa ympäristössä, sisäisten muutosten kannalta taas miten haastateltava näkee tilanteen, miten sisäisiä työprosesseja tulisi kehittää ja mikä on todennäköisin strategia, jonka organisaatio omaksuu. Olemassaolevien ongelmien suhteen tulee tiedustella mitä asioita haastateltava pitää ongelmallisina työorganisaatiossa sekä voiko niitä ongelmia liittää laajempiin konteksteihin merkinä vajaavaisesta ulkoisten vaatimusten omaksumisesta.

Salmela kuitenkin painottaa, että olisi epärealistista olettaa olevan mahdollista selvittää kyselemällä kaikki tarpeet informaation systemin vaatimuksille. Tästä johtuen tulee analysoida olemassaolevaa systemiä, käyttäjien esittämiä vaatimuksia sekä kehittämisvaihtoehtoja, jotta saadaan realistinen tulos. Haastateltavien valinnan tärkeimpänä kriteerinä Salmela pitää sitä, ketkä pystyvät parhaiten parantamaan tulevaisuuden ennusteiden laatua. Toisaalta hän tuo esille, että on hyvä haastatella ihmisiä organisaation

eri puolilta ja tasoilta, korkein johto on yleensä paras ennustamaan sisäisiä ja ulkoisia muutoksia mutta toisaalta operationaalisen tason henkilöstö on paras kertomaan käytännöstä. Lisäksi muutos tarvitsee kaikkien ryhmien tuen, jotta se voisi toteutua.

Samaan teemaan viittaavat myös Kahler ja Rohde [1996] artikkelissaan koskien organisationaalista oppimista. He määrittelevät päivittäisen tiimityön välttämättömäksi alkuehdoksi organisationaaliseen oppimiseen. Oppivia organisaatioita he taas pitävät kyvykkäimpinä selviytymään monimutkaisessa ja dynaamisessa ympäristössä, joka on usein myös vihamielinen. Oppivan organisaation tunnusomaisia piirteitä ovat jatkuva kehittyminen ja kyky itsensä muuntamiseen. Sellaiset organisaatiot osaavat paitsi muuttaa omia tavoitteitaan ja päämääriään kollektiivisesti, myös muuttaakseen ja kehittääkseen strategioita, rakennetta, yleisiä organisationaalisen päätöksenteon sääntöjä ja normeja ja siten organisationaalista kulttuuria.

Tämä ajattelutapa johtaa siis yksilön oppimisesta kohti yhteisöllistä oppimista, jolloin myös asenteet muutosta kohtaan muuttuvat minä-ajattelun sijasta me-ajatteluun. Kyseinen malli on eräs tapa toimia muutoksista aiheutuvaa vastarintaa ehkäisevänä, sillä sen mukaan ei synny uhkaa yksilöä kohtaan, vaan halu olla mukana kehittämässä organisaatiota. Kahler ja Rohde kuitenkin toteavat, että käsite organisationaalista kulttuurista perustuu oletukselle siitä, että jaetut uskomukset ja asenteet määräävät yksilön ja kollektiivisen käyttäytymisen. Kuitenkin monet sosiaalipsykologiset tutkimukset osoittavat, että raportoidut asenteet ja todellisuudessa tapahtuva käyttäytyminen usein poikkeavat toisistaan.

Tukea organisaation kulttuuristen arvojen merkitykselle muutosprosessissa antaa myös Carlsonin [1997] tekemä tutkimus. Hän pitää tärkeänä sitä, että organisaation strategiset arvot linkitetään kulttuuristen arvojen kanssa yhteen, sillä monet case-tutkimukset osoittavat, että hyvää tarkoittaneet innovaatiot epäonnistuvat niiden joutuessa törmäyskurssille organisaatioon syvälle juurtuneiden toimintatapojen kanssa. Mikäli muutosta johdetaan ylhäältä käsin ja tarvetta koulutukselle sekä organisaatiokulttuurin vanhan rakenteen merkitystä vähätellään, tuloksena on yleensä epäonnistunut prosessi. Uudesta teknologiasta tulee työntekijöille vain taakka, jota täytyy väkisin hyödyntää ilman, että oikeastaan tiedetään miten sitä voidaan käyttää organisaation toiminnan parantamiseksi.

Tällaisissa tapauksissa uutta IT-innovaatiota aletaan pahimmillaan pitää uhkana omalle työlle. Uuden teknologian käyttöönotossa tulisikin Carlsonin mukaan hyödyntää työntö- ja vetovoimia (push and pull), eli suoritusta

palkitsevia ja toisaalta moraalisia mahdollistajia, jotta käyttöönotto sujuisi käyttäjiä tukevalla eikä uhkaavalla tavalla.

Wulf ja Rohde [1995] esittävät integroidun mallin organisaation ja teknologian kehitykselle. Tämä koostuu kuudesta vaiheesta, joita ovat prosessin perustamisvaihe, varsinaisen tilan analyysivaihe, vaihtoehtoisen mahdollisuuksien luomisvaihe, puuttumiskohtien suunnitteluvaihe, puuttumisvaihe sekä osallistumisen pätevyysvaihe.

Ensimmäisessä vaiheessa aloitetaan päivittäisessä työskentelyssä havaitusta organisationaalisen yksikön ongelmasta ja keskustellaan prosessin muutoksen tarpeesta, seuraavassa vaiheessa analysoidaan ongelmaa laajemmin organisaation rakenteen, teknologian ja pätevyyden kannalta. Kolmannessa vaiheessa luodaan vaihtoehtoiset mahdollisuudet prosessinmuutokselle, suunnittelussa voidaan käyttää hyväksi eri yhdistelmiä organisaation, teknologian ja laadun mittauksista ja arvioinnista. Näiden pohjalta keskustellaan ja pyritään löytämään yksimielinen ratkaisu. Työpsykologiset menetelmät ovat hyvä arvioinnin tuki eri vaihtoehtojen arvioinnissa.

Neljänneksi suunnitellaan miten puututaan organisationaalisiin, teknologisiin sekä laadullisiin ulottuvuuksiin ja tarpeen vaatiessa tutkitaan, miten sovelluksenkehittämisprojekti liitetään mukaan. Viidenneksi toteutetaan tekniset muutokset tarpeen mukaan räätälöinnin, uudelleen suunnittelun tai poistamisen keinoin, puututaan organisationaalisiin muutoksiin sekä varmistetaan tuloksen laatu. Kaikkien näiden vaiheiden läpi varmistetaan myös osallistujien pätevyys, siis että mukana ovat oikeat henkilöt oikeissa vaiheissa. Wulf ja Rohde painottavat, että integroidussa muutoksessa on tärkeää pitää oikeita henkilöitä mukana prosessissa, jotta varmistetaan, että muutokset vastaavat haluttua päämäärää ja niiden laatu on myös haluttua tasoa. Sosiaalisen kompetenssin eli kommunikaation ja yhteistyön taidot ovat alkuehtoina osallistumiselle. Lopuksi, kun muokattu työsystemi on saatu aikaiseksi, sitä pyritään arvioimaan ja tämän jälkeen mahdollisesti iteroimaan saman kaavan mukaan mahdollisten ongelmien poistamiseksi.

Nance [1996] puolestaan kirjoittaa informaatioteknologian ja informaationtoiminnan toimimisesta toisaalta muutoksen mahdollistajana ja toisaalta sitä ajavana voimana. Mahdollistajana IT helpottaa uudenlaisten kommunikointi- ja työtapojen muodostamista, jotka tukevat organisaation yrityksiä uudistaa työpaikan käytäntöjä. Muutosta ajavana voimana IT/IS voi myös laukaista ennalta suunnittelemtomia organisationaalisia siirtymiä kun taitovaatimukset muuttuvat organisaatiossa työntekijöiden käytössä olevien työkalujen ja teknologioiden tarjoamien uusien etujen myötä. Hän painottaa

sitä, että mahdollistajan ja ajavan roolin suhteen ei tulisi sotkea käsitteitä keskenään, kuten monissa aiemmissa tutkimuksissa on tehty.

IT/IS:n rooli muutoksen mahdollistajana on Nancen mukaan monien tutkimusten tuloksena kiteytetty ihmistyövoiman tarpeen vähenemiseen, ajan ja etäisyyksien vähenemiseen sekä työtehtävien suorittamisen nopeutumiseen. Ne voivat muuttaa työtehtävien keskittymispistettä, työssä käytettäviä työkaluja ja monia muotoja, joilla kontrolloidaan työskentelyä. Joskus ne voivat myös helpottaa koko organisaation uudelleenjärjestelyä ja toimia tehokkaana kulttuurallisen muutoksen mahdollistajana.

Nance kommentoi IT/IS:n roolin muutosta ajavana olevan huomattavasti vähemmän tutkittu alue, ja että usein ne keskittyvät laajempialaisiin strategisiin ulottuvuuksiin. Toisaalta hän jatkaa, että joissakin tapauksissa teknologiaa kehitetään ja levitetään koko yhteisön läpi, jolloin yksilöt ja organisaatiot tutkivat uusia mahdollisuuksia. Kun uusia käyttökelpoisia mahdollisuuksia löytyy, tehdään mahdollisesti päätöksiä muuttaa entisiä toimintoja, jotta uusien mahdollisuuksien hyödyntämiseksi.

4. Muutosvastarinnan aiheuttamien ongelmien kohtaaminen

Tutkimusaineistoni perusteella voidaan päätellä, että tutkimusta siitä miten muutosvastarinnan aiheuttamia ongelmia tulisi organisaatiossa käsitellä on tehty varsin vähän. Suurin osa tähän mennessä tehdyistä tutkimuksista näyttää keskittyvän määrittelemään muutosprosesseissa tapahtuneita epäonnistumisia ja mitä niistä on opittu, eli miten uuden teknologian käyttöönotto tulisi suorittaa ilman, että muutosvastarinnasta syntyisi ongelmia organisaatiolle. Viittaukset eteen tulleiden ongelmien kohtaamiseen ovat hajanaisia ja nekin yleensä epäsuoria, enemmän sivulauseissa olevia mainintoja kuin käytännön tutkimuksesta saatuja selkeitä tuloksia.

Rivien välistä tulkittuna näistä tutkimuksista voidaan johtaa ongelmien kohtaamisen tärkeimmäksi apuvälineeksi organisaation yleisen ilmapiirin luominen sellaiseksi, että avoin keskustelu ja asioista tiedottaminen on mahdollista, tätä kautta ongelmien kohtaaminen sujuu parhaiten. Mikäli ilmapiiri ei tue avointa keskustelua, ongelmaa on vaikea lähestyä, kun sitä voidaan enimmäkseen pienissä kuppikunnissa organisaation eri osissa. On tärkeää, että organisaatiokulttuuri pystyttäisiin säilyttämään yhtenäisenä ja kaikki kokisivat muuttuvan tilanteen vievän eteenpäin kohti parempaa tulosta.

5. Yhteenveto

Muutosvastarintaa on tutkittu vuosien saatossa paljon ja monia syitä sen ilmenemiselle on esitetty, toiset niistä ovat yleisesti vakiintuneet

määrittelemään mitä muutosvastarinnalla tarkoitetaan. Myös monenlaisia teorialleja muutosvastarinnan ennaltaehkäisyyn on kehitetty ja monia niistä on myös käytännössä testattu ja havaittu toimiviksi ratkaisuuksi, toiset teorit ovat saaneet uudenlaisia muotoja ja lisäyksiä myöhemmässä tutkimuksessa.

Teknologian kehittyessä muutokset organisaatioissa tulevat kuitenkin yhä jatkumaan ja huomiota muutosvastarinnan ilmenemiseen on syytä kiinnittää jatkossakin. Tulevaisuudessa voitaisiin myös keskittyä tutkimaan miten organisaatio parhaiten toipuu muutosvastarinnan aiheuttamista ongelmista, monet tutkimukset esittävät esimerkkitapauksia epäonnistuneista projekteista mutta harva tarjoaa keinoja työstää tilannetta epäonnistumisesta eteenpäin.

Viiteluettelo

- [Carlson, 1997] Patricia Carlson, Information Technology and Organizational Change. In: *Proc. of the 17th Annual International Conference on Computer Documentation*, 26-35.
- [Dray, 1994] Susan M. Dray, Minimizing Organizational Risks of Technological Change. In: *Conference Companion on Human Factors In Computing Systems*, 385-386.
- [Hartikka, 1990] Juha Hartikka, Uuden ajatuksen tie menestykseen kulkee muutosvastarinnan kautta: innovaatio yhteiskunnassa. Saatavissa osoitteesta <http://guns.connect.fi/innoplaza/etusivu/kehitys.html> (linkki tarkastettu 18.11.2003)
- [Hodgson and Aiken, 1998] Lynda Hodgson and Peter Aiken, Organizational Change Enabled by the Mandated Implementation of New Information Systems Technology: a Modified Technology Acceptance Model. In: *Proc. of the 1998 ACM SIGCPR Conference on Computer Personnel Research*, 205-213.
- [Kahler and Rohde, 1996] Helge Kahler and Markus Rohde, Changing to stay itself. *ACM SIGOIS Bulletin* 17, 3 (Dec. 1996), 62-64.
- [Keen, 1981] Peter G. W. Keen, Information systems and organizational change. *Communications of the ACM* 24, 1 (Jan. 1981), 24-33.
- [Nance, 1996] William D. Nance, An Investigation of Information Technology and the Information Systems Group as Drivers and Enablers of Organizational Change. In: *Proc of the 1996 ACM SIGCPR/SIGMIS Conference on Computer Personnel Research*, 49-57.
- [Qualitas Fennica Oy, 2000], saatavissa osoitteesta <http://cgi.qualitas-fennica.fi/artikkelit/muutosvastarinta.html> (linkki tarkastettu 23.11.2003)

- [Salmela, 1993] Hannu Salmela, Designing Information Systems for Changing Organizations. In: *Proc. of the 1993 Conference on Computer Personnel Research*, 243-254.
- [Sonquist, 1976] John A. Sonquist, Organizational Change and Educational Computing. In: *Proc. of the Annual Conference*, 462-466.
- [Turun yliopiston täydennyskoulutuskeskus, 2001], saatavissa osoitteesta http://www.tkk.utu.fi/osaaminen/osaamisen_ohjaaminen/43muutosvas.tarinta.html (linkki tarkastettu 24.11.2003)
- [Wulf and Rohde, 1995] Volker Wulf and Markus Rohde, Towards an Integrated Organization and Technology Development. In: *Proc. of the Conference on Designing Interactive Systems Processes, Practices, Methods & Techniques*, 55-64.

Kämmentietokone ryhmän tukemisessa

Tuomo Kivinen

Tiivistelmä.

Tutkimuksessani perhehdyn kämmentietokoneisiin ja siihen miten niitä voidaan käyttää ryhmän tukena. Esittelen myös jonkin verran ryhmätuen vaatimaa verkkoarkkitehtuuria, jonka avulla kämmentietokoneet ja muut laitteet kommunikoivat keskenään. Kartoitan ja arvioin myös erityyppisiä ryhmätukisovelluksia, jotka mahdollistavat kämmentietokoneella toimivan ryhmän yhteistoiminnallisuuden. Nykyään on jo olemassa ja kehitteillä monia käyttökelpoisia ja innovatiivisia sovelluksia, jotka tarjoavat aivan uusia mahdollisuuksia ryhmätyöskentelyyn ja käyttäjien väliseen yhteydenpitoon.

Avainsanat ja -sanonnat: kämmentietokone, langaton tiedonsiirto, mobiiliverkko, yhteistoiminnallisuus, ryhmätukisovellus.

CR-luokat: C.2.1, C.2.4, C.5.3, K.3.1

1. Johdanto

Kämmentietokoneiden käyttö on yleistynyt ja laitteet ovat kehittyneet kohtuullisen tehokkaiksi ja monipuolisiksi. Nykyään niillä on paljon muutakin käyttöä kuin pelkästään käyttäjän kalenterina ja muistiona toimiminen. Kämmentietokoneet mahdollistavat tietokoneiden käytön aivan uusissa ympäristöissä ja tilanteissa, kuten sairaaloissa ja kenttäolosuhteissa, mikä oli aikasemmin oikeastaan mahdotonta.

Langaton tiedonsiirto on kehittynyt viime vuosina räjähdysmäisesti ja sen myötä myös mobiililaitteiden tiedonsiirto on saanut aivan uuden merkityksen. Matkapuhelin ei ole välttämättä enää ainut laite, joka käyttäjällä on mukanaan kommunikointia varten. Nyt se voi yhtä hyvin olla kämmentietokone, joka toimii käyttäjien välisenä kommunikointilaitteena sähköposteineen ja satelliitti-paikannuksineen. Langaton tiedonsiirto mahdollistaa myös erilaisten yhteistoiminnallisten mobiiliverkkojen muodostamisen ja näissä toimivien ryhmätukisovellusten käytön.

Ryhmätukisovellukset ovat koko ajan kehittymässä yhä paremmiksi ja monipuolisemmiksi, koska niiden avulla työskenteleminen on aikaisempaa tehokkaampaa ja mielekkäämpää. Sovelluksia kehitetään yhdessä käyttäjien kanssa ja testataan jo hyvissä ajoin ennen markkinoille tuontia. Näin niistä

saadaan huomattavasti käyttäjäystävällisempiä, joka osaltaan edesauttaa niiden käyttöönottoa erilaisissa ympäristöissä.

Luvussa kaksi käsitellään kämmentietokonetta laitteena sekä sen ominaisuuksia ja käyttömahdollisuuksia. Langattoman tiedonsiirron eri tekniikoita ja tiedonsiirtoon liittyviä asioita, kuten tietoturvaa esitellään lyhyesti kolmannessa luvussa. Neljäs luku sisältää tietoa yhteistoiminnallisen mobiiliverkon suunnittelusta ja siihen liittyvistä ongelmista. Luvussa esitellään myös kaksi erilaista yhteistoiminnallisen verkon arkkitehtuuria. Viidennessä luvussa käsitellään kämmentietokoneille suunniteltuja ryhmätukisovelluksia, jotka olen jakanut kolmeen eri ryhmään: perinteiset ryhmätukisovellukset, koulussa käytettävät ryhmätukisovellukset sekä erikoistilanteisiin suunnatut sovellukset. Kuudennessa luvussa pohdin mitä nykyisillä sovelluksilla on saatu aikaan ja mitä tulevaisuudessa tulisi tehdä ja kehittää.

2. Kämmentietokoneet

Kämmentietokoneet yleistyvät nykypäivän informaatioyhteiskunnassa yhä nopeammin. Erilaisia sähköisessä muodossa olevia tietoja on jo tarvetta kantaa mukanaan joka paikassa ja niitä pitää myös pystyä siirtämään laitteesta toiseen helposti ja vaivattomasti. Esimerkiksi erilaiset tekstidokumentit tai kalvoesitykset voidaan kuljettaa kämmentietokoneen muistissa kotoa töihin ja sen jälkeen siirtää ne langattomasti omalle työkoneelle ja vaikka tulostaa ne. Kaikki henkilökohtaiset tiedot voidaan siis kämmentietokoneen avulla kuljettaa aina helposti mukana omassa taskussa.

Ennen kämmentietokoneet olivat vain eräänlaisia elektronisia kalentereita, jotka eivät oikeastaan sisältäneet muita toimintoja kuin erilaisten tapaamisten ja muistutusten organisoimista. Nykyään laitteiden laskentateho ja muistikapasiteetti ovat sitä luokkaa, että kämmentietokoneiden avulla voidaan tehdä lähes kaikkea aina tekstinkäsittelystä elokuvien katseluun. Työasemiin verrattuna kämmentietokoneet ovat vielä kuitenkin auttamattoman hitaita, mutta kehitystä tapahtuu jatkuvasti.

Tyypillinen kämmentietokone on litteällä kosketusnäytöllä varustettu ja nimensä mukaisesti noin kämmenti kokoinen tietokone, joka saa virtansa aku(i)sta. Nykyisin myynnissä olevat laitteet on jo suurilta osin varustettu värinäyttöillä, mikrofoneilla ja kaiuttimilla, joten käytettävät sovellukset voivat tarvittaessa sisältää jopa multimediaominaisuuksia. Kämmentietokoneet voidaan usein myös kytkeä työasemiin erillisen linkkikaapelin tai telakointiyksikön avulla ja siirtää tietoa laitteiden välillä.

Kämmentietokoneen toimintoja ohjataan rungossa sijaitsevien painonappien, kosketusnäytön tai joissakin tapauksissa jopa puheen avulla. Tekstin-

syöttö laitteeseen onnistuu käsialatunnistuksen tai näytölle ilmestyvän näppäimistön avulla. Yleensä kämmentietokone sisältää myös mahdollisuuden monien lisälaitteiden, kuten esimerkiksi ulkoisen näppäimistön tai langattoman verkkokortin, liittämiseksi. Myös erilaisia muistilaajennuspaikkoja on yleensä tarjolla ainakin yksi kappale.



Kuva 1. Tyypillinen kämmentietokone

3. Langaton tiedonsiirto

Langattoman tiedonsiirron suosio on jatkuvassa nousussa ja eri tekniikat kehittyvät koko ajan. Mainittakoon näistä yleisimpinä Bluetooth, WLAN (Wireless Local Area Network), IrDA (Infrared Data Association), sekä GPRS (General Packet Radio Service).

Näistä kaksi ensimmäistä toimivat radiotaajuuksilla, IrDA infrapunavalolla ja GPRS matkapuhelinverkon välityksellä. Varsinkin IrDA vaatii tiedonsiirtoon lyhyen etäisyyden ja selvän näköyhteyden laitteiden välillä, mikä rajoittaa jonkin verran sen käyttöä. Pitkien etäisyyksien väliseen tiedonsiirtoon sopii parhaiten GPRS, koska se toimii lähes kaikkialla missä matkapuhelimetkin toimivat. Tekniikoista Bluetooth ja WLAN ovat yleisimmin käytettyjä, sillä ne ovat esitellyistä tekniikoista nopeimmat.

Tänä päivänä langattoman tiedonsiirron nopeus on jo sitä luokkaa, että sitä käytetään jo suurtenkin datamäärien siirtoon. Myös laitteiden hinnat ovat pudonneet kuluttajaystävälliselle tasolle ja kämmentietokoneidenkin langattomia yhteysratkaisuja on saatavilla kohtuuhintaan.

Suurimmiksi ongelmiksi langattomassa tiedonsiirrossa ovat osoittautumassa yhteystkatkot ja sitä paljon vakavampi tietoturva. Erilaisten langattomien yhteyksien heikkeneminen esimerkiksi sään tai muiden fyysisten esteiden takia on aivan normaalia. Tietoturvakysymys on hankala, koska langattoman tiedonsiirron salaaminen on paljon haastavampaa kuin tavallisen, sillä signaalit

kulkevat yleisesti tiedossa olevilla radiotaajuuksilla ja kuka tahansa saattaa päästä niihin käsiksi. Lisäksi langattomien laitteiden, kuten kämmen-tietokoneiden resurssit eivät riitä kovinkaan raskaiden salausalgoritmien käyttämiseen.

4. Yhteistoiminnallisen mobiiliverkon suunnittelu

Yhteistoiminnallisella verkolla tarkoitetaan tietoverkkoa, joka toimii erilaisten ryhmien apuvälineenä yhteisen tavoitteen saavuttamisessa. Verkon avulla voidaan muun muuassa jakaa informaatiota ja olla vuorovaikutuksessa muiden ryhmän jäsenten kanssa. Erityisesti jos toimitaan mobiilissa ympäristössä, on yhteistoiminnallisen verkon mukauduttava uusiin laitteisiin ja resursseihin, jotka ovat joko tulleet verkkoon tai lähteneet sieltä pois.

4.1. Yleisiä suunnitteluun liittyviä ongelmia

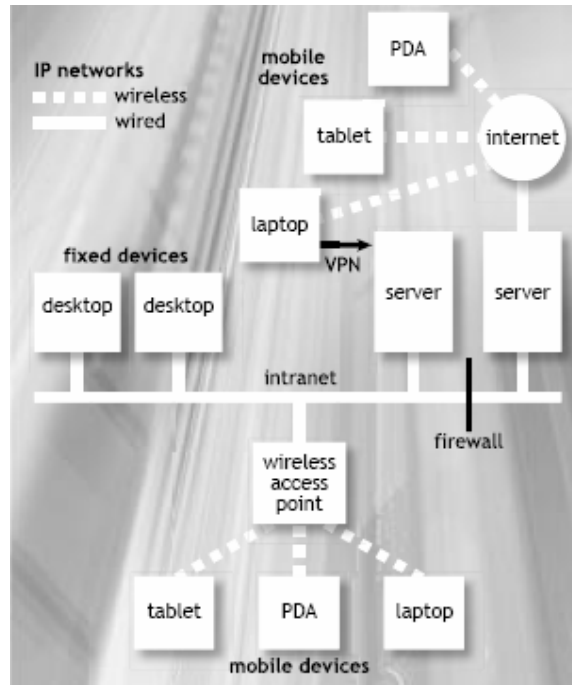
Luvussa kolme mainittujen langattomaan tiedonsiirtoon liittyvien ongelmien lisäksi kokonaisen langattoman yhteistoiminnallisen verkon suunnitteluun liittyy monia muitakin ongelmia. Sen suunnittelu ja toteuttaminen on todella vaativaa, sisältäen monia huomioonotettavia asioita.

Eräs ongelma on se, että laitteiden ja ohjelmistojen tulisi olla helppokäyttöisiä eli ne eivät saisi viedä käyttäjien huomiota itseensä, pois oleellisesta eli vuorovaikutuksesta. Mobiililaitteilla tapahtuvan vuoro-vaikutuksen tulisi olla sujuvaa ja ennen kaikkea joustavaa [Wiberg, 2001].

Suunnittelun vaikeutena on omalta osaltaan ollut myös se, että ohjelmistokehityksessä on keskitytty kahteen eri asiaan, mobiiliuteen ja toisaalta erikseen yhteistoiminnallisuuteen. Yhteistoiminnallisuuden tulisi olla osa muuta suunnittelua, jotta ohjelmistoista ja palveluista syntyisi sellaisia, että ne mukautuvat mahdollisimman hyvin erilaisiin tilanteisiin. Jos näin ei tehdä, järjestelmä pakottaa käyttäjän mukautumaan hankaloittaen käyttäjän toimintaa [Bartram and Blackstock, 2003]. Eli tämä tarkoittaa sitä, että käyttäjä joutuu esimerkiksi käsin muuttamaan sovellusten asetuksia, jotta ne toimisivat halutulla tavalla muuttuneessa tilanteessa.

4.2. Keskitetyt yhteistoiminnalliset mobiiliverkot

Keskitetyllä yhteistoiminnallisella verkolla tarkoitetaan yhteistoiminnallista tietoverkkoa, jonka arkkitehtuuri koostuu niin langattomista päätelaitteista kuin kiinteässä verkossa toimivista laitteista, kuten pöytätietokoneista ja palvelimista. Nämä kiinteästi verkossa olevat laitteet toimivat siten koko verkon runkona ja langattomat laitteet kytkeytyvät tähän erilaisin langattomin menetelmin.



Kuva 2. Esimerkki keskitetyn yhteistoiminnallisen verkon arkkitehtuurista [Bartram and Blackstock, 2003].

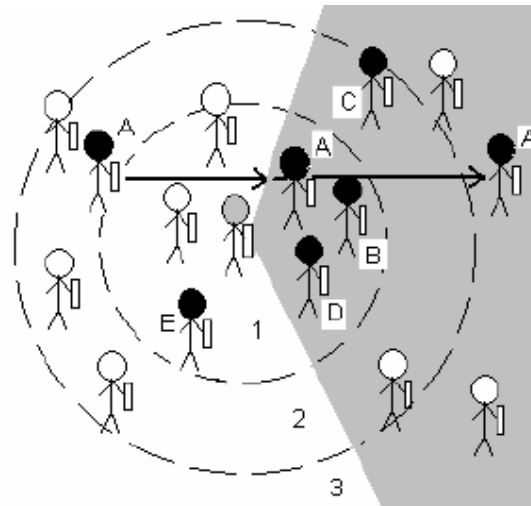
Keskitetyn yhteistoiminnallisen verkon palvelut keskitetään pöytätietokoneille ja palvelimille, koska mobiililaitteiden resurssit ovat huomattavasti niitä huonommat. Mobiililaitteet kommunikoivat toistensa kanssa keskitetysti jonkin kiinteässä verkossa olevan palvelimen kautta. Palvelin hoitaa datan tallennuksen ja levityksen lisäksi myös erilaisten yhteistoiminnallisten sovellusten ajamisen, joita mobiililaitteet käyttävät.

4.3. Dynaamiset yhteistoiminnalliset mobiiliverkot

Dynaamiset yhteistoiminnalliset mobiiliverkot koostuvat kannettavista laitteista, kuten kämmentietokoneista. Verkko on nimensä mukaan dynaaminen, koska siihen voidaan liittyä ja siitä voidaan poistua minä hetkenä hyvänsä. Kannettavat laitteet tiedustelevat langattoman verkon avulla mitä laitteita verkossa on ja tämän perusteella määritetään yhteisesti käytettävissä olevat resurssit. Esimerkiksi jos verkon yhdessä laitteessa on GPS paikannusjärjestelmä, niin verkon muut jäsenet voivat seurata tämän avulla omaa sijaintiaan tai jos yhdessä laitteessa on suuri tallennuskapasiteetti, niin sitä voidaan käyttää yhteisesti.

Dynaamisen yhteistoiminnallisen verkon pääominaisuus on juuri se, että verkon toiminnot ja ylläpitäminen hajautetaan käyttäjien kesken. Tämän pitäisi toimia mahdollisimman näkymättömästi kaiken toiminnan taustalla, koska

kukaan osanottajista tuskin haluaa ottaa toimintojen johtamista hoidettavakseen [Buszko *et al.*, 2001].



Kuva 3. Dynaamisen mobiiliverkon muodostaminen [Wiberg, 2001].

Wiberg [2001] esittää tutkimuksessaan neljä erilaista muuttujaa, joiden avulla voidaan erottaa dynaamisesta mobiiliverkosta erilaisia kokonaisuuksia eli käyttäjäryhmiä. Nämä ovat etäisyys, ennaltamääräytyt ryhmät, suunta, sekä aika. Kuvassa kolme on kuvattuna satunnainen ryhmä, josta keskimmäisen käyttäjä haluaa erottaa itselleen merkitykselliset jäsenet. Huomioitavaa on, että henkilö A liikkuu nuolten osoittamasti oikealle.

Etäisyys on jaettu kolmeen eri alueeseen, joista tässä tapauksessa otetaan mukaan vain lähimmän eli alueen numero yksi jäsenet eli yhteensä kuusi jäsentä. Seuraavaksi tutkitaan ennaltamääräytyä ryhmää, joiden jäsenillä on kuvassa mustat päät (A, B, C, D ja E). Näistä ryhmään otetaan mukaan vain A, B, D ja E, koska ne ovat etäisyydellä yksi, C:n ollessa liian kaukana. Tämän jälkeen tutkitaan suuntaa, joka voi olla esimerkiksi käyttäjän rintamasuunta (kuvassa harmaa alue). Nyt ryhmään kuuluvat jäsenet A, B ja D, koska E ei selvästikään ole käyttäjän etupuolella. Viimeiseksi tarkastellaan vielä aikaa, jonka ryhmän jäsenet ovat käyttäjän läheisyydessä. Valitusta ryhmästä A, B ja D karsiutuu vielä pois jäsen A, koska hän liikkuu käyttäjän ohitse, eikä siis viivy käyttäjän läheisyydessä tarpeeksi kauaa.

Menetelmä on siis hyvin yksinkertainen ja se voidaan esittää muodossa etäisyys \cap ennaltamääräytyt ryhmät \cap suunta \cap aika. Tästä seuraa, että kaikkien kyseisten ehtojen tulee täyttyä, jotta tulee valituksi ryhmän jäseneksi. Jotain tämänkaltaista on varmasti pakko olla käytössä kun halutaan dynaamisesti muodostaa järkeviä ja toimivia ryhmiä, muuten ryhmät jäsenet olisivat aivan satunnaisesti valittuja.

5. Ryhmätukisovelluksia

Ryhmätukisovellus on kirjaimellisesti sovellus, jota monta käyttäjää pystyy käyttämään, joko synkronisesti tai asynkronisesti. Lisäksi sovellus sisältää yhteistoiminnallisuutta eli käyttäjät voivat yhdessä muokata tietoja ja toimia yhteistä tavoitetta kohti. Kämmentietokoneiden verkko-ominaisuuksien koko ajan kehittyessä paremmiksi ovat myös kämmentietokoneiden ryhmätukisovellukset kehittyneet.

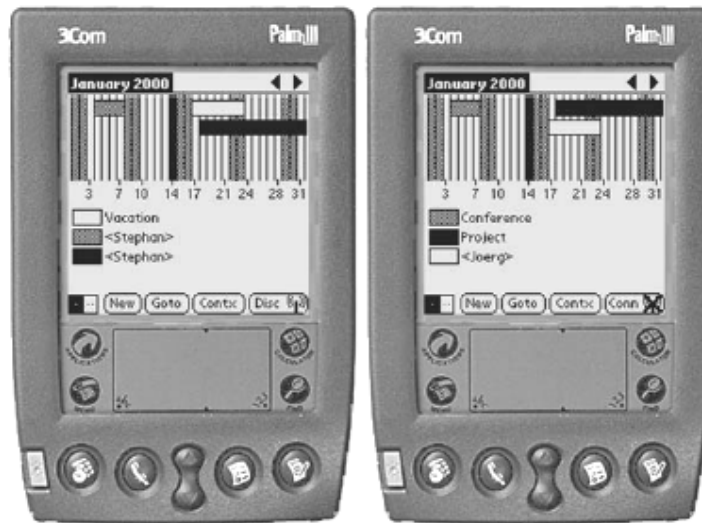
5.1. Perinteisiä ryhmätukisovelluksia

Perinteiset ryhmätukisovellukset ovat ohjelmistoja, joiden avulla käyttäjät pystyvät helposti jakamaan tietoa ja pitämään toisiinsa yhteyttä, sekä suunnittelemaan yhteisiä aikatauluja. Lisäksi perinteiset ryhmätukisovellukset sisältävät usein myös mahdollisuuden ryhmän eri jäsenten paikantamiseen.

5.1.1. Quickstep

Quickstepin avulla voidaan kehittää ryhmätukisovelluksia kämmentietokoneille. Järjestelmä on suunniteltu pääosin synkronista eli samanaikaista toimintaa ajatellen. Quickstep sisältää myös yleisen palvelinohjelman, jota voidaan käyttää kaikkien Quickstepin alustalle kehitettyjen sovellusten apuna ilman palvelimen uudelleenkonfigurointia [Roth and Unger, 2001].

Esimerkkinä Quickstepin alustalle kehitetystä sovelluksesta on ryhmäkalenteri, jonka avulla ryhmän jäsenet pystyvät sopimaan tapaamisia ja kokouksia ajalle, joka sopii kaikille jäsenille. Sovelluksessa on otettu hyvin huomioon myös yksityisyys; toiset ryhmän jäsenet eivät näe, mitä muut ovat kalenteriinsa laittaneet vaan pelkästään kyseisen ajan olevan varattu. Tämä näkyy kuvassa neljä siten, että varatun ajan kohdalla on kyseisen henkilön nimi, jolla kyseinen aika on jo varattu muuhun toimintaan.

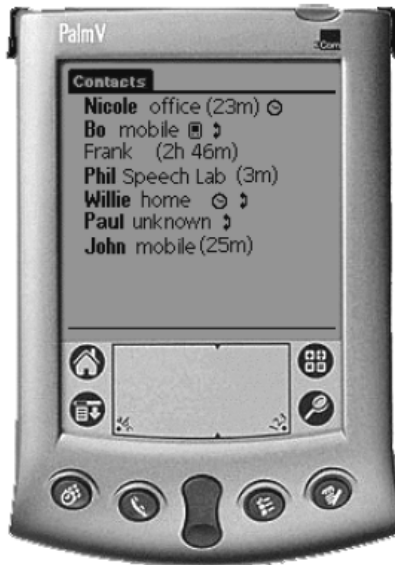


Kuva 4. Ryhmäkalerisovellus. Vasemmalla Joergin ja oikealla Stephanin näyttöruutu [Roth and Unger, 2001].

Quickstepin alustalle on kehitetty myös muuan muassa ryhmän aivoriihi sekä käyntikorttien vaihtoon soveltuva ohjelma. Järjestelmä sisältää myös mahdollisuuden seurata verkkoyhteyden tilaa, mihin palvelimeen ollaan kulloinkin kytkeytyneenä, missä palvelin sijaitsee, mihin otetaan yhteyttä vikatilanteessa sekä ketkä muut ryhmän jäsenet ovat kytkeytyneenä verkkoon [Roth and Unger, 2001]. Nämä ovat varsin hyödyllisiä toimintoja, koska langattomat verkkoyhteydet saattavat katketa hetkinä minä hyvänsä ja tällä tavoin saadaan selkeä kuva siitä ketkä ovat saavutettavissa. Hyvä puoli on myös se, että kaikki ryhmän käytössä olevat tiedot myös tallennetaan ja varmuuskopioidaan eri palvelimille, jotta ne olisivat varmasti tallessa.

5.1.2. Awarenex

Awarenex on ryhmäsovellus, jonka avulla voidaan saada tietoa siitä, missä kukin ryhmän jäsenistä kulloinkin on ja miten heihin mahdollisesti saa yhteyttä. Järjestelmää on kehitetty haastatteleamalla kokeneita mobiilikäyttäjiä ja sen pohjalta nousivat esiin seuraavat tarpeet: omiin tietoihin on päästävä käsiksi joka paikasta (kotoa, töistä, liikkeessä) ja mobiililaitteita tulisi pystyä käyttämään yhdessä (kämmentietokone - puhelin). Mobiilikäyttäjät vaativat ohjelmilta myös hyvää suunnittelua ja kohtuullisessa ajassa saatavaa palautetta [Tang *et al.*, 2001]. Nämä ovat varmasti hyvä pohja suunnittelulle.



Kuva 5. Awarenexin kontaktilista [Tang *et al.*, 2001].

Kuten kuvasta viisi nähdään on Awarenexin kontaktilistassa näkyvissä ryhmän eri jäsenet ja heidän tämänhetkinen tilansa. Listasta näkee aktiiviset ryhmän jäsenet (korostettu nimi), heidän sijaintinsa, sovitut tapaamiset (kello) ja vielä tavan, jolla heihin saa yhteyttä (esim. puhelin, kämmentietokone). Ruudulla sulkeissa oleva aika ilmoittaa myös, kuinka kauan kyseinen henkilö on ollut toimettona (idle), joten senkin avulla voidaan päätellä saako henkilöön halutessaan yhteyttä.

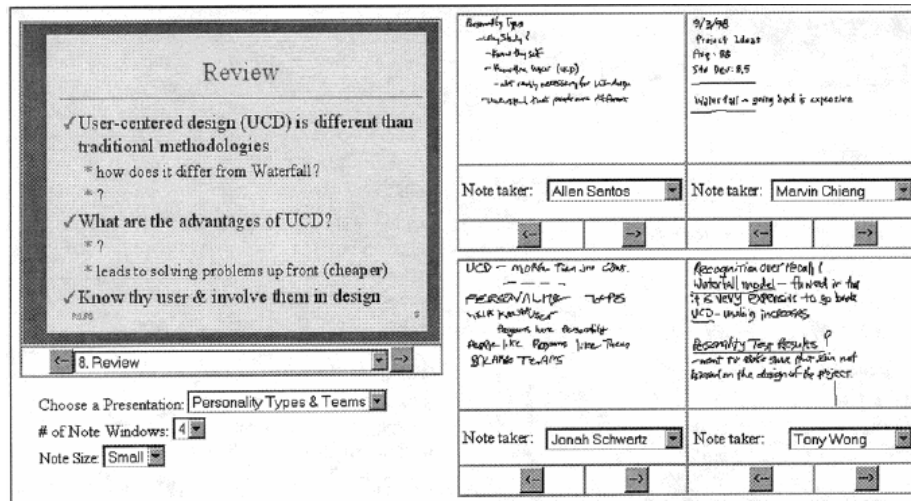
Kun käyttäjä haluaa ottaa yhteyttä johonkin ryhmän jäseneseen, hän voi valita kyseisen henkilön kontaktilistasta ja valita yhteystavan, kuten sähköposti tai matkapuhelin. Tämän jälkeen on mahdollista, että Awarenex jopa soittaa puhelun käyttäjän puhelimella halutulle ryhmän jäsenelle käyttäjän puolesta eli hänen ei tarvitse edes näppäillä puhelinnumeroa. Samalla järjestelmä päivittää molempien käyttäjien tilaksi kesken olevan puhelun kaikkien ryhmän jäsenten kontaktilistoihin. Tilan päivitys tapahtuu myös tilanteessa, jossa käyttäjä kirjautuu esimerkiksi sisään työasemalleen tai kämmentietokoneelleen.

Awarenex on varsin käyttökelpoinen ryhmätukisovellus varsinkin sellaisille ryhmille, joiden täytyy pitää tiiviitä yhteyksiä toisiinsa ja tietää kohtuullisen hyvin, missä kukin ryhmän jäsenistä on ja mitä he mahdollisesti tekevät. Voi toisaalta myös olla kiusallista, jos toiset ryhmän jäsenet tietävät koko ajan käyttäjän sijainnin.

5.1.3. Notepals

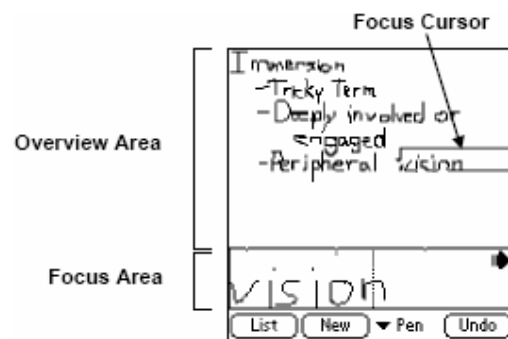
Notepals on ryhmän muistiinpanosovellus, joka mahdollistaa myös muistiinpanojen tehokkaan jakamisen muiden kesken. Sovellus sisältää

kämmmentietokoneella tapahtuvan muistiinpanojen kirjoittamisen ja jakamisen lisäksi mahdollisen käsialatunnistuksen, työasemalla tapahtuvan muistiinpanojen selaamisen sekä liittämisen kalvoesityksiin. Muistiinpanot voidaan myös tallentaa keskitetysti palvelimelle.



Kuva 6. Notepalsin webbikäyttöliittymä, jonka avulla voidaan selata monen käyttäjän muistiinpanoja ja kalvoesitystä rinnakkain [Davis *et al.*, 1999].

Notepalsia on testattu erilaisissa konferensseissa ja myös yhdessä koulussa. Testeissä huomattiin, että muistiinpanojen kirjoittaminen, kuten myös lukeminen on hitaampaa kuin normaalien paperimuistiinpanojen. Kirjoitetuista muistiinpanoista oli hankalampi saada selvää, koska huono käsiala saattoi kämmmentietokoneelle kirjoitettuna olla vielä epäselvempää kuin normaalisti [Davis *et al.*, 1999]. Tämän huomaa varsinkin kuvassa seitsemän.



Kuva 7. Muistiinpanojen kirjoittamista kämmmentietokoneella Notepalsin avulla [Landay, 1999].

Koulussa varsinkin nuoremmat opiskelijat vierastivat Notepalsin käyttöä, koska se oli heidän mielestään liian hidasta ja hankalaa. Suurin osa heistä lopettikin Notepalsin käytön kurssin muistiinpanojen tekemiseen ja käytti sen

sijaan tavallista kynää ja paperia. Vanhemmat opiskelijat käyttivät Notepalsia normaalisti muistiinpanojensa tekemiseen ja totesivat sen erityisen käteväksi ryhmätyöskentelyssä [Landay, 1999].

Notepals voisi pienen kehitystyön jälkeen olla hyvä sovellus, mutta tällä hetkellä en ainakaan itse käyttäisi sitä. Toistaiseksi käytän itsekin suurimmaksi osaksi muistiinpanojen tekemiseen kynää ja paperia samasta syystä kuin esimerkin opiskelijatkin. Toisaalta kämmentietokone kulkee mukana paljon helpommin kuin tavalliset muistiinpanovälineet, joten tämänkaltaisiin sovelluksiin tulisi mielestäni panostaa enemmän. Ryhmämuistiinpanojen toimivuutta olisi myös mukava kokeilla käytännössä.

5.1.4. Community viewer

Community viewer on ryhmätukisovellus, jonka avulla käyttäjät voivat etsiä samoista asioista kiinnostuneita henkilöitä ja lähettää viestejä toisilleen. Sovellus käyttää hyväkseen kokoustila -metaforaa, joka osaltaan helpottaa ihmisten lähestymistä [Nishimura *et al.*, 1998]. Kämmentietokoneen näytöllä on siis huoneen näköinen tila, jossa käyttäjät liikkuvat erinäköisinä ikoneina.

Aluksi käyttäjä syöttää Community viewerille omat tietonsa: itsensä esittely, työn ja vapaa-ajan kiinnostuksen kohteet, harrastukset jne. Tämän jälkeen käyttäjä pääsee kokoushuoneeseen, jossa kaikki henkilöt esitetään kasvojen näköisinä ikoneina. Järjestelmä tekee automaattisesti samoista asioista kiinnostuneista käyttäjistä ryhmiä, jotka erottuvat ruudulla selvästi. Muut käyttäjät levittyvät näiden ryhmien ympärille. Tämän jälkeen käyttäjä voi halutessaan katsella toisten käyttäjien profiileja, jolloin heidän ikoninsa lähestyvät toisiaan. Käyttäjät voivat myös lähettää toisilleen viestejä, jolloin heidän ikoninsa muuttuvat puhuvien näköisiksi. Ilmeet muuttuvat myös sen mukaan, miten paljon tai vähän kyseinen käyttäjä on kommunikoinut muiden kanssa.



Kuva 8. Käyttäjä katsoo toisen käyttäjän henkilötietoja [Nishimura *et al.*, 1998].

Nishimura ja kumppanit [1998] testasivat Community vieweriä vuoden 1996 ICMAS (international conference on multiagent systems) konferenssissa.

Konferenssin osanottajille jaettiin kämmentietokoneet, joista oli langaton yhteys matkapuhelinten avulla Community viewerin palvelimeen. Konferenssin päätyttyä huomattiin, että järjestelmää oli käytetty ahkerasti koko tapah-tunan ajan, jopa konferenssialueen ulkopuolella ja öiden pikkutunneille asti. Community viewerin avulla pystyttiin siis edesauttamaan käyttäjien välistä kanssakäymistä [Nishimura *et al.*, 1998].

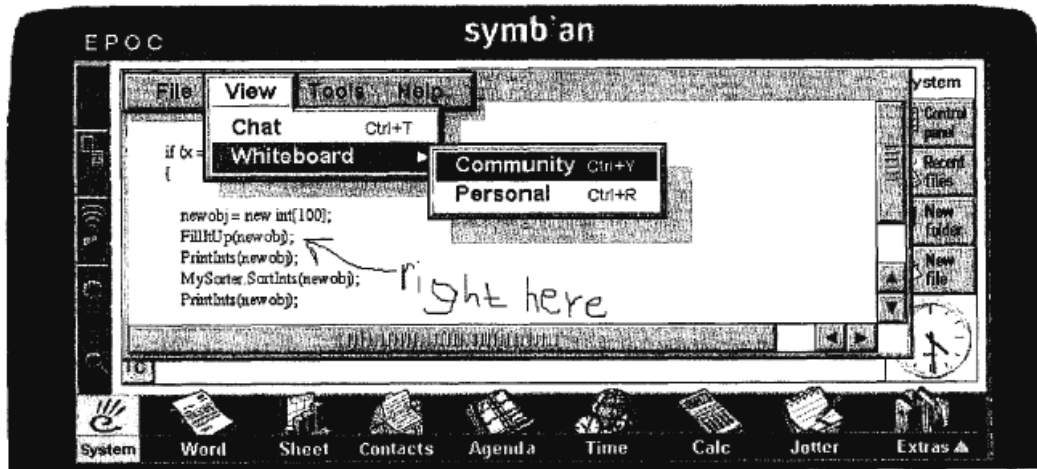
Community viewerin avulla on varmasti helpompi luoda kontakteja samanhenkisiin henkilöihin, koska asia esitetään selkeästi kämmentietokoneen ruudulla ja järjestelmän käyttö ei vaadi juuri lainkaan opettelua. Koska järjestelmä on kohtuullisen vanha (ainakin vuodelta 1996), olisi mukava tietää, onko sitä kehitetty eteenpäin. Tämänkaltainen järjestelmä soveltuu hyvin käytettäväksi erilaisissa kokouksissa ja konferensseissa, koska sen avulla löytää nopeasti ihmisiä, joilla on samat kiinnostuksen kohteet, näin heitä ei tarvitse etsiä väkijoukon seasta.

5.2. Kämmentietokoneet koulussa

Kämmentietokoneiden käyttöä kouluissa on jo kokeiltu, mutta toistaiseksi se on todella vähäistä. Syinä tähän voivat olla kämmentietokoneiden hinta ja koulukäyttöön soveltuvien ohjelmistojen puuttuminen markkinoilta. Tulevaisuudessa asia luultavasti muuttuu ja kämmentietokoneet lisääntyvät opetuksen tukivälineinä ja korvaavat perinteiset muistiinpanovälineet ainakin osittain. Kämmentietokone on kätevä ja helposti mukana kulkeva laite, jonka avulla tietoa on helppo muokata ja jakaa, mikä tekee siitä ihanteellisen apuvälineen niin opettajille kuin oppilaillekin.

5.2.1. Hearts

Hearts (Harness for Education And Research Testing System) on osa kehitystyötä, joka pyrkii tuomaan kämmentietokoneita ja tietotekniikkaa oppilaiden avuksi sekä lisäämään oppilaiden yhteistoiminnallisuutta, osallistumista ja kokeilunhalua [Jipping *et al.*, 2001]. Heartsin avulla oppilaat pystyvät testaamaan sekä arvioimaan erilaisia Java -ohjelmointikielen luokkia ja mittaamaan niiden eri ominaisuuksia. Sovelluksen avulla oppilaat voivat myös jakaa Javan luokkia, kommentoida niitä ja keskustella toistensa kanssa.



Kuva 9. Heartsin näyttöruutu [Jipping *et al.*, 2001].

Oppilaat voivat ladata Heartsin avulla erilaisia Javan luokkia ja määrittää kuinka monta kertaa luokan toimintaa testataan ja mitä luokan ominaisuuksia mitataan. Tämän jälkeen testi voidaan käynnistää ja tulokset tallentuvat erilliseen tiedostoon, josta niitä voidaan jälkepäin tutkia. Samoja luokkia voidaan testata myös pöytätietokoneilla tai muilla Javaa tukevilla laitteilla ja verrata sitten tuloksia keskenään. Käytön aikana oppilaat huomasivat, että kämmentietokoneet ovat varteenotettavia apuvälineitä ja suorituskykyisiä laitteita.

Hearts on mielestäni askel kohti uudenlaista oppimista, koska tällä tavoin oppilaat pääsevät konkreettisesti kokeilemaan ja mittaamaan mahdollisesti myös omien ohjelmointituotostensa tehokkuutta. Ohjelmakoodin optimointi on erittäin tärkeää ja tulevaisuudessa sen merkitys varmasti korostuu yhä enemmän, koska pienet ja kohtuullisen tehottomat mobiililaitteet lisääntyvät koko ajan. Tällä tavoin ohjelmointia olisi myös varmasti paljon mukavampaa opetella kuin itsekseen.

5.2.2. Geney

Geney on sovellus, jonka avulla oppilaat voivat tutkia perinnöllisyyden perusasioita risteyttämällä eräänlaisia virtuaalikalvoja. Sovelluksen avulla oppilaat kykenevät tekemään yhteistyötä ja ratkomaan ongelmia yhdessä. Toiminta on ongelmalähtöistä eli oppilaille annetaan jokin tietty tilanne, joka heidän pitää saada aikaan erilaisten risteytysten avulla [Danesh *et al.*, 2001].

Oppilailla on kämmentietokoneissaan virtuaaliakvaario, joka sisältää perimältään erilaisia kaloja. Oppilaat voivat vaihdella akvaarioidensa kaloja keskenään siirtämällä niitä infrapunayhdeyden avulla kämmentietokoneesta toiseen. Tämän jälkeen kalat voivat sopivassa ympäristössä taas risteytyä

toistensa kanssa ja syntyy perimältään uudenlaisia kaloja, joiden sukupuuta ja perimää voidaan tarkastella. Risteytysten avulla pyritään siis saamaan aikaan haluttuilla ominaisuuksilla varustettu kala ja opitaan samalla kuinka eri geenit vaikuttavat toisiinsa.



Kuva 10. Oppilaat siirtävät kaloja laitteesta toiseen [Danesh *et al.*, 2001].

Geneyn avulla perinnöllisyyden tutkiminen on varmasti paljon havainnollisempaa ja monipuolisempaa tavalliseen opetukseen verrattuna. Tämä osoittaa senkin, että pelejä pelaamalla voi todella oppia jotain ja vielä paljon paremmin kuin muilla menetelmillä. Geney on suunnattu ala-aste ikäisille oppilaille, jotka pitivät siitä ja jopa hiljaisemmatkin oppilaat ottivat osaa toimintaan. Tämänkaltaiseen opetukseen pitäisi mielestäni panostaa paljon enemmän, koska onko mitään parempaa kuin se, että oppiminen on samalla hauskaa ja hyödyllistä.

5.3. Erikoistilanteisiin suunnattuja sovelluksia

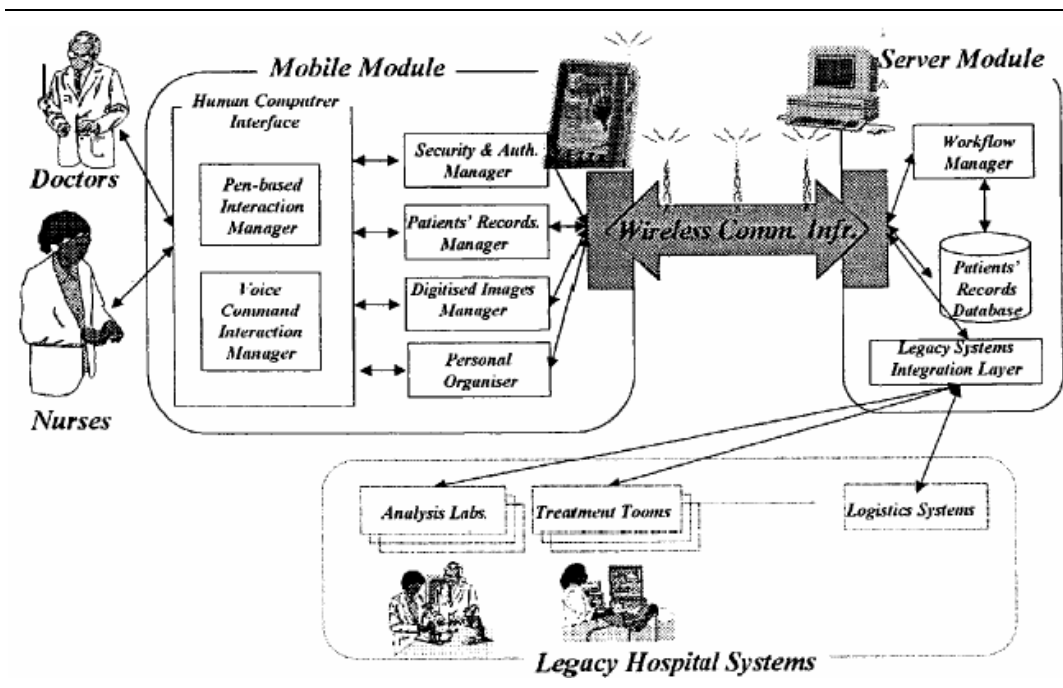
Kämmmentietokonetta voidaan käyttää moneen eri tarkoitukseen ja monessa eri paikassa. Sen käyttö ei rajoitu pelkästään perinteisiin tai koulussa käytettäviin (ryhmätuki)sovelluksiin vaan siitä voi olla apua myös paikoissa, joissa tavallisilla tietokoneilla ei ole ollut käyttöä tai käyttö on ollut liian hankalaa. Kämmmentietokoneita on jo otettu käyttöön sairaaloissa ja ainakin testattu muissakin työympäristöissä.

5.3.1. Ward-in-hand

Ward-in-hand on nimensä mukaisesti suunniteltu sairaalaympäristöön, mutta sitä ei ole vielä otettu käyttöön. Sovelluksen tavoitteena on käyttää tavallisia kämmmentietokoneita ja mahdollisimman standardisoituja tiedostomuotoja, jotta järjestelmä toimisi mahdollisimman varmasti ja kustannukset pysyisivät alhaisella tasolla [Ancona *et al.*, 2000]. Järjestelmään on suunniteltu myös

puheohjaus ja tietoturvakin on otettu huomioon, koska sairaalaympäristössä liikkuu paljon luottamuksellista tietoa.

Sovelluksen pääideana on se, että hoitajat ja lääkärit kantavat mukanaan kämmentietokoneita, joiden avulla he pääsevät käsiksi esimerkiksi potilas-tietoihin ja muihin tietokantoihin, jotka sijaitsevat palvelimella. Kämmentietokoneen avulla voi nopeasti hakea potilaan röntgenkuvat tai potilas-kertomuksen ilman, että niitä tarvitsee kaivaa esiin mapeista tai laatikoista. Kaikki toiminnot toimivat joko puheohjauksella tai kynän ja kosketusnäytön avulla.



Kuva 11. Ward-in-handin arkkitehtuuria [Ancona et al., 2000].

Tämänkaltaisia sovelluksia kehitetään myös Yhdysvaltain armeijan lääkäreiden käyttöön niin sairaaloihin kuin kenttäkäyttöön. Englannissa Liverpoolin naistensairaalassa langattoman verkon avulla on siirrytty jopa osittain paperittomaan tietojenkäsittelyyn. Ward-in-hand projekti saa tukea EU:lta ja sen kehittämiseen on sitoutunut kolme sairaalaa, yrityksiä ja yliopistoja [Ancona et al., 2000]. Kämmentietokone soveltuu hyvin siis myös käytännön toimiin, eikä pelkästään vain muistiinpanojen ja kalenterien ylläpitoon.

5.3.2. Teknikkojen apuri

Kämmentietokone voi olla myös erilaisten teollisuuden työntekijöiden apuna, kuten seuraavaksi esiteltävä projekti todistaa. Järjestelmä on vielä prototyyppi-asteella, mutta sen toimintaa on kuitenkin jo testattu. Sen ideana on tuoda

kämmmentietokone erilaisten asentajien ja teollisuuden työntekijöiden käyttöön kiinnittämällä laite käyttäjän käsivarteeseen.

Kämmmentietokoneet sisältäisivät puhelimen ja langattoman tiedonsiirron lisäksi myös paikannusjärjestelmän. Järjestelmän avulla käyttäjät voisivat saada työtehtävänsä sen mukaan missä päin tehdasta tai rakennusta he ovat; sijainti tunnistettaisiin rakennuksen eri osiin ja laitteisiin upotettujen pienten radiolähettimien avulla. Eri osia ja laitteita osoittamalla voitaisiin myös saada aikaan vuorovaikutusta, kuten kämmmentietokoneen näytölle ilmestyvät kyseisen komponentit tiedot ja mahdollisuus soittaa valmistajalle jne. Käyttäjät voisivat myös jättää elektronisia muistilappuja eri laitteisiin, joissa selitettäisiin mitä on tehty tai mikä laitteessa mahdollisesti on vialla [Fallman, 2003].



Kuva 12. Kämmmentietokone kiinnitettynä käsivarteeseen [Fallman, 2003].

Järjestelmän avulla työnteosta tulisi paljon kätevämpää, koska eri laitteiden tietoja ei tarvitsisi kaivaa esiin ohjekirjoista ja laitteet voisivat itse sisältää ohjeistusta elektronisessa muodossa. Työnteosta tulisi myös organisoituneempaa, koska työntekijöitä voitaisiin koordinoida yhdestä paikasta ja uusia tehtäviä voitaisiin jakaa reaaliaikaisesti kentällä oleville työntekijöille. Myös työntekijöiden keskinäistä kommunikointia voitaisiin tämänkaltaisella järjestelmällä huomattavasti parantaa.

6. Lopuksi

Kuten esitellyistä sovelluksista voidaan todeta, on kämmmentietokoneiden käyttöä uusissa ympäristöissä ja uusiin tarkoituksiin ainakin suunniteltu, mutta toteutukset ovat vielä jääneet aika vähälle. Kämmmentietokoneisiin ja niiden sovelluksiin tulisi panostaa paljon nykyistä enemmän, koska kämmmentietokoneet ja langaton tiedonsiirto yleistyvät ja kehittyvät koko ajan.

Nykyiset sovellukset osoittavat jo pienen osan kämmmentietokoneen käyttömahdollisuuksista, mutta ovat vielä monilta osin puutteellisia, kuten

tietoturvan osalta. Niissä on kuitenkin jo monia hyviä ja innovatiivisiakin ratkaisuja, joita toivottavasti kehitetään jatkossa yhä paremmiksi ja tehokkaammiksi. Varsinkin Geney -sovellus osoittaa todella hyvää ja uutta lähestymistapaa uusien asioiden opettamiseen, oppimiseen ja ryhmätyöskentelyyn.

Yhteistoiminnallisuus on tulevaisuuden tärkein työskentelymuoto ja kämmentietokoneilla on suuri osa sen kehityksessä, koska ne kulkevat aina mukana ja niiden mahdollisuudet ovat miltei rajattomat.

Lähteet

- [Ancona *et al.*, 2000] M. Ancona, G. Dodero, F. Minuto, M. Guida and V. Gianuzzi, Mobile computing in a hospital: the WARD-IN-HAND project. *Proceedings of the 2000 ACM symposium on Applied computing*, March 2000, 554-556.
- [Bartram and Blackstock, 2003] L. Bartram and M. Blackstock, Designing portable collaborative networks. *ACM Queue*, 1 (3), May 2003, 40-49.
- [Buszko *et al.*, 2001] D. Buszko, W. Lee and A. Helal, Decentralized ad-hoc groupware API and framework for mobile collaboration. *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, 2001, 5-14.
- [Danesh *et al.*, 2001] A. Danesh, K. Inkpen, F. Lau, K. Shu and K. Booth , GeneyTM: designing a collaborative activity for the palmTM handheld computer. *Proceedings of the SIGCHI conference on Human factors in computing systems*, March 2001, 3 (1), 388-295.
- [Davis *et al.*, 1999] R.C. Davis, J.A. Landay, V. Chen, J. Huang, R.B. Lee, F.C. Li, J. Lin, C.B. Morrey, B. Schleimer, M.N. Price and B.N. Schilit, NotePals: lightweight note sharing by the group, for the group. *Proceedings of the SIGCHI conference on Human factors in computing systems*, May 1999, 338-345.
- [Fallman, 2003] D. Fallman, Enabling physical collaboration in industrial settings by designing for embodied interaction. *Proceedings of the Latin American conference on Human-computer interaction*, August 2003, 41-51.
- [Jipping *et al.*, 2001] M.J. Jipping, J. Krikke, S. Dieter and S. Sandro, Using handheld computers in the classroom: laboratories and collaboration on handheld machines. *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*, 33 (1), February 2001, 169-173.

- [Landay, 1999] J.A. Landay, Using note-taking appliances for student to student collaboration. *Proceedings of the 29th ASEE/IEEE Frontiers in Education Conference*, November 1999. Also available as www.cs.berkeley.edu/~landay/research/publications/fie99.pdf.
- [Nishimura *et al.*, 1998] T. Nishimura, H. Yamaki, T. Komura and T. Ishida, Community viewer: visualizing community formation on personal digital assistants. *Proceedings of the 1998 ACM symposium on Applied Computing*, February 1998, 433-438.
- [Roth and Unger, 2001] J. Roth and C. Unger, Using handheld devices in synchronous collaborative scenarios, *Personal and Ubiquitous Computing*, **5** (4), January 2001, 243-252.
- [Tang *et al.*, 2001] J.C. Tang, N. Yankelovich, J. Begole, M. Van Kleek, F. Li and J. Bhalodia, ConNexus to awarenex: extending awareness to mobile users. *Proceedings of the SIGCHI conference on Human factors in computing systems*, **3** (1), March 2001, 221-228.
- [Wiberg, 2001] M. Wiberg, RoamWare: an integrated architecture for seamless interaction in between mobile meetings. *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, September 2001, 288-297.

Lapsille suunnattujen verkkosivujen käytettävyys

Matleena Koivisto

Tiivistelmä.

Tässä tutkielmassa tutustutaan lapsille suunnattujen verkkosivujen suunnitteluperiaatteisiin. Tarkoituksena on kerätä suunnitteluohjeet, jotka auttavat suunnittelemaan juuri lapsille sopivia, hyvän käytettävyyden omaavia verkkosivuja. Lisäksi tutkielmassa arvioidaan kolmea tällä hetkellä Internetistä löytyvää lapsille suunnattua verkkosivustoa, ja tutustutaan noin 8-12-vuotiaiden lasten mielipiteisiin ja kokemuksiin sekä lapsille suunnatuista verkkosivuista että Internetistä yleensä.

Avainsanat ja -sanonnat: Verkkosivu, käyttöliittymä, lapset, käytettävyys.

CR-luokat: H.5.2

1. Johdanto

Muutaman viime vuosikymmenen aikana on tietokone ja Internet tullut osaksi yhä suuremman ihmisjoukon jokapäiväistä arkea. Tietokoneiden tehot ovat kasvaneet, kustannukset ovat laskeneet ja tiedon siirtämisessä käytettävä tekniikka on parantunut [Preece *et al.*, 1994]. Yhä useammat yritykset, yhteisöt ja yksityiset ihmiset julkaisevat jatkuvasti uusia verkkosivuja. Internet on jo nyt kasvanut valtavan suureksi; erilaisia sivustoja on satoja miljoonia.

Nykyään arviolta jo kymmenet miljoonat lapsetkin ympäri maailmaa käyttävät Internetiä, ja selailevat sekä lapsille että aikuisille suunniteltuja verkkosivuja [Fishler, 1998]. Heistä useat ovat aloittaneet tietokoneen käytön ainakin jossain määrin jo muutaman vuoden ikäisenä. Sama kehitys näyttäisi jatkuvan vielä tulevaisuudessakin; lapsikäyttäjien määrän uskotaan jatkossa kasvavan useilla miljoonilla joka vuosi [Nielsen, 2002]. Samalla kun tietokoneet ja Internet leviävät yhä laajemmalle, on lapsillakin yhä paremmat mahdollisuudet vieraillla Internetissä.

Koska lasten osuus Internetin käyttäjistä on kasvanut valtavaa vauhtia, on myös lapsille suunnattujen verkkosivujen määrä kasvanut. Perinteisten opetussivujen rinnalle on monien yritysten ja laitostenkin sivuille ilmestynyt varta vasten lapsille tarkoitettuja osioita [Nielsen, 2002]. Sivuille tarjotaan kaiken ikäisille lapsille sekä tietoa ja neuvoja että lukemattomia erilaisia huvituksia.

Lapset ovat verkkosivustojen käyttäjinä aivan erilainen kohderyhmä kuin aikuiset. Heitä ei voi ajatella vain pienikokoisina aikuisina, vaan heillä on omat toiveensa ja tarpeensa [Druin and Solomon, 1996]. Lapsille suunnattujen verkkosivujen suunnittelun lähtökohtana ei voi olla pelkästään perinteiset näkemykset aikuiskäyttäjille suunnittelusta, vaan lapsikäyttäjien omat erityistarpeet tulee asettaa etusijalle. Verkkosivujen suunnittelussa ja arvioinnissa ei siis kannata vain luottaa aikuisten ajatuksiin siitä, mitä lapset haluavat, vaan lasten oman panoksen saaminen mukaan on erittäin tärkeää [Druin *et al.*, 1998].

Lapsikäyttäjäkunnan kasvusta huolimatta nykyään tiedetään vielä varsin vähän siitä, kuinka lapset todella käyttävät nyt jo olemassa olevia verkkosivuja, ja kuinka sivustoja tulisi heitä varten suunnitella [Nielsen, 2002]. Käytettävyytystutkimus on keskittynyt lähinnä aikuisiin, ja heille suunniteltuihin sivuihin, eikä lasten mahdollisista omista erityistarpeista tiedetä paljoakaan [Sullivan *et al.*, 2000].

Tutkielman tavoitteena onkin laatia jo olemassa olevan tutkimustiedon pohjalta ohjeita lapsille suunnattujen verkkosivujen suunnittelua varten, ja näin parantaa niiden käytettävyyttä. Näitä suunnitteluperiaatteita voivat hyödyntää tietysti myös laajemmalle käyttäjäkunnalle tarkoitettujen verkkosivujen suunnittelijat. Lisäksi tutkielmassa kartoitetaan 8-12-vuotiaiden lasten Internetin käyttöä, ja arvioidaan kolmea jo verkosta löytyvää lapsille suunnattua verkkosivustoa.

2. Verkkosivujen käytettävyys

2.1. Käytettävyyden merkitys

Tietokoneiden käytön leviämisen myötä myös käytettävyyden merkitys on kasvanut. Ohjelmat ovat käyneet laajemmiksi ja monimutkaisemmiksi. Tietokoneita ei enää käytetäkään vain tiedon esittämiseen ja prosessointiin, vaan ihmiset hoitavat arkielämän asioitaan ja työtehtäviään vuorovaikutuksessa tietokoneen kanssa. Suurin osa käyttäjistä ei enää olekaan alansa eksperttejä kuten 1970-luvulla [Preece *et al.*, 1994].

Käytettävyydellä tarkoitetaan yleensä tietokonejärjestelmien suunnittelua siten, että ne tukevat ihmisiä heidän haluamansa tehtävän tuottoisassa ja turvallisessa tekemisessä [Preece *et al.*, 1994]. Järjestelmän käyttämisen tulee olla tehokasta ja sen pitää toimia oikein [Preece *et al.*, 1994]. Koska virheitä kuitenkin joskus sattuu, järjestelmän pitää huomata ja korjata ne mahdollisimman hyvin [Brinck *et al.*, 2002]. Käyttämisen tulee olla miellyttävää ja nopea oppia [Brinck *et al.*, 2002]. Käyttäjällä ei pitäisi myöskään olla tarvetta

myöhemmin opetella järjestelmän käyttöä uudestaan, vaan käytön tulee olla helposti muistettavaa [Brinck *et al.*, 2002].

Hyvä käytettävyys on ensiarvoisen tärkeää huomioida kaikkia tietokoneohjelmia suunniteltaessa. Vaikka ohjelman toiminnallisuus olisi kuinka hienoa tahansa, se on täysin epäonnistunut, jos sitä on käytännössä mahdoton käyttää. Hyvä käytettävyys pitää käyttäjät tyytyväisinä ja tuotteelle uskollisina. Se taas tuottaa yrityksille enemmän rahaa [Preece *et al.*, 1994].

2.2. Käytettävyys verkkosivuja suunniteltaessa

Erityisesti verkkosivuja suunniteltaessa on käytettävyyden merkitys kasvanut suuresti. Nykyään on erilaisia sivustoja Internetissä niin suuri määrä, että käyttäjällä on varaa valita monen eri vaihtoehdon väliltä. Jos jokin verkkosivu ei miellytä käyttäjää jostain syystä, tai jos haluttua tietoa on hidas löytää, käyttäjä vain yksinkertaisesti siirtyy toiselle sivustolle [Nielsen, 2000].

Parhaita keinoja houkuttaa käyttäjät sivuilleen onkin suunnitella sivut hyvän käytettävyyden periaatteiden mukaisesti. Verkkosivustolla pitää käyttäjien haluamien tehtävien tekemisen olla helppoa ja tehokasta [Nielsen, 2000]. Vain käytettävillä sivuilla vierailaan [Nielsen, 2000].

Jotta verkkosivut saataisiin miellyttäväksi käyttää, tulisi suunnittelun ja arvioinnin jokaisessa vaiheessa ajatella ihmistä. Sivusto pitää suunnitella niin, että se sopii ihmisen tapaan havainnoida ja tehdä tehtäviä. Myös ihmisen muistin aiheuttamat rajoitukset tulee ottaa huomioon. Esimerkiksi sivuston linkit kannattaa suunnitella sellaisiksi, että ne muistamisen sijasta on helppo tunnistaa [Brinck *et al.*, 2002].

Lapsille suunnattujen verkkosivujen suunnittelussa on käytettävyyden merkitys vielä tavallistakin suurempi. Lapset nimittäin kärsivät aikuisia herkemmin erilaisista käytettävyysongelmista [Nielsen, 2002]. Erityisesti erilaisista virhetilanteista toipuminen on lapsilla huomattavasti aikuista hankalampaa [Nielsen, 2002]. Huono käytettävyys yhdistettynä lasten huonoon kärsivällisyyteen johtaakin monien poistumiseen sivuilta [Nielsen, 2002].

2.3. Käytettävyyden arviointi

Verkkosivujen käytettävyyttä voidaan arvioida monella eri tavalla. Arvioinnissa voidaan käyttää apuna erilaisia sivuston suunnittelun ohjeita ja sääntöjä. Käyttäjää voidaan haastatella tai heitä voidaan tarkkailla sivustoa käyttämässä. Sivustoa voi myös arvioida tekemällä tarkkaan suunniteltuja käytettävyydestejä. Yhteistä kaikille arvioinneille on kuitenkin se, että pyritään selvittämään, mitkä ovat kohderyhmään kuuluvan käyttäjän ominaisuudet, mitkä ovat ne tehtävät, jotka käyttäjä haluaa tehdä ja miten niiden tehtävien tekeminen onnistuu.

Heuristinen arviointi on yksi suosituimmista tavoista arvioida erilaisten käyttöliittymien käytettävyyttä [Nielsen, 1994]. Arviointi suoritetaan käymällä läpi lyhyttä listaa hyvän käytettävyyden ominaisuuksista, ja arvioimalla niiden toteutumista sivuilla.

Yksi heuristisen arvioinnin käytetyimmistä hyvän käytettävyyden ominaisuuslistoista on Jakob Nielsenin [1994] lista Ten Usability Heuristics:

1. järjestelmän tilan näkyvyys,
2. yhteensopivuus järjestelmän ja todellisen maailman välillä,
3. käyttäjän hallinta ja vapaus,
4. yhdenmukaisuus ja standardi,
5. virheiden ehkäisy,
6. tunnistaminen muistamisen sijasta,
7. käytön tehokkuus ja joustavuus,
8. esteettinen ja yksinkertainen suunnittelu,
9. käyttäjän auttaminen virheiden huomaamisessa, tunnistamisessa ja niistä toipumisessa ja
10. ohjeet ja dokumentointi.

Eryteisesti verkkosivujen arviointiin on olemassa myös omia listoja. Useat niistä on muokattu Nielsenin listan pohjalta. Yksi hyödyllinen lista löytyy kirjasta Usability for the Web: Designing Web Sites That Work [Brinck *et al.*, 2002]:

1. sivuston sisältö ja kattavuus,
2. nopeus,
3. navigaatio,
4. tehtävän tarkoituksenmukaisuus,
5. visuaalinen suunnittelu,
6. yhteensopivuus,
7. yksinkertaisuus,
8. johdonmukaisuus,
9. virheidenhallinta ja
10. käyttäjän kunnioittaminen.

Molempia listoja hyödynnetään tutkielmassa lapsille suunnattujen esimerkkisivustojen arvioinnissa.

3. Arvioitavat verkkosivustot

Tutkielmassa arvioidaan kolmea hieman erityyppistä lapsille suunnattua verkkosivustoa. Kaikki sivustot on tarkoitettu noin 8-12-vuotiaille lapsille, ja ne löytyvät Internetistä melko helposti joko monien linkkilistojen tai

hakukoneiden avulla. Sivustoista kaksi on suomenkielisiä ja yksi englanninkielinen.

3.1. Särkänniemen sivut lapsille

Tampereella sijaitsevalla Särkänniemen huvipuistolla on Internetissä melko laajat sivut, joilta löytyy asiaa sekä aikuisille että lapsille. Sivuilla on mm. tietoa huvipuistosta, sen historiasta, aukioloajoista ja palveluista. Lapsille sivustolla on aivan omat sivunsa: Särkkis, virtuaalihuvipuisto.

Lasten sivujen sisältö on runsasta ja antoisaa. Niille on kerätty monenlaisia eri ikäisille lapsille suunnattuja pelejä ja opetusosioita, jotka liittyvät hienosti Särkänniemen huvipuiston kohteisiin, kuten planetaarioon, akvaarioon, lasten eläintarhaan ja delfinaarioon. Sivuilla on myös paljon lasten sosiaalisuutta ja luovuutta tukevia tehtäviä. Lapset voivat lähettää sähköisiä kortteja, kirjoittaa toisilleen tervehdyksiä tai vaikkapa tulostaa itselleen värityskuvia. Monet tehtävistä ja peleistä ovat kiinnostavia ja haastaviakin, ainakin nuoremmille lapsille. Sivuja myös päivitetään melko usein, joten sisältö on ainakin osittain tuoretta, ja innostaa vierailemaan sivuilla useinkin.

Sivuston visuaalinen ilme vaihtelee eri sivuilla runsaastikin. Akatemiaosuuden sivuille mentäessä voi ensin jopa luulla, että on vahingossa poistunut Särkkiksen sivuilta kokonaan. Visuaalinen suunnittelu voisi olla myös yksinkertaisempaa. Jotkin elementit ovat tehtävien tekemisen kannalta aivan tarpeettomia ja saattavat jopa häiritä niiden suorittamista. Särkkiksen pääsivun pohjakuvana olevan ruudukon voisi esimerkiksi poistaa kokonaan, jolloin sivusta tulisi rauhallisempi ja yksinkertaisempi.



Kuva 1. Särkkiksen pääsivu.

Visuaalisen ilmeen mukana vaihtuu myös navigointityyli. Yhdessä osiossa opittu navigointitapa ei enää toimikaan seuraavassa, vaan sitä varten pitää opetella erilainen tapa tehdä tehtäviä. Tämä on melko turhauttavaa ja vie liian paljon aikaa. Sivujen eri osiot eivät myöskään ole arkkitehtuuriltaan samanlaisia, ja käyttäjän onkin vaikea välillä tiedostaa missä päin sivustoa hän on. Akatemian sivut aukenevat lisäksi uuteen ikkunaan, joka saattaa hämätä varsinkin kokemattomampia Internetin käyttäjiä. Jos käyttäjä ei nimittäin huomaa uuden ikkunan aukenemista, hän saattaa joutua pulaan yrittäessään palata takaisin pääsivulle. Sinnehän ei pääse uuden selainikkunan peruutus-painikkeen avulla.

Navigointipainikkeet on pääsivuilla suunniteltu hyvin. Ne ovat kooltaan riittävän isoja, jotta pienemmätkin lapset onnistuvat osumaan niihin hiirellä. Niiden paikka myös säilyy samassa kohtaa eri sivuilla, Akademia-sivuja lukuun ottamatta. Joillakin sivuilla esiintyy kuitenkin Delfinikouluttaja-linkki eri näköisenä kahdessa eri kohdassa sivua. Käyttäjä ei tiedä kokeilematta, että kyseessä on sama linkki. Tällöin erityisesti pienet lapset eivät välttämättä huomaa jo olleensa samalla sivulla.

Multimedian mahdollisuuksia sivuilla ei ole hyödynnetty parhaalla mahdollisella tavalla. Erityisesti äänet puuttuvat valtaosasta sivuja. Niiden mukaan tuominen lisäisi varmasti käyttäjien tyytyväisyyttä. Erityisesti eläinsivuilla olisi mukava saada eläinten ääniä mukaan.

Peliluolan peleissä on jonkin verran ongelmia. Ne latautuvat muita sivuja huomattavasti hitaammin, eivätkä niiden ohjeet aina toimi kunnolla. Myöskään virhetilanteissa käyttäjälle ei tarjota tietoa tilanteesta tai opastusta virheen korjaamiseksi.

3.2. Pelastustoimen sivut lapsille

Sisäasiainministeriön pelastusosaston verkkosivuilta löytyy useita lapsille suunnattuja sivuja. Niillä esitellään pelastustoimen tehtäviin ja paloturvallisuuteen liittyviä tietoja erilaisten pelien ja satujen avulla. Sivut on nähtävästi tarkoitettu melko pienille lapsille, vaikka osa peleistä vaatiikin vasta-alkajaa kokeneemman tietokoneen käyttäjän.

Lasten sivujen löytäminen pelastustoimen sivuilta ei ole aivan yksinkertaista. Varsinkin nuoremmat lapset tarvitsevat aikuisten apua sivujen löytämisessä. Peleille kun ei ole tehty omaa pääsivua, vaan linkit niihin ovat muiden, aikuisille tarkoitettujen linkkien seassa. Pelastustoimen pääsivulla ei myöskään ole minkäänlaista mainintaa lapsille suunnatuista sivuista. Lisäksi osa linkeistä jää myös ainakin pienemmällä näytöllä vieritettävän ruudun alareunaan niin, ettei niitä ole mitenkään helppo havaita.



Kuva 2. Pelastustoimen lasten sivujen kuvitusta.

Toiminnallisuuden kanssa sivuilla on ongelmia. Osa peleistä on kankeita ja vaivalloisia käyttää. Esimerkiksi vesitykin kanssa loppuu käyttäjältä mielenkiinto nopeasti liian hankalan ja virheille alttiin suuntaustavan takia. Osa

peleistä ei myöskään toimi kunnolla. Minkäänlaisia virheilmoituksia ei kuitenkaan anneta käyttäjälle missään vaiheessa. Myös pelien ohjeet ovat puutteelliset.

Visuaalisesti pelit ja tarinat ovat pääosin miellyttäviä ja hauskoja, joskin hieman vanhanaikaisen oloisia. Osa kuvista näyttää keskeneräisiltä ja harrastelijamaisilta. Häätäpuhelun puhelimesta on jopa unohtunut nollanäppäin kokonaan pois. Äänen käyttöä voisi myös lisätä reilusti; osasta sivuja ne puuttuvat kokonaan. Varsinkin hälytysajoneuvojen esittelysivuilla ne kyllä kuuluisivat asiaan.

Luovuuteen tai sosiaalisuuteen innostavia elementtejä sivuilla ei ole ollenkaan. Lapsille ei muutenkaan juuri anneta mahdollisuutta toteuttaa tehtäviä tai pelata pelejä omalla tavallaan, vaan kaikki toiminnallisuus tapahtuu tiukasti suunnittelijan määräämällä tavalla. Tällöin lapset menettävät nopeasti mielenkiintonsa sivuihin, eivätkä luultavasti enää palaa sivuille toistamiseen. Sivuja ei myöskään päivitetä ollenkaan, joten niille ei ole mitään uutta annettavaa toisella käyntikerralla.

3.3. International Children's Digital Library

Lasten kansainväliseen digitaaliseen kirjastoon pyritään keräämään seuraavan viiden vuoden aikana yli 10 000 sadalla eri kielellä kirjoitettua kirjaa digitaalisessa muodossa. Kirjat edustavat erilaisia kulttuureja, yhteiskuntia ja elämäntapoja ympäri maailmaa. Niiden avulla suunnittelijat haluavat auttaa 3-13-vuotiaita lapsia ymmärtämään sitä maailmaa, jossa he elävät. Tällä hetkellä kirjoja on saatavilla jo satoja.

Tarkoituksena on ollut suunnitella sivusto, jossa kirjojen hakeminen tapahtuu visuaalisesti perinteisen tekstihaun sijasta [SearchKids]. Kirjoja voi hakea kirjastosta joko erilaisten kategorioiden avulla tai sitten valitsemalla hiiren avulla näytöllä olevasta maapallosta haluamansa maanosa. Hakemisen apuna käytetään mahdollisuutta valita etsintätuloksesta pienempi alue, joka sitten suurennetaan lähempää tarkastelua varten. Tällä tavalla käyttäjä pysyy paremmin perillä siinä, missä kohtaa sivustoa ollaan ja miten hakua voisi tarkentaa [SearchKids]. Suurin osa sekä aikuis- että lapsikäyttäjistä piti tätä hakutapaa perinteistä tapaa parempana [SearchKids]. Sivuilla on myös mahdollista hakea kirjoja perinteisen sanahaun perusteella. Jokaista kirjastossa olevaa kirjaa voi lukea tietokoneen näytöltä sivu kerrallaan. Sivuilta löytyy myös suomenkielisiä kirjoja.



Kuva 3. ICDL:n perusversion etusivu.

Sivujen visuaalinen ilme on rauhallinen ja yksinkertainen. Kirjojen hakemisen kannalta turhia visuaalisia elementtejä ei juurikaan ole. Kuvituksena toimii kirjojen hakuun liittyvät ikonit, kirjastosta löytyvien kirjojen kannet ja sivut sekä navigointipainikkeet.

Navigointi sivuilla tapahtuu pääasiassa valitsemalla kuvaikoneita hiiren avulla. Ikonit ovat riittävän isokokoisia, jotta pienenkin lapsen on helppo osua niihin. Ikonit on myös suunniteltu niin, että niiden sisältö antaa lapselle viitteitä linkin sisällöstä. Esimerkiksi ikoni, jossa on kuvattu sekä ohut että paksu kirja, on linkkinä sellaiselle sivulle, jossa kirjat on järjestetty pituuden mukaan. Myös sellaisten lasten jotka eivät osaa englantia, on ikonien avulla mahdollista käyttää kirjaston sivuja. Lisäksi sivuilla on näkyvästi esillä paluu-painike, jonka avulla pääsee helposti takaisin edelliselle sivulle.

Äänet sivuilta puuttuvat kokonaan. Kun kohderyhmään kuuluu kuitenkin varsin pieniäkin lapsia, voisi äänen käyttö navigaation apuna olla hyödyllistä. Animaatioitakaan sivuilta ei löydy ollenkaan, mutta eivät ne oikein kuulukaan kirjastosivuston luonteeseen. Sosiaalisuutta sivuilla on toteutettu siten, että lasten on mahdollista etsiä ja lukea kirjoja yhdessä.

4. Kysely verkkosivujen käytettävyydestä

Kostian kouluun Pälkäneellä jaettiin 30 kyselylomaketta Internetin käytöstä ja verkkosivujen käytettävyydestä (Liite 1). Kysely oli suunnattu 2.-6. luokkalaisille, eli noin 8-12-vuotiaille lapsille. Kyselylomakkeista puolet jaettiin

tytöille ja puolet pojille, koska lapsilla sukupuolierot Internetin käytössä ovat suuremmat kuin aikuisilla [Nielsen, 2002]. Kyselyistä palautettiin 17 kappaletta. Niistä 10 oli tytöiltä ja 7 pojilta. Vastajat olivat iältään 8-13-vuotiaita.

4.1. Kyselyn suunnittelu

Kyselylomakkeissa kysyttiin vastaajilta muutamia taustatietoja: ikä, sukupuoli, onko kotona tietokone ja Internet-yhteys, miten usein käy Internetissä ja kuinka usein siellä on kerrallaan. Varsinaisessa kysymysoiossa pyydettiin vastaajia listaamaan muutama verkkosivusto, joilla usein vierailee. Heitä pyydettiin myös kertomaan, miksi vierailevat näillä sivuilla usein. Lisäksi pyydettiin vastaajia hieman arvioimaan kyseisen sivuston hyviä ja huonoja puolia, sekä vierailemaan tutkielmassa arvioitavilla sivustoilla ja kertomaan niistä mielipiteensä.

Kyselylomakkeessa käytetyt kysymykset oli laadittu niin, että ne ohjaisivat vastaajaa mahdollisimman vähän. Lomakkeen avulla pyrittiin keräämään tietoa siitä, miten lapset itse kokevat verkkosivujen selailemisen, minkälaisiin asioihin he itse kiinnittävät sivuilla huomiota ja millaisina he näkevät tutkielmassa arvioitavat sivustot. Tarkoituksena oli saada kyselyn avulla ideoita lapsille suunnatun verkkosivuston suunnittelua varten. Avoimet, haastattelunomaiset kysymykset ovat hyvä tapa saada osviittaa sivujen tarkempaa arviointia varten [Brinck *et al.*, 2002].

4.2. Lasten käyttämät verkkosivustot

Lähes kaikki verkkosivut, joilla lapset usein vierailevat, ovat joko aikuisille tai nuorille suunnattuja. Edes 8-vuotiaat vastaajat eivät vieraile mielellään lapsilla suunnatuilla verkkosivuilla.

Erilaiset hakukoneet sekä tv-ohjelmien ja -kanavien sivut ovat vastaajien keskuudessa käytetyimpiä. Seuraavaksi eniten lapset vierailevat erilaisilla peli- ja harrastussivuilla. Erityisesti pojat etsivät Internetistä eniten peleihin liittyviä sivustoja. Tytöt taas ovat enemmän kiinnostuneita eläinharrastuksiin liittyvistä sivustoista. Myös sosiaalinen kanssakäymisen mahdollistavat verkkosivustot kiinnostavat vastaajia kovasti. Varsinkin tytöt käyvät usein lukemassa sähköpostinsa. Jopa yhdellä kaikkein nuorimmista vastaajista on sähköposti ahkerassa käytössä. He myös lähettävät mielellään sähköisiä postikortteja toisilleen. Vanhemmat lapset käyvät myös jonkin verran erilaisilla keskustelupalstoilla.

Ylivoimaisesti eniten lapsia harmittavat sivujen pitkät latautumisajat. Lähes kaikki vastaajat toivoivat sivujen toimivan nopeammin. Lisäksi toimimattomat linkit ovat aiheuttaneet monille ongelmia. Myös haluttujen tietojen hidas löytyminen harmittaa monia lapsia. Joidenkin vastaajien mielestä sivut, joissa

on paljon valittavia vaihtoehtoja, ovat hankalia käyttää. Toisaalta taas parin vastaajan mielestä liian vähän vaihtoehtoja sisältävät sivutkin ovat käytön kannalta epämiellyttäviä. Yksi vastaaja ei myöskään pitänyt sivuista, joilla joutuu kirjautumaan sisään käyttäjätunnuksella. Muiden ongelmien lisäksi moni on törmännyt lempisivuillaan myös erilaisiin virhetilanteisiin. Niiden tunnistaminen ja niistä toipuminen vaikuttaa ongelmalliselta.

4.3. Tutkielmassa arvioitavat verkkosivustot

Suurin osa kyselyyn vastanneista pitää kaikkia tutkielmassa arvioitavia sivustoja hieman tylsinä ja auttamattoman lapsellisina. Ainoastaan kaksi vastaajaa ilmoitti käyneensä arvioitavilla sivustoilla ennenkin; toinen oli käynyt lähettämässä sähköisiä postikortteja Särkänniemen huvipuiston sivuilta, ja toinen vieraili usein pelastustoimen lastensivuilla.

Särkänniemen huvipuiston sivut ovat suurimman osan mielestä arvioitavista sivustoista mielenkiintoisimmat. Monet olivat kylläkin Särkkiksen pääsivun perusteella päätelleet sivut lapsellisiksi, ja sen tähden tylsiksi. Vaikka sivustolta löytyy myös hieman vanhemmillekin lapsille suunnattuja osioita, ne taisivat monelta jäädä alkuvaikutelman takia löytämättä. Sivuston yksinkertaisesta navigoinnista annettiin jonkin verran kehuja. Myös kuvat ja väritys ovat monen mieleen. Sivujen suurimpia haittapuolia taas ovat pelien pitkät latausajat. Yleisesti ottaen vastaajien mielestä sivut olivat ihan kivat, mutta he eivät aio vierailla siellä uudestaan.

Pelastustoimen sivuista vastaajat eivät juurikaan pitäneet. Yksi vastaajista oli kuitenkin käynyt sivuilla ennenkin, ja aikoo käydä jatkossakin. Sivujen käyttö koettiin helpoksi, mutta sisältö ja visuaalinen ympäristö köyhäksi. Navigoinnista ei ollut kellekään mitään valittamista, mutta muutama vastaaja oli huomaamattaan käynyt tutustumassa sivujen ulkopuolellakin oleviin sivuihin.

International Children's Digital Libraryn sivut ovat lähes kaikkien vastaajien mielestä aivan liian vaikeat käyttää. Ainoastaan yksi vastaaja kehui sivujen käytettävyyttä, mutta hänen mielestään ne ovatkin sitten oikein todella helpot käyttää. Suurin osa oli törmännyt heti vieraan kielen aiheuttamiin hankaluuksiin. Käytännössä he eivät olleet osanneet edes aloittaa sivuston käyttöä. Lapset ovatkin luultavasti käyttäneet Internetiä ja arvioineet sivustoja itsenäisesti, ilman esimerkiksi vanhempien apua.

5. Verkkosivujen sisällön suunnitteluperiaatteet

5.1. Multimedian käyttäminen

Nykyajan lapset ovat kasvaneet koko elämänsä ajan television ja videopelien tarjoamassa erittäin multimediarikkaassa ympäristössä [Druin *et al.*, 1998]. He

vaativat samaa moninaisuutta myös verkkosivustoilta. Jotta sivusto olisi heidän mielestään kiinnostava ja viihdyttävä, sen pitää sisältää paljon erilaisia multimedian elementtejä [Druin *et al.*, 1998].

Perinteisen lukemisen tai kuuntelemisen sijasta lapset myös oppivat asioita paremmin käyttämällä kaikkia aistejaan [Fishler, 1998]. Suunnittelijoiden tulisikin suunnitella verkkosivut sellaisiksi, että ne voisi pelkän näkemisen lisäksi myös kuulla ja jopa tuntea.

Lapset myös pitävät animaatioista verkkosivuilla, ainakin huomattavasti enemmän kuin aikuiset [Nielsen, 2002]. Liian paljon animaatiota kuitenkin saattaa häiritä halutun tehtävän suorittamista [Bernard, 2003]. Animaatioiden käyttöä tuleekin harkita tarkasti. Multimedia ilman varsinaista tarkoitusta ei paranna verkkosivujen käytettävyyttä.

5.2. Visuaalisuuden suunnitseminen

Lapset etsivät verkkosivuilta lähinnä hauskuutta ja viihdyttäviä elementtejä, joten sivujen tulisi olla leikin omaisia ja niiden tulisi hyödyntää lasten luontaista uteliaisuutta [Bernard, 2003]. Niiden pitää olla värikkäitä ja sisältää paljon multimediaa [Nielsen, 2002]. Sivut eivät kuitenkaan saa olla lapsellisen oloisia. Ne eivät myöskään saa olla alentavan näköisiä tai epäillä lasten älykkyyttä [Druin *et al.*, 1998].

Visuaalinen suunnittelu ei saa olla epäselvää tai vaikeaa tulkita, muuten sivuston sanoma hukkuu median taakse [Druin and Solomon, 1996]. Lapset eivät aina kykene erottamaan varsinaista sisältöä tai pelkkiä viihdyttäviä elementtejä toisistaan. Yksinkertaisuus voittaa monimutkaisuuden [Nielsen, 2000].

5.3. Sisällön suunnitseminen

Lapsille suunnitellulla verkkosivulla tulisi olla korkeatasoinen sisältö, sitä tulisi päivittää usein, sen pitäisi aueta nopeasti ja sitä tulisi olla helppo käyttää [Nielsen, 2000]. Sen pitää olla kiinnostava, hauska ja myös haastava [Bernard, 2003].

Verkkosivusto on lasten mielestä mielenkiintoinen niin kauan kun sen sisältö on innovatiivista ja hauskaa. Siksi sivuilla pitäisi olla usein uusia ominaisuuksia tai lapset yksinkertaisesti eivät enää käy niillä [Bernard, 2003]. Kun lapsille suunnitellaan Internet-ympäristöä, suunnittelijoiden tuleekin myös ottaa huomioon, miten toiminta ja informaatio sivuilla pidetään ajan tasalla [Druin and Solomon, 1996].

Myös vertaisvaikutuksella on suuri merkitys sisällön suunnittelussa. Lapset haluavat sivuilta sitä mikä on "coolia", eli sitä mikä on sillä hetkellä

muodikasta, ja mitä muutkin tekevät [Nielsen, 2002]. Sisältö ei myöskään saa olla missään nimessä lapsellista [Nielsen, 2002].

Pojat haluavat pelata pelejä, tehdä kotisivuja ja ladata ohjelmistoja ja musiikkia [Bernard, 2003]. Tytöt taas lukevat verkosta löytyviä lehtiä, lähettävät e-kortteja ja tekevät kotitehtäviä [Bernard, 2003]. Jotta lapsia saisi houkuteltua sivuille, sieltä pitäisikin löytyä kohderyhmän suosimaa toimintaa.

6. Verkkosivujen toiminnallisuuden suunnitteluperiaatteet

6.1. Sosiaalisuuden mahdollistaminen

Lapsille suunnitelluilla verkkosivuilla pitäisi erityisesti kiinnittää huomiota sosiaalisuuden ja yhdessä tekemisen tukemiseen. Multimediarikkaat ympäristöt ovat lasten mieleen, mutta vieläkin enemmän he rakastavat kanssakäymistä joko toistensa tai digitaalisten hahmojen kanssa [Fishler, 1998].

Lapset ovat hyvin sosiaalisia, ja he kokevat Internetissä viestien ja tarinoiden kirjoittamisen sosiaalisesti kanssakäymiseksi muiden lasten kanssa [Fishler, 1998]. Internetin avulla he voivat saada oman äänensä ja ajatuksensa miljoonien saataville [Druin and Solomon, 1996]. Lapsille suunnattujen verkkosivujen tulisikin tarjota työkaluja, jotka mahdollistavat sosiaalisen kanssakäymisen [Druin *et al.*, 1998].

6.2. Kontrollin mahdollistaminen

Lapset haluavat itse kontrolloida verkkosivuilla kulkemistaan ja tehtävien tekemistään [Druin *et al.*, 1998]. He kokevat ympäristön omakseen vasta kun he tuntevat, että heillä on kontrolli. Sivuja tulisi osata käyttää itse, ilman aikuisten apua [Brinck *et al.*, 2002]. Jos kontrollin määrä verkkosivuilla on vähäinen, he menettävät mielenkiintonsa sivuihin nopeasti. Kontrolli pitäisi myös saada nopeasti, eli sivuston käytön oppimisen tulisi tapahtua nopeasti [Druin *et al.*, 1998].

Lasten kokemuspohja tai taustatiedot eivät ole vielä samantasoista kuin aikuisilla, joten verkkosivuilla käytettävien metaforien tulisi heijastaa lasten omaa ymmärrystä, ympäristöä ja kieltä [Bernard, 2003]. Metaforien tulisi olla myös riittävän konkreettisia, sillä varsinkin pienet lapset eivät kykene vielä ajattelemaan abstraktisti [Bernard, 2003]. Lasten maailmaan sopivat metaforat ovatkin ensiarvoisen tärkeitä, jotta lapset kykenisivät itse käyttämään verkkosivuja.

6.3. Luovuuden mahdollistaminen

Lapset usein etsivät sivuilta sosiaalisuutta, luovuuden mahdollisuutta ja hauskuutta [Bernard, 2003]. He haluavat kertoa tarinoita, keksiä pelejä tai

leikkejä ja rakentaa asioita. He nauttivat äänestä, kuvasta, liikkeestä ja fyysisestä esiintymisestä [Druin *et al.*, 1998].

Sivuilla pitäisikin olla erilaisia työkaluja mahdollistamassa monenlaista luovuutta. Suosituimpia luovia toimintoja ovat tällä hetkellä tulostettavat värityskuvat, musiikin luominen ja kuvien piirtäminen [Bernard, 2003]. Tekemällä sivut riittävän kiinnostaviksi ja myös hieman haastavaksi, jaksavat lapset vierailla ja oleskella sivuilla usein [Bernard, 2003].

6.4. Turvallisuuden varmistaminen

Lapset ovat erityisen alttiita verkkosivujen kautta tapahtuvalle yksityisyyden loukkauksille. Sen tähden heille suunnitelluista ympäristöistä tulisi tehdä mahdollisimman turvallisia [Fishler, 1998]. Lapset eivät itse ymmärrä niitä riskejä, mitä esimerkiksi heidän sähköpostiosoitteensa antamisessa voi olla [Fishler, 1998]. Vastuu lasten turvallisuudesta onkin jo verkkosivujen suunnittelijoilla. Erityisesti tämä vastuu tulee ottaa huomioon silloin, kun lapsille suunnitellaan reaaliaikaista verkkoympäristöä [Fishler, 1998].

Myös verkkosivuilla tapahtuva mainonta aiheuttaa ongelmia lapsille. Lapset eivät osaa erottaa sivujen varsinaista sisältöä ja mainoksia toisistaan, vaan heistä suurin osa aukaisee sivuilla olevat mainokset [Nielsen, 2002]. Mainosten iloiset värit ja hauskat tai jännittävät kuvat houkuttelevat lapsia [Nielsen, 2002]. Mainosten kautta lapset saattavat usein myös eksyä varsinaisilta sivuilta.

7. Verkkosivujen käyttämisen suunnitteluperiaatteet

7.1. Sivujen aukeaminen

Lasten kärsivällisyys ei riitä sivujen tai pelien pitkiin latautumisaikoihin [Bernard, 2003]. He nauttivat kovasti multimediarikkaista ympäristöistä, mutta jos sivujen aukeaminen kestää liian kauan, he nopeasti siirtyvät muualle [Fishler, 1998].

Suurin osa lapsista myös odottaa, että kaikki verkkosivulla olevat kuvat ovat auenneet kokonaan ennen siirtymistä seuraavalle sivulle. He tuntuvat ajattelevan että se on pakollista [Sullivan *et al.*, 2000]. Ylenmääräinen odottelu tietysti aiheuttaa lapsissa suurta tyytymättömyyttä verkkosivuja kohtaan.

7.2. Navigoinnin toteuttaminen

Kun lapset käyttävät verkkosivuja, he harvoin tietävät mistä etsiä haluamaansa tietoa tai toiminnallisuutta. Lapset saattavat jopa käydä läpi samaa väärää reittiä yhä uudelleen, koska eivät yksinkertaisesti tiedä mitä muuta kokeilisivat. Erilaiset navigaatioavut ja yksinkertainen sivukaavio ovat sen tähden

suositeltavia erityisesti lapsille suunnatuilla verkkosivuilla. [Sullivan *et al.*, 2000]

Navigointilinkkien sijainnin tulee olla loogista ja yhdenmukaista. Jos yhdellä sivulla on eri kohdissa useita linkkejä samaan paikkaan, lapset eivät usein huomaa jo vierailleensa linkkien osoittamalla sivulla [Nielsen, 2002].

Navigointipainikkeiden tulee olla riittävän suuret. Lasten motoriset kyvyt eivät vielä ole kehittyneet kovin hyväksi, joten kovin tarkat hiirenkohdistukset eivät vielä onnistu [Nielsen, 2000].

Painikkeiden ja linkkien tulee olla selvästi esitettyjä. Niiden pitää edustaa lapsille tuttuja asioita ja paljastaa tarkoituksensa helposti [Bernard, 2003]. Niitä tulisi myös korostaa, jos niiden yli mennään hiirellä [Nielsen, 2002].

Verkkosivuilla tulee olla mahdollisuus palata takaisin edelliselle sivulle hyvin näkyvillä. Lasten on havaittu käyttävän takaisin-painiketta navigoinnissaan erittäin usein [Sullivan *et al.*, 2000].

7.3. Ohjeiden suunnitleminen

Lapset suostuvat lukemaan aikuisia useammin verkkosivuilla olevia ohjeita. Erityisesti tytöt viihtyvät sivuilla, joilla on riittävästi ohjeita, kun taas pojat taas eivät pidä liikaa tekstiä sisältävistä sivuista. [Nielsen, 2002]

Verkkosivuilla olevien ohjeiden tulisi olla lyhyitä, yksinkertaisia ja helppoja lukea [Bernard, 2003]. Sanasto ei saa olla liian monimutkaista [Nielsen, 2002]. Myös ohjeissa käytetyllä kirjasimella on merkitystä ohjeiden luettavuudelle ja miellyttävyydelle. Lasten mielestä mukavimmat fontit ovat Comic Sans Ms kokoa 12 tai 14 pistettä ja Arial kokoa 14 pistettä [Bernard, 2003].

7.4. Virhetilanteiden hallitseminen

Lapset eivät tule toimeen virhetilanteiden kanssa, eivätkä he yleensä kykene niistä toipumaan [Nielsen, 2002]. Virheilmoitusten yhteydessä lapset usein jättävätkin tehtävän kesken ja poistuvat sivustolta kokonaan [Nielsen, 2002]. Virhetilanteiden ennaltaehkäisy ja kunnollinen opastus ovatkin tärkeitä huomioida lapsille suunnatuilla verkkosivuilla.

Lapsilla on myös usein käytössään vanhoja, käytettyjä tietokoneita, joissa olevat ohjelmatkin ovat usein vanhentuneita [Nielsen, 2002]. Siksi lapset usein kohtaavatkin monenlaisia teknisiä ongelmia verkkosivuilla liikkueensa. Myös lasten tekniset taidot ovat vielä melko vaatimattomat. Nämäkin asiat tulee huomioida sivuja suunniteltaessa.

8. Yhteenveto

Nykyään jo monet aivan pienetkin lapset käyttävät Internetiä, ja selailevat sekä lapsille suunnattuja että aikuiselle suunniteltuja verkkosivuja. Tulevaisuudessa

lapsikäyttäjien määrä yhä kasvaa kovaa vauhtia tekniikan kehittymisen ja tietokoneiden leviämisen myötä.

Sitä mukaa kun käyttäjien määrä on noussut, on myös käytettävyyden merkitys verkkosivujen suunnittelussa kasvanut suuresti. Tällä hetkellä käyttäjällä on mahdollisuus valita miljoonien eri Internet-sivujen väliltä. Jotta saisi käyttäjät houkutelluksi omille sivuilleen, pitääkin suunnittelun ja käytettävyyden olla laadukasta ja kohderyhmälle sopivaa.

Lasten verkkosivujen suunnittelussa pitää ottaa huomioon samoja käytettävyyden näkökohtia kuin aikuisillekin suunniteltaessa, mutta samalla pitää myös huomioida lasten omaan kehitykseen, kulttuuriin, kieleen ja ympäristöön liittyviä ominaisuuksia. Niiden suunnittelun, toteutuksen ja arvioinnin jokaisessa vaiheessa pitää ajatella lapsia itseään, ja mielellään ottaa lapset mukaan kehitystyöhön.

Jotta lapset vierailisivat sivustolla usein, ja viihtyisivät siellä, pitää sivuston sisältää monia multimedian elementtejä. Sen pitää olla viihdyttävä ja hauska, innostava ja haastava. Sen pitää tukea lasten luontaista sosiaalisuutta ja luovuutta. Sisällön tulee olla korkealaatuista, ja sitä tulisi päivittää usein.

Navigoinnin tulisi sujua helposti ja itsenäisesti, ilman aikuisten apua. Sivuston rakenteen ja navigointityylin tulee olla selkeää ja yksinkertaista. Ikonien ja painikkeiden suunnittelussa tulee ottaa huomioon lasten motoriset kyvyt samoin kuin heidän vielä melko niukka kokemuspohjansa.

Lapset törmäävät verkkosivustojen käyttäjinä aivan samoihin käytettävyysongelmiin kuin aikuisetkin, mutta niistä toipuminen on heille huomattavasti vaikeampaa. Virhetilanteissa he monesti jättävät tehtävän kesken ja poistuvat sivulta. Heillä ei myöskään ole kärsivällisyyttä pitkiin latausaikoihin.

Lapset ovat myös huomattavasti alttiimpia yksityisyyden loukkauksille ja mainonnalle kuin aikuiset. He eivät erota sivujen sisältöä ja mainoksia toisistaan. He eivät myöskään ymmärrä henkilötietojensa paljastamisen riskejä. Vastuu lasten turvallisuudesta onkin verkkosivujen suunnittelijoilla ja ylläpitäjillä.

Jatkossa voisi näiden suunnitteluperiaatteiden pohjalta suunnitella ja toteuttaa lapsille suunnatut verkkosivut, joiden käytettävyyttä tutkittaisiin käytettävyydestien avulla. Testausten perusteella voisi suunnitteluperiaatteita vielä tarkentaa ja selventää. Näin saataisiin vieläkin hyödyllisemmät ohjeet lapsille sopivien, hyvän käytettävyyden omaavien verkkosivujen suunnitteluun.

Viiteluettelo

- [Bernard, 2003] Michael L Bernard, *How can I make my site more accessible to children?* Optimal Web Design, 2003. Available
<<http://psychology.wichita.edu/optimalweb/children.htm>> Viitattu 26.11.2003.
- [Brinck *et al.*, 2002] Tom Brinck, Darren Gergle, and Scott D. Wood, *Usability for the Web: Designing Web Sites That Work*. Academic Press, 2002.
- [Druin and Solomon, 1996] Allison Druin and Cynthia Solomon, *Designing Multimedia Environments for Children*. Wiley, 1996.
- [Druin *et al.*, 1998] Allison Druin, Ben Bederson, Angela Boltman, Adrian Miura, Debby Knotts-Callahan, and Mark Platt, *Children as Our Technology Design Partners*. 1998. Available
<<http://citeseer.nj.nec.com/druin98children.html>> Viitattu 26.11.2003.
- [Fishler, 1998] Karen D. Fishler, *Kids Talk Back*. Adobe Magazine, 1998. Available
<<http://www.adobe.com/products/adobemag/archive/pdfs/98smfekf.pdf>> Viitattu 13.12.2003.
- [ICDL] International Children's Digital Library.
<<http://www.icdlbooks.org/>> Viitattu 18.11.2003.
- [Nielsen, 1994] Jakob Nielsen, *Ten Usability Heuristics*. 1994. Available
<http://www.useit.com/papers/heuristic/heuristic_list.html> Viitattu 26.11.2003.
- [Nielsen, 2000] Jakob Nielsen, *Designing Web Usability: the Practice of Simplicity*. New Riders Publishing, 2000.
- [Nielsen, 2002] Jakob Nielsen, *Kid's Corner: Website Usability for Children*. Alertbox, 2002. Available
<<http://www.useit.com/alertbox/20020414.html>> Viitattu 29.10.2003.
- [Pelastustoimi] Sisäasiainministeriön pelastusosaston verkkosivujen lapsille suunnatut sivut. <<http://www.pelastustoimi.net/>> Viitattu 18.11.2003.
- [Preece *et al.*, 1994] Jenny Preece, Yvonne Rogers, Helen Sharp, David Benyon, Simon Holland, and Tom Carey, *Human-Computer Interaction*. Addison-Wesley, 1994.
- [SearchKids] *Digital Libraries for Children*. Human-Computer Interaction Lab, University of Maryland, USA. Available
<<http://www.cs.umd.edu/hcil/kiddesign/searchkids.shtml>> Viitattu 29.10.2003.
- [Sullivan *et al.*, 2000] Terry Sullivan, Cathleen Norris, Martha Peet, and Elliot Soloway, *When Kids Use the Web: A Naturalistic Comparison of Children's Navigation Behavior and Subjective Preferences on Two WWW*

Sites. *6th Conference on Human Factors & the Web*, 2000. Available
<<http://www.pantos.org/ts/papers/wkutw/>> Viitattu 26.11.2003.

[Särkkis] Särkänniemen virtuaalihuvipuisto.

<<http://www.sarkanniemi.fi/sahkis/index.php>> Viitattu 09.12.2003.

KYSELY INTERNETIN KÄYTÖSTÄ 2. - 6. LUOKKALAISILLE

Opiskelen Tampereen yliopistossa tietojenkäsittelytieteitä, ja tutkin tällä hetkellä lapsille suunnattujen verkkosivustojen käytettävyyttä. Tämän kyselyn tarkoituksena on kartoittaa 2. - 6. luokkalaisten Internetin käyttöä, sekä heidän ajatuksiaan muutamista lapsille suunnatuista verkkosivustoista. Lomakkeet käsitellään anonyymisti, ja tietoja yksittäisistä vastauksista ei anneta ulkopuolisille. Kyselyn tulokset raportoidaan siten, ettei niistä voi päätellä kenenkään yksittäisen vastaajan henkilöllisyyttä. Olkaa hyvä ja palauttakaa täytetty lomake opettajalle mahdollisimman pian, kuitenkin viimeistään maanantaina 24.11.2003. Kiitos yhteistyöstä!

Pälkäneellä 17.11.2003

Matleena Koivisto (sähköposti: matleena.koivisto@uta.fi)

Tietojenkäsittelytieteiden laitos, Tampereen yliopisto



KYSYMYKSET:

Ikä _____ vuotta.

Olen tyttö poika.

Onko teillä kotona tietokone?

Kyllä Ei

Onko teillä kotona Internet-yhteys?

Kyllä Ei

Kuinka usein käyt Internetissä?

Useita kertoja viikossa.

Pari kertaa kuukaudessa.

Kerran kuukaudessa tai harvemmin.

Kuinka kauan kerrallaan olet Internetissä? Noin 30 minuuttia.
 30 - 60 minuuttia.
 Yli 60 minuuttia.

Kerro, millä Internet-sivuilla käyt eniten, ja miksi.

1. Sivun osoite: _____

Miksi käyt tällä sivulla usein? _____

Mistä pidät tällä sivulla eniten? _____

Mikä sivulla on huonoa? _____

Onko sivulla helppoa vai vaikeaa liikkua tai tehdä asioita? Miksi?

2. Sivun osoite: _____

Miksi käyt tällä sivulla usein? _____

Mistä pidät tällä sivulla eniten? _____

Mikä sivulla on huonoa? _____

Onko sivulla helppoa vai vaikeaa liikkua tai tehdä asioita? Miksi?

3. Sivun osoite: _____

Miksi käyt tällä sivulla usein? _____

Mistä pidät tällä sivulla eniten? _____

Mikä sivulla on huonoa? _____

Onko sivulla helppoa vai vaikeaa liikkua tai tehdä asioita? Miksi?

Mitä mieltä olet näistä seuraavista sivustoista? Käy katsomassa ja kerro mielipiteesi.

1. Särkänniemen sivut lapsille - Särkkis 

Sivun osoite: www.sarkanniemi.fi/sahkis

Oliko sivujen löytäminen ja avaaminen helppoa vai hankalaa?

Mistä pidät näillä sivuilla eniten? _____

Mikä sivuilla on huonoa? _____

Onko sivuilla helppoa vai vaikeaa liikkua tai tehdä asioita? Miksi?

2. Pelastustoimen sivut lapsille



Sivun osoite: www.pelastustoimi.net

Valitse ensin hiirellä sivun ylälaidasta linkki [Turvatietoa](#). Sen jälkeen löydät lasten sivut vasemmalta.

Oliko sivujen löytäminen ja avaaminen helppoa vai hankalaa?

Mistä pidät näillä sivuilla eniten? _____

Mikä sivuilla on huonoa? _____

Onko sivuilla helppoa vai vaikeaa liikkua tai tehdä asioita? Miksi?

3. International Children's Digital Library



Tämä sivusto on englanniksi.

Sivun osoite: www.icdlbooks.org

Oliko sivujen löytäminen ja avaaminen helppoa vai hankalaa?

Mistä pidät näillä sivuilla eniten? _____

Mikä sivuilla on huonoa? _____

Onko sivuilla helppoa vai vaikeaa liikkua tai tehdä asioita? Miksi?

WWW-hakukoneiden käyttö opiskelun tiedonlähteenä

Anne Kunnari

Tiivistelmä

Tämä tutkielma käsittelee www-hakukoneiden käyttöä opiskelun tiedonlähteenä. Tutkielman keskeisenä sisältönä ovat www-hakukoneilla toteutetut käytettävyytestit ja niiden tulokset.

Avainsanat ja -sanonnat: Www-hakukoneet, internet-tiedonhaku.

Cr-luokat: H.3.3, H.3.5.

1. Johdanto

1.1. Aiheen kuvaus

Tutkimukseni aiheena on www-hakukoneiden käyttö opiskelun tiedonlähteenä. Tutkailen www-hakukoneiden ominaisuuksia käyttäjän näkökulmasta tieteellisen tiedon etsimisprosessissa. Tutkimuksessa käsitellään hakukoneiden osuutta tiedonhakuprosessissa, jossa käyttäjän on tulkittava saatuja tuloksia ja arvioitava niiden luotettavuutta. Tarkoitus on selvittää, miten käyttäjä lähestyy tieteellistä ongelmaa hakukoneen avulla, miten hakukoneen käyttö vaikuttaa tiedonhakuprosessin onnistumiseen, sekä millä perusteilla käyttäjä arvioi saatuja tuloksia ja niiden luotettavuutta.

Jotta voitaisiin ymmärtää www-hakukoneiden käyttöä opiskelun tiedonlähteenä, on ymmärrettävä hakukoneiden toimintaperiaatteet. Millä tavoin koneet hakevat, tunnistavat ja ilmoittavat haetun tiedon? Miten tietoja lajitellaan ja missä järjestyksessä ja millä tavalla ne ilmoitetaan?

Hakukoneiden käytettävyyssominaisuuksien lisäksi tavoitteenani on selvittää, miten käyttäjä suhtautuu www-hakukoneiden tarjoamiin linkkeihin. Millaisia sivustoja käyttäjä selaa, millaiset jättää huomiotta ja mihin luottaa, kun on kyseessä opiskelua varten tapahtuva tiedonhaku? Lisäksi aion huomioida hakukoneiden käytettävyyssominaisuuksia juuri tiedonhakuprosessia silmälläpitäen. Selvitän myös miten helppoa hakukoneiden käyttö on ja millaisilla sanoilla tiedonhaku onnistuu.

1.2. Aiheen tarpeellisuus

Yliopisto- ja korkeakouluopiskelussa itse hankitulla tiedolla ja oman tutkimuksen toteuttamisella on merkittävä rooli. Lisäksi erilaisten kurssien harjoitustyöt ja -tehtävät vaativat opiskelijalta jatkuvaa pääsyä tietolähteisiin.

Kirjastopalvelut, erilaiset arkistot ja tutkimusmateriaalit toimivat toki tiedonlähteenä, mutta nykyään internet ja erityisesti www-hakukoneiden käyttö on lisääntynyt rajusti.

Opiskeluun littyvässä tiedonhaussa tärkeää on tiedon saatavuus, luotettavuus ja tiedonhakuprosessin helppous. Internetin loputtomiin tietovarastoihin pääsee käsiksi pienellä vaivalla, omalta tai yliopiston koneelta. Kirjastolähteiden saatavuus taas voi ajoittain olla huono – varaukset suosituimpiin kirjoihin on tehtävä erittäin aikaisin, ja jotakin kirjallisuutta on mahdollisuus tutkailla vain käsikirjastossa ilman kotilainan mahdollisuutta.

Lähdeaineiston luotettavuus on tärkeä huomioitava seikka, kun etsitään tietoa internetistä. Painetut julkaisut tarjoavat tietoa kirjoittajasta ja niiden alkuperä on helppo selvittää. Internetissä julkaistavan aineiston alkuperä sitä vastoin on usein epäselvä. Jokainen voi julkaista aineistoa internetissä, eivätkä virallisiltakaan vaikuttavat dokumentit välttämättä ole tarkkoja tai totuudenmukaisia. Etsinnässä on siis syytä selvittää sekä sivuston ja kyseisen informaation julkaisija, julkaisutarkoitus ja viimeinen päivittämisajankohta [Ackermann, 1999].

Vaikka internetistä hankittu tieto vaatii tiedonhakilta erityistä tarkkaavaisuutta ja aineiston arviointi voi olla hyvinkin vaikeaa, on internetissä tapahtuva tiedonhaku opiskelijoiden joukossa hyvin yleinen ja suosittu tiedonhakutapa. Opiskelijat kokevat internetin käytön nopeana ja helppona tapana päästä käsiksi monenlaiseen tietoon. Tämän vuoksi www-hakukoneiden avulla tapahtuva tiedonhakuprosessi on kiinnostava ja tarpeellinen tutkimuskohde.

1.3. Yhtymäkohdat

Www-hakukoneiden käyttö opiskelun tiedonlähteenä sisältää ensisijaisesti kysymyksen siitä, miten ja millaista tietoa www-hakukoneita käyttäen voi löytää. Aihe käsittelee siis käytettävyyden lisäksi informaatiotutkimuksen ja myös tiedon visualisoinnin perusteita. Hakukoneiden käytettävyysominaisuuksia mitataan tiedonhakuprosessin vaivattomuudella ja hakukoneiden erilaisten ominaisuuksien käyttäytymisellä. Lisäksi tiedonhakuprosessin kannalta tärkeä kysymys on tiedon luotettavuus. Tällöin tutkimuskohteena on myös käyttäjän näkemys informaation sisällöstä, oikeellisuudesta ja laadusta.

1.4. Tutkimusohjelma

Tutkimuksen lähtökohtana ja pohjatietoina ovat internet-tiedonhausta julkaistu kirjallisuus, sekä internetistä saatava hakukoneita koskeva materiaali. Tämän jälkeen toteutetaan yksi pilottitesti, joka toimii lähtökohtana muiden testien suunnittelulle.

Varsinaisista käytettävyystesteistä raportoidaan testihenkilöiden toiminta, kohdatut ongelmat ja testien yhteydessä toteutettujen haastattelujen tulokset. Lisäksi testien suoritukseen käytettyä aikaa ja vastausten oikeellisuutta arvioidaan.

Aikaisemman tiedon ja toteutettujen testien pohjalta tehdään yhteenveto www-hakukoneiden tiedonhakuominaisuuksista ja pohditaan ratkaisuja havaittuihin ongelmiin.

1.5. Johdatus tutkielman lukuihin

Luvussa 1. esitellään tutkimusaihe ja sen rajaus, ja määritellään aiheen tarpeellisuus sekä tutkimuksen rakenne. Luvussa 2. perehdytään hakukoneiden taustaan, toiminnallisuuteen ja käyttötarkoituksiin. Lisäksi tässä luvussa esitellään kaikki testeissä käytetyt hakukoneet ja niiden tärkeimmät ominaisuudet ja toimintaperiaatteet. Luvussa käsitellään myös hakukoneiden ja internet-tiedonhaun ongelmia.

Luvussa 3. kuvataan toteutettavien käytettävyystestien rakenne ja tarkoitus. Lisäksi esitellään testitehtävät ja raportoidaan pilottitestin tulokset. Tässä luvussa myös raportoidaan yksityiskohtaisesti kaikki toteutetut käytettävyystestit ja testihenkilöiden haastattelujen tulokset.

Luku 4. käsittelee www-hakukoneista tutkimuksen perusteella havaittuja tuloksia. Aluksi tehdään yleisiä huomioita, minkä jälkeen raportoidaan testeissä havaitut seikat ja käsitellään mahdollisia parannusehdotuksia. Luvussa 5. tehdään yhteenveto tutkimuksen tärkeimmistä johtopäätöksistä.

2. Taustaa

2.1. Www-hakukoneiden toiminta

Www-hakukoneet voidaan lajitella karkeasti kahteen ryhmään sen mukaan, miten ne keräävät ja lajittelevat www-sivustojen sisältöä. Hakurobotteja hyödyntävät hakukoneet, kuten Google, keräävät tietokantansa sisällön automaattisesti. Toisessa ryhmässä ovat ihmisen tekemiä indeksointitehtäviä käyttävät hakukoneet, esimerkiksi Open Directory, joiden sisältö tulkitaan sivustojen kuvausten perusteella [Sullivan, 2003].

Hakurobotiksi nimitetään ohjelmistoa, joka kulkee internetissä puumaisesti, hypertekstilinkkejä pitkin tallentaen löytämiensä www-sivustojen tekstisisällön tietokantaansa. Ohjelma toimii annetuilla asetuksilla ja aloitusosoitteella, ja se päivittää jatkuvast tietokantaansa uusia, netissä julkaistuja sivustoja. Tällaisia ohjelmia kutsutaan myös hakuroboteiksi. Nämä robotit siis indeksoivat ja tallentavat tietosisältöjä tietokantoihin, joista käyttäjä suorittaa sanahakuja [Nikkanen, 1997].

Sivustoja voidaan indeksoida eri periaatteilla. Joissain tapauksissa niistä tunnustetaan avainsanat ja keskeiset termit, joiden perusteella hakutuloksia tuotetaan. Eräät hakukoneet tallentavat sivustojen koko tekstisisällön tietokantaansa, jolloin on kyse toisenlaisesta indeksoinnista [Koski, 2003]. Tätä indeksointia noudattavatkin yleisimmät käytössä olevat, tässäkin tutkimuksessa esille tulevat hakukoneet. Lisäksi useat hakukoneet jättävät huomiotta tietyt usein esiintyvät sanat, kuten "http", ".com" tai yksittäiset numerot [Glossbrenner, 1998].

Tulosten esittäminen on tärkeä osa hakukoneen toimintaa. Erilaisilla hakukoneilla tulosten järjestely ja luokittelu tapahtuu erilaisia algoritmeja käyttäen. Hakukoneet pyrkivät yleisesti tarjoamaan hakua parhaiten vastaavat tulokset ensimmäisenä listauksessa. Jotkut hakukoneet käyttävät luokittelukriteerinä sivustolle johtavien linkkien määrää, toiset hakusanan esiintymistiheyttä [Ackermann, 1999].

Hakukoneiden kattavuutta verrattaessa voidaan esittää erilaisia lukuarvioita. Sivuston www.searchenginewatch.com mukaan syyskuussa 2003 hakukoneiden indeksoimien tekstidokumenttien lukumäärät olivat seuraavat: Google 3.3 miljardia sivustoa, AllTheWeb 3.2 miljardia sivustoa ja AltaVista 1.0 miljardia sivustoa [Sullivan, 2003].

2.2. Testissä käytetyt hakukoneet ja niiden toimintaperiaatteet

2.2.1. Google

Google - hakukone osoittautui testeissä yleisimmin käytetyksi. Oulun yliopiston listauksessa kyseinen hakukone mainitaan vuoden 2002 kattavimpana hakukoneena [OY, 2003]. Googlen käyttämä hakutekniikka perustuu Larry Pagen ja Sergey Brinin Stanfordin yliopistossa kehittämään sivujen luokittelun PageRank-järjestelmään. PageRank -järjestelmä käyttää internetin linkkirakennetta hyväkseen selvittääkseen tietyn sivun laadun. Linkki sivustolta A sivustolle B tulkitaan siis eräänlaiseksi puoltoääneksi sivuston B laadukkuudesta. Tämän järjestelmän ohella Google käyttää ominaisuuksia, jotka tutkivat tekstien vastaavuutta [Google, 2003].

Googlea käyttämällä voi suorittaa haun yhdellä tai useammalla sanalla. And -optiota ei tarvita, sillä Google hakee vain tuloksia, jotka sisältävät kaikki hakutermit. Google jättää huomiotta muutamat tavallisimmat sanat ja merkit, jotka voivat vaikeuttaa hakua. Tällaisia sanoja ovat esimerkiksi "http" tai ".com" tai yksittäiset numerot. Google ei tue sanan osalla tapahtuvaa hakua. Näin ollen esimerkiksi haku sanan rungolla ja *-merkillä ei tuota tulosta. Google ei myöskään tee eroa isoilla ja pienillä kirjaimilla. Kokonaisia lauseita tai ilmaisuja käyttämällä voi myös suorittaa hakuja. Tällöin hakusanat on merkittävä lainausmerkkeihin. [Google, 2003].

Googlessa on myös lukuisia erityisominaisuuksia, joiden avulla hakua voi kohdentaa tehokkaammin. Tällaisia ominaisuuksia ovat muun muassa lisähakuoitot ja kielivalinnat. Googlen antamat hakutulokset ryhmitellään siten, että hakua parhaiten vastaavat on luetteloitu ensimmäisinä [Google, 2003].

2.2.2. AllTheWeb

AllTheWeb.com - sivustolta löytyvä hakukone on norjalainen, Overture Services Inc. - yrityksen tuote. Yhtiön sivustolla hakukoneen mainostetaan löytävän mitä tahansa, nopeammin kuin mikään muu hakukone. AllTheWeb - hakukone kattaa miljardeja tiedostoja, sivustoja ja multimedialla, ja se päivittää indeksinsä noin 7-11 päivän välein pysyäkseen ajan tasalla jatkuvasti kasvavassa internetissä. AllTheWeb -hakuja voi suorittaa 49 eri kielellä, ja hakukone sisältää erillisen uutishaun, jonka kautta voi hakea ajankohtaista uutistietoa eri puolilta maailmaa[AllTheWeb, 2003].

AllTheWeb -hakukone sisältää lisähakuominaisuuksia, joiden avulla käyttäjä voi rajata haun erilaisilla ehdoilla ja suodattimilla. Käyttäjä voi hakea pelkän sanalla tai ilmaisulla, käyttää "and"-, "andnot"-, "or"- tai "rank" - optioita, tai suodattaa vastauksia tiedostotyyppin, päivämäärän, mediatyyppin tai kielen perusteella[AllTheWeb, 2003].

2.2.3. AskJeeves

AskJeeves.com -sivuston tarjoama hakukone lähestyy hakuprosessia eri tavalla. Käyttäjä suorittaa haun kirjoittamalla hakukenttään valmiin kysymyksen. Sivuston mukaan kyseinen hakukone käyttää Natural Language Processing -tekniikkaa, minkä avulla hakukone ymmärtää haetut sanat sopivassa kontekstissa ja osaa näin tarjota sopivampia hakutuloksia. Hakutekniikan perustana on Teoma -ohjelmisto, joka indeksoi www-sivustoja niiden asiayhteyden mukaan - ei siis niihin johtavien linkkien perusteella. The

Teoma Crawler – ohjelmiston avulla AskJeeves –hakukone indeksoi sivustot ja yhdistelee niitä asiakokonaisuuksien mukaisesti [AskJeeves, 2003].

2.2.4. AltaVista

AltaVista.com –sivuston tarjoama hakukone on perinteisesti ollut yksi suurimmista ja käytetyimmistä hakukoneista. AltaVista on myös Overture Services, Inc. –yrityksen tuote, mikä selittää hakukoneiden samankaltaisen ulkoasun. AltaVista julkaisi ensimmäisen sanallisen internet-hakupalvelun vuonna 1995, minkä jälkeen hakukoneeseen on lisätty kieli- ja kuvahakuoitoita. Altavista vastaa lähestulkoon kaikilta ominaisuuksiltaan AllTheWeb –hakukonetta [AltaVista, 2003].

2.3. Www-hakukoneet ja tiedonhakuprosessi

Internet tarjoaa valtavan määrän helposti saatavilla olevaa tietoa. Näin ollen moni tiedonhakija on siirtynyt painetuista lähteistä täysin sähköisessä muodossa olevaan. Kun kaiken tiedon voi periaatteessa löytää verkosta, moni kokee painetun materiaalin käytön turhaksi. Usein kysymykseksi nouseekin juuri se, miten löytää varsinainen tieto internetin informaatiotulvasta [Kolari, 1999].

Tiedonhakuprosessin kannalta merkittävimmät vaikeudet ovat hakuehtojen määrittely, liian laaja hakutulosjoukko ja toisaalta liian vähäinen hakutulosjoukko. Hakukoneen onnistunut käyttö tiedonhaussa voi vaatia käyttäjältä erityissanaston ja kielen hallintaa. Monimerkityksiset sanat ja erityistä osaamista vaativa terminologia vaikeuttavat hakuprosessia. Tästä seuraavat kaksi muuta mainittua ongelmaa, informaation paljous tai vähäisyys [Morville, 1996].

Internetistä löytyy yhä enemmän laadukkaita ja asiallisia, painettuja julkaisuja vastaavia materiaaleja, sekä maksullisena että maksuttomana. Joitakin maksullisia aineistoja on saatu lisenssinvaraisina käyttöön juuri oppilaitosten aineistona käytettäväksi. Tämän lisäksi myös opinnäytteitä, tutkimusraportteja, väitöskirjoja ja verkkolehtiä löytyy runsaasti. Informaatiokanavina ovat hyödyllisiä myös erilaiset organisaatioiden, viranomaisten ja julkisten tahojen ylläpitämät sivustot. Tästä kehityksestä huolimatta painetut julkaisut pysyvät juuri luotettavuutensa ansiosta usein vähintään rinnakkaisena menetelmänä tiedonhaussa [YTK, 2003].

Käyttäjät voi hakea tietoa internetistä selailemalla erilaisia hakemistoja, etenemällä linkistä toiseen tai varsinaisia tiedonhakupalveluita, www-hakukoneita, käyttäen. Tässä tutkimuksessa olen keskittynyt juuri www-hakukoneiden tiedonhakuominaisuuksiin ja tiedonhakuprosessiin näiden

hakukoneiden avulla. Internetin tietotulva aiheuttaa sen, että mikään tiedonhakutapa ei takaa halutun ja toivotun tiedon löytämistä. Hakupalvelut kehittyvät ja muuttuvat jatkuvasti, hakuperusteet ovat erilaisia ja dokumenttien kuvaukset ovat puutteellisia ja epäyhtenäisiä [BTJ, 2001].

Tiedonhakuprosessin onnistuminen riippuu monista eri tekijöistä. Varsinaisen tiedonhaun suorittamisen jälkeen on tiedon oikeellisuutta ja tarkoituksenmukaisuutta arvioitava. Se, kuinka hyvin tiedonhaku on onnistunut, määräytyy sen mukaisesti kuinka paljon ja millaista tietoa hakuprosessin aikana on saatu hankittua. Hakutuloksia arvioidaan myös sen relevanttiuden kannalta. Aiherelevantilla aineistolla tarkoitetaan tiedonhaun aiheen kannalta hyvää ja keskeistä aineistoa, joka ei kuitenkaan välttämättä vastaa käyttäjän todellista tarvetta. Käyttäjärelevantti aineisto taas vastaa käyttäjän varsinaisia tiedontarpeita. Ongelmana on se, että tiedonhakija ei välttämättä kykene määrittelemään prosessin kuluessa oikeita hakusanoja tai haun asetteluja saavuttaakseen käyttäjärelevanttia aineistoa, todellisia tiedontarpeitaan vastaavaa materiaalia [TAYK, 2003].

Internetissä julkaistun materiaalin arvioinnissa on huomioitava monia seikkoja. Materiaalin luonteen lisäksi on arvioitava, voiko lähdettä käyttää hyväksi. Lisäksi on selvitettävä kuka materiaalin kirjoittaja tai julkaisija on, ja millainen tieteellinen tausta kyseisellä taholla on. On myös selvitettävä se, onko materiaali käynyt läpi tarvittavat arviointiprosessit ennen julkaisua. Julkaisupäivämäärä ja viimeinen päivytyspäivämäärä ovat myös oleellista tietoa, kun etsitään internetistä tietoa opiskelua varten [TAY, 2003].

Hakukoneiden erilaisten ominaisuuksien vuoksi tiedonhakuprosessin kannalta on oleellista tutustua moniin hakukoneisiin, ja opetella käyttämään niitä rinnakkaisina tiedonhakukanavina [Basch, 1996].

3. Tutkimuksen toteutus

3.1. Toteutettavien testien kuvaukset

Tutkimusta varten toteutetaan 8 käytettävyydestä. Koehenkilöinä toimivat yliopisto-opiskelijat, joilla kaikilla on kokemusta internetin ja www-hakukoneiden käytöstä. Testit toteutetaan vaihtelevissa ympäristöissä, kuitenkin siten että testitilanne kuvaa mahdollisimman hyvin koehenkilön reaalista tiedonhakutilannetta.

Testitehtävät ovat kysymyksiä, joihin on olemassa ja löydettävissä oikea vastaus www-hakukoneiden avulla. Kysymykset on suunniteltu mahdollisimman monipuolisiksi. Näin ollen niiden aihepiirit ovat vaihtelevia, ja joukossa on sekä historiallisia, että ajankohtaisia tiedonhakutehtäviä (ks. luku

3.2.). Lisäksi on pyritty huomioimaan se, että opiskelijoiden taustatiedot ovat erilaiset. Tehtävien yleisluontoisuus aiheuttaa sen, että tutkimus keskittyy itse tiedonhakuprosessiin ja sen ongelmiin.

Testi toteutetaan siten, että koehenkilölle kerrotaan lyhyesti tutkimuksen tarkoituksesta. Koehenkilöä pyydetään aluksi täyttämään aikaisempia asenteita ja hakukoneiden käyttötottumuksia mittaava kyselylomake. Tämän jälkeen koehenkilölle annetaan tehtävälappu, jossa on kuvattu kaikki tiedonhakutehtävät. Koehenkilö voi tämän jälkeen etsiä vastaukset kysymyksiin käyttämällä haluamaansa hakukonetta. Koehenkilö voi suorittaa tehtävät haluamassaan järjestyksessä. Löydettyään vastauksen, koehenkilö ilmoittaa suorittaneensa tehtävän kertomalla vastauksen ja sivuston testin vetäjälle.

Testin vetäjä tarkkailee testin kulkua ja tekee huomioita käytetyistä hakukoneista ja hakusanoista, sekä valituista linkeistä. Lisäksi tehtävien suorittamiseen kulunutta aikaa mitataan. Testin suorittamiseen varattiin alun perin noin 30 minuuttia, minkä ajan puitteissa koehenkilö voi tehtäviä suorittaa. Ajan ylittyessä testiä ei keskeytetä, mutta koehenkilöllä on mahdollisuus lopettaa testi sopivaksi katsomassaan kohdassa. Tehtävien suorittamisen jälkeen testin vetäjä kysyy koehenkilöltä kommentteja hakuprosessin etenemisestä. Tässä haastattelussa käyttäjältä kysytään, mitkä tehtävät käyttäjä koki erityisen vaikeaksi ja missä tehtävissä hän olisi mieluummin käyttänyt muuta tietolähdettä kuin hakukonetta.

3.2. Testitehtävät

Hakukoneiden käyttö tiedonhaussa on perusteltua, kun kyseessä on erityinen, selkeästi nimetty ongelma. Useampia näkökohtia sisältävät tutkimusaiheet ovat myös sopivia hakukoneiden avulla selvitettäväksi[Ackermann, 1999]. Tämä pyrittiin huomioimaan testitehtävien määrittelyssä.

Alla on kuvattu testissä käytetyt kysymykset tarkasti. Lisäksi alla olevassa listauksessa on tarkennettu kysymyksen erityinen tarkoitus tutkimuksen kannalta.

1. Grace Hopper menestyi eräällä tieteenalalla.

a) Mikä tämä ala oli?

b) Mikä oli Mark ja miten se liittyi Grace Hopperiin?

Tehtävä 1 on perinteinen tiedonhakutehtävä, jossa haetaan tietoa henkilöstä. Tarkoituksena oli saada testiin mukaan tehtävä, jonka koehenkilöt todennäköisemmin suorittavat samalla hakusanalla. Näin nähdään, eriävätkö

tulosten tulkinnat toisistaan, vaikka haku suoritettaisiinkin samoilla sanoilla. Lisäksi kyseessä on tehtävä, joka ei vaadi ajankohtaisuuden huomioimista. Oikea vastaus tehtävään löytyy sivustolta <http://www.sdsc.edu/ScienceWomen/hopper.html>.

2. Mainitse jokin tapahtuma/tilaisuus, johon pääministeri Matti Vanhanen osallistuu kuluvalla viikolla.

Tehtävässä 2 on tarkoitus mitata hakukoneiden kykyä tarjota ajankohtaista tietoa. Tämä tehtävänasettelu mittaa myös sitä, miten hakusana täytyy asetella, jotta päästään toivottuun tulokseen. Vastaus tehtävään löytyy valtioneuvoston ajankohtaiskalenterista osoitteesta <http://www.valtioneuvosto.fi>.

3. Elvis Presleyn uusi albumi on julkaistu.

a) Mikä on albumin nimi ja mikä ennen julkaisematon levytys sisältyy kyseiseen albumiin?

b) Elviksen tytär julkaisi myös äskettäin albuminsa. Mikä on hänen uusimman albuminsa nimi?

Tehtävässä 3 pyritään kartoittamaan, pystyykö käyttäjä löytämään tietoa myös alueelta, joka on selkeän kaupallinen. Lisäksi kysymyksenasettelu jättää hakusanan valitsemisen ja kielivalinnan käyttäjän päätettäväksi. Oikean vastaukset voi löytää esimerkiksi sivustoilta <http://www.click2music.fi/finland/viikkarit/2003/vk41/vk41.htm> ja <http://www.lisamariapresley.com>.

4. Mikä teatteri Suomessa toi syksyllä 2003 ensi-iltaan näytelmän "My Fair Lady"?

Tehtävä 4 käsittelee ajankohtaisaihetta, jonka todennäköisin hakusana on englanninkielinen. Näin saadaan tietoa hakukoneiden ominaisuuksista maa- ja kielikohtaisten hakutoimintojen suhteen. Tähän kohtaan löytyy kaksi oikeaa vastausta, Svenska Teatern sivustolta <http://www.svenskateatern.fi> ja pieksämäkeläinen Poleenin teatteri osoitteesta <http://www.pieksamaki.fi/sivistys/kulttuuri/teatteri/myfairlady.htm>.

5. Mitkä sähkölaitokset Suomessa ovat ottaneet käyttöön sähkömittareiden kaukoluennan?

Tehtävän 5 tarkoitus oli kartoittaa tilannetta, jossa tieto on löydettävä useista eri lähteistä. Sähkölaitosten toiminnoista kerrotaan yleensä ainoastaan niiden omilla sivustoilla, ja näin ollen käyttäjän on osattava löytää tietoa yleisemmistä lähteistä. Kysymyksenasettelu jättää avoimeksi sen, montako vastausta on ilmoitettava. Ainoa sähkölaitos, jonka kaukoluennatiedot löytyvät netistä on Turku Energia Oy, <http://www.turkuenergia.fi/fi/ajankohtaista/?position=4>.

6. Kuka voi suorittaa Suomessa laillisen avioliittoon vihkimisen?

Tehtävässä 6 viitataan lakiin, ja näin ollen tietolähde ja sen luotettavuus on merkittävässä asemassa. Lisäksi kysymyksenasettelu jättää varaa erilaisille hakusanoille, esimerkiksi "avioliittolaki" ja "vihkiminen". Oikea vastaus löytyy oikeusministeriön sivustolta osoitteesta <http://www.om.fi/81.htm>.

7. Montako ministeriä kuuluu Ruotsin hallitukseen?

Tehtävä 7 mittaa ruotsinkielisen informaation saatavuutta. Tehtävässä on myös tarpeen lähestyä vastausta hiukan eri tavalla, kuin kysymyksenasettelussa; ministerien määrää ei välttämättä suoranaisesti ole ilmoitettu, vaan ne on lueteltu ja laskettavissa. Ministerien lukumäärän löytää ruotsin hallituksen internetsivuilta osoitteesta <http://www.regeringen.se/regeringen/index.htm>.

8. Missä kolmessa EU-maassa sovelletaan poikkeavia säännöksiä Schengenin sopimuksen suhteen?

Tehtävä 8 korostaa myös tietolähteen luotettavuutta ja voi vaatia koehenkilöltä pitkienkin dokumenttien läpikäyntiä. Lisäksi kysymys jättää tulkinnanvaraa tuloksista – poikkeavia säännöksiä on käytössä monilla eri sopimuksen osaluilla. Schengeninsopimuksen poikkeavat säännökset löytyvät osoitteesta <http://www.europa.eu.int/scadplus/leg/fi/lvb/l33020.htm>.

3.3. Pilottitesti

Tutkimusta varten toteutettiin testaus, jossa seurattiin opiskelijaa reaali-tilanteessa. Tässä testissä oli tarkoitus saada suunta-antavaa näyttöä siitä, millaisia reaalisia tilanteita käyttäjä kohtaa etsiessään tietoa www-hakukoneita käyttämällä. Testissä pyrittiin myös tulevia testejä silmälläpitäen havainnoimaan, millaisia ongelmia tiedonhaussa ilmenee, ja miten toteutettavien testien tehtävät voitaisiin suunnitella tarkoitusta vastaaviksi. Opiskelija etsi ratkaisua erään kurssin harjoitustehtäviin internet-hakukoneen avulla.

Pilottitestin kysymyksiä oli Tampereen Teknillisen Yliopiston sähkömagnetiikka- kurssin harjoitustehtävät. Opiskelija etsi vastausta seuraaviin kysymyksiin: "Miten vähennetään DM-säteilyä (Differential-Mode Radiation)?" ja "Miten ilmoitetaan tehon muutos desibeleinä?".

3.3.1. Pilottitestin raportointi - osa I

Kysymykseen "Miten vähennetään DM-säteilyä?" opiskelija etsi vastausta Google-hakukoneella. Hakusanoina hän käytti fraasia "differential mode radiation". Tuloksista löytyy runsaasti aihepiiriä koskevia sivustoja, mutta näihin sivuihin perehtyminen veisi liikaa aikaa. Opiskelija lisää hakusanoihin sanan "reduce". Näin saadut tulokset liukuvat kauaksi haetusta aiheesta. Opiskelija kokeilee myös sanoja "reduction", "reducing" sekä sanojen suomenkielisiä vastineita, saavuttamatta tuloksia.

Opiskelija vaihtaa hakukonetta, ja hakee AllTheWeb - hakukoneella vastaavia sanoja. Opiskelija tutkailee paria ensimmäistä linkkiä, mutta ei löydä vastausta. Opiskelija kokeilee vielä kolmatta hakukonetta, ja esittää AskJeeves - hakukoneelle kysymyksen "How to reduce differential mode radiation?". Opiskelija tutkailee jälleen kerran muutamia linkkejä, mutta ei löydä vastausta kysymykseen. Opiskelija palaa käyttämään Googlea, ja suorittaa haun jälleen kerran samoilla hakusanoilla. Selailtuaan haun ensimmäistä linkkiä, hän löytää vastauksen kysymykseen.

Vastaus kysymykseen löytyy EeDesign.com - sivustolta, jota käyttäjä tutkaili jo aikaisemmin haussa. Syy siihen, että opiskelija ei löytänyt vastausta hakutuloksista aikaisemmin, on linkin esittelyssä mainittu ilmaus PCP. Tämä viittaa piirilevyihin, minkä vuoksi opiskelija ei tutkinut linkkiä tarkemmin.

3.3.2. Pilottitestin raportointi - osa II

Toisessa pilottitehtävässä opiskelija etsii vastausta kysymykseen "Miten tehon muutos ilmoitetaan desibeleinä?". Opiskelija käyttää Google -hakukonetta, ja hakusanoja "teho desibeleinä". Opiskelija kohtaa tuloksissa termit "teho" ja "tehovahvistus" rinnakkain, eikä löydä soveltuvaa vastausta. Opiskelija muuttaa hakusanaksi "tehon muutos", mistä saa tuloksena kaavan tehovahvistuksen ilmoittamiseksi desibeleinä. Tulos löytyy edu.fi - sivustolta.

3.3.3. Pilottitestin yhteenveto

Pilottitestin etenemistä tarkkailtaessa havaittiin, että tieteellisen tiedon haku opiskelusuorituksia varten voi olla hyvinkin monimutkaista. Kysyttäessä kommentteja hakuprosessista opiskelijalta, hän kertoo tutkailevansa pääsääntöisesti ensimmäiset linkit hakutuloksista ja välttävänsä sivustoja joissa mainitaan yritysten nimiä. Lisäksi opiskelija kertoo tutkivansa hakutuloksista

erityisesti koulujen ja tutkimuslaitosten sivustot. Opiskelija huomasi etsineensä liiaksi yleistä ja yleistettävää tietoa. Vastaus ongelmaan löytyi kuitenkin kaupallisen yrityksen sivuilta, jossa erityistapauksiin liittyvä tieto on suuressa roolissa.

Tärkeimmät pilottitestissä havaitut ongelmat olivat seuraavat:

- Hakusana tai hakusanat on vaikea määritellä yksiselitteisesti. Tehtävänannot ovat usein epäselviä, ja käyttäjän on osattava hakea monilla hakusanan muodoilla, sekä tilannekohtaisesti jollain muulla kielellä.
- Linkkeihin liittyvät kuvaukset eivät vastaa sivustojen todellista sisältöä.
- Tulosten alkuperä ja ajankohtaisuus ovat usein arveluttavia.
- Internet-yhteyden ongelmat vievät aikaa.

3.4. Käytettävyydestit

3.4.1. Testi 1

Käytetyt hakukoneet: Google

Tehtävien suorittamiseen kulunut aika: 29 minuuttia

Tehtävä 1: Grace Hopper menestyi eräällä tieteenalalla. a) Mikä tämä ala oli? b) Mikä oli Mark ja miten se liittyi Grace Hopperiin?

Käytetty hakukone: Google

Käytetyt hakusanat: "Grace Hopper"

Tehtävään käytetty aika: 3 min.

Tehtävän suoritus: valmis, vastaus oikea.

Sivusto, jolta vastaus löydettiin: www.sdsc.edu/ScienceWomen

Ensimmäinen haku toteutetaan sanoilla "Grace Hopper". Käyttäjä tutkii ensimmäistä tulosta, ja kertoo sivuston vaikuttavan vakuuttavalta, sillä linkin esittelyssä mainitaan "ScienceWomen". Tutkii ensimmäistä linkkiä ja avaa sivun www.sdsc.edu/ScienceWomen. Käyttäjä ilmoittaa vastauksen testikysymykseen 1a). Käyttäjä selaa samaa sivustoa edelleen ja löytää vastauksen myös kohtaan 1b).

Tehtävä 2: Mainitse jokin tapahtuma/tilaisuus, johon pääministeri Matti Vanhanen osallistuu kuluvalle viikolla?

Käytetty hakukone: Google

Käytetyt hakusanat: "Matti Vanhanen"

Tehtävään käytetty aika: 5 min.

Tehtävän suoritus: valmis, vastaus oikea.

Sivusto, jolta vastaus löydetty: www.eduskunta.fi

Käyttäjät suorittaa haun sanoilla "Matti Vanhanen" ja tutkii ensimmäisiä tuloksia. Käyttäjä siirtyy sivulle www.mattivanhanen.net, pääministerin kotisivuille ja etsii sivuilta ajankohtaiskalenteria. Käyttäjä tutkailee sivustoa, valitsee "Tavattavissa" -linkin, ja saa ilmoituksen "Sivua päivitetään". Käyttäjä siirtyy etusivun linkin kautta www.eduskunta.fi -sivustolle, josta valitsee linkin "Eduskunnassa tapahtuu". Tästä osuudesta käyttäjä löytää eduskunnan "päivän kokoukset". Käyttäjä pohtii, mihin kokouksiin pääministerin tulee osallistua. Käyttäjä ilmoittaa vastaukseensa eduskunnan täysistunnon, ilman selvää informaatiota pääministerin läsnäolosta kyseisessä kokouksessa.

Tehtävä 3: Elvis Presleyn uusi albumi on julkaistu. a) Mikä on albumin nimi, ja mikä ennen julkaisematon levytys sisältyy kyseiseen albumiin? b) Elviksen tytär julkaisi myös äskettäin albuminsa. Mikä on hänen uusimman albuminsa nimi?

Käytetty hakukone: Google

Käytetyt hakusanat: "Elvis Presley"

Tehtävään käytetty aika: 4 min.

Tehtävän suoritus: keskeneräinen, osa vastauksesta löydetty

Sivusto, jolta vastaus löydetty: www.elvis.com

Testihenkilö suorittaa haun sanoilla "Elvis Presley" ja tutkii ensimmäisiä tuloksia. Käyttäjä valitsee linkin, joka vie www.elvis.com -sivustolle. Käyttäjä selaa sivustoa, etsien kohtaa, jossa mainittaisiin sana "CD -levy". "Shop Elvis" -linkin kautta käyttäjä etenee "Music" -osuuteen, josta löytyy tiedote "Christmas Peace" -julkaisusta. Käyttäjä ilmoittaa tämän vastauksenaan, mutta ei halua jatkaa tehtävän suorittamista. Käyttäjä siirtyy tehtävään 4.

Tehtävä 4: Mikä teatteri Suomessa toi syksyllä 2003 ensi-iltaan näytelmän "My Fair Lady"?

Käytetty hakukone: Google

Käytetyt hakusanat: "My Fair Lady", haku suomi-sivuilta

Tehtävään käytetty aika: 3 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.pieksamaki.fi

Koehenkilö suorittaa haun hakusanoilla "My Fair Lady" ja valitsee option "haku suomi-sivuilta". Koehenkilö valitsee tuloksista ensimmäisen linkin sivulle www.pieksamaki.fi, josta löytyy vastaus kysymykseen. Käyttäjä ilmoittaa vastaukseen Poleenin Teatteri.

Tehtävä 5: Mitkä sähkölaitokset Suomessa ovat ottaneet käyttöön sähkömittareiden kaukoluennan?

Käytetty hakukone: Google

Käytetyt hakusanat: "sähkölaitos", "sähkömittari", "Helsingin energia"

Tehtävään käytetty aika: 5 min.

Tehtävän suoritus: keskeneräinen

Sivusto, jolta vastaus löydetty: -

Käyttäjä valitsee aluksi hakusanakseen "sähkölaitos" ja tutkii tuloksia. Käyttäjä ei löydä toivomaansa vastausta, ja muuttaa hakusanaksi "sähkömittari". Tuloksista käyttäjä valitsee www.tampere.fi/sl -sivuston, johon tutustuu tarkemmin. Käyttäjä selaa "Ajankohtaista" -osuutta, josta etsii uutisia aiheesta. Käyttäjä arvelee vastauksen olevan Helsingin energia, ja vaihtaa tämän hakusanoiksi. Tällä haulla käyttäjä pääsee www.helsinginenergia.fi -sivustolle, josta hän käy läpi "sähkö", "muut palvelut" ja "mittauspalvelut" -osuudet. Tämän jälkeen käyttäjä ilmoittaa, ettei löydä vastausta, ja jatkaa seuraavaan tehtävään.

Tehtävä 6: Kuka voi suorittaa Suomessa laillisen avioliittoon vihkimisen?

Käytetty hakukone: Google

Käytetyt hakusanat: "avioliitto"

Tehtävään käytetty aika: 1 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.om.fi

Koehenkilö suorittaa haun sanalla "avioliitto", ja selailee tuloksia. Oikeusministeriön sivusto herättää käyttäjän luottamuksen, ja hän valitsee kyseisen linkin. Lyhyen selailun jälkeen käyttäjä ilmoittaa oikean vastauksen kysymykseen sivulta www.om.fi.

Tehtävä 7: Montako ministeriä kuuluu Ruotsin hallitukseen?

Käytetty hakukone: Google

Käytetyt hakusanat: "Ruotsin hallitus", "Ruotsi"

Tehtävään käytetty aika: 4 min.

Tehtävän suoritus: keskeneräinen
Sivusto, jolta vastaus löydetty: -

Käyttäjä suorittaa haun sanoilla "Ruotsin hallitus", mutta ei löydä haluamaansa tulosta. Käyttäjä selailee muutamia linkkejä pikaisesti, ja törmää kahteen linkkiin, jotka antavat palautteen "Sivua ei voi näyttää." Käyttäjä valitsee linkin www.mtv3.fi -sivuston uutisosuuteen, jolta ei löydy vastausta kysymykseen. Seuraavaksi käyttäjä tutkailee www.naviterm.com -sivua, jolta löytyy termejä ja sopimuksia. Käyttäjä jättää tehtävän ratkaisematta ja siirtyy seuraavaan tehtävään.

Tehtävä 8: Missä kolmessa EU-maassa sovelletaan poikkeavia säännöksiä Schengenin sopimuksen suhteen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Schengenin sopimus", "EU"

Tehtävään käytetty aika: 4 min.

Tehtävän suoritus: keskeneräinen

Sivusto, jolta vastaus löydetty: www.join.fi

Koehenkilö hakee sanoilla "Schengenin sopimus". Ensimmäisen linkin kautta käyttäjä siirtyy oikeusministeriön sivuille, eu -säännösten kohdalle. Käyttäjä ei löydä vastausta, mutta mainitsee pitävänsä sivustoa oikeana lähteenä. Käyttäjä kokeilee hakusanaa "EU", mutta tulosjoukko näyttää liian laajalta. Käyttäjä valitsee googlen ominaisuuksista lisähaku -option. Käyttäjä ei kuitenkaan ymmärrä lisähaun ominaisuuksia, vaan palaa takaisin ja ilmoittaa, ettei löydä ratkaisua tehtävään.

Testihenkilön kommentit: Testihenkilö kokee, ettei osaa käyttää hakukoneita riittävän hyvin. Hänen mielestään lisähakuominaisuudet ja kieliopiot auttavat hakua, mutta niiden ominaisuudet ja käyttötavat on tunnettava hyvin. Käyttäjän mielestä www-hakukoneiden käyttö on kuitenkin nopeampaa kuin esimerkiksi tiedonhaku kirjoista.

3.4.2. Testi 2

Käytetyt hakukoneet: Google

Tehtävien suorittamiseen kulunut aika: 30 min.

Tehtävä 1: Grace Hopper menestyi eräällä tieteenalalla. a) Mikä tämä ala oli? b) Mikä oli Mark ja miten se liittyi Grace Hopperiin?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Grace Hopper"

Tehtävään käytetty aika: 3 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.cs.helsinki.fi

Testihenkilö suorittaa haun sanoilla "Grace Hopper". Hän tutkailee linkkejä ja siirtyy hakutulosten toiselle sivulle. Käyttäjä valitsee linkeistä sivuston www.cs.helsinki.fi, josta löytyy opinnäytetyö aiheesta. Tästä materiaalista käyttäjä löytää tarkan vastauksen kysymykseen.

Tehtävä 2: Mainitse jokin tapahtuma/tilaisuus, johon pääministeri Matti Vanhanen osallistuu kuluvalle viikolla?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Pääministeri Matti Vanhanen", haku suomi-sivuilta

Tehtävään käytetty aika: 6 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.helsinginsanomat.fi

Testihenkilö asettaa hakusanoiksi "Pääministeri Matti Vanhanen". Näitä tuloksia käyttäjä selaa pitkään, edeten hakutulosten seitsemännelle sivulle. Käyttäjä ei löydä tuloksista haluamaansa, ja palaa hakutulosten ensimmäiselle sivulle. Käyttäjä avaa linkin valtioneuvoston sivustolle "Ajankohtaista" - osuuteen, ja huomauttaa kyseisen lähteen olevan luotettava. Tutkailtuaan sivustoa hetken, käyttäjä palaa takaisin hakutuloksiin ja jatkaa tulosten selailua. Käyttäjä päätyy Helsingin sanomien sivulle uutisosuuteen, josta löytyy vastaus tehtävään.

Tehtävä 3: Elvis Presleyn uusi albumi on julkaistu. a) Mikä on albumin nimi, ja mikä ennen julkaisematon levytys sisältyy kyseiseen albumiin? b) Elviksen tytär julkaisi myös äskettäin albuminsa. Mikä on hänen uusimman albuminsa nimi?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Elvis Presley", haku suomi-sivuilta, "Lisa Marie Presley"

Tehtävään käytetty aika: 5 min.

Tehtävän suoritus: keskeneräinen, osa vastauksesta löydetty

Sivusto, jolta vastaus löydetty: www.elisa.net/viihde

Koehenkilö suorittaa haun sanoilla "Elvis Presley", käyttäen optiota "haku suomi-sivuilla". Käyttäjä selaa hakutuloksia neljännelle sivulle, mutta ei tutkaile mitään linkkiä lähemmin. Lopulta käyttäjä palaa ensimmäiselle tulossivulle, ja valitsee ensimmäisen linkin "official fan club" -sivustolle. Sivulta käyttäjä etenee osuuteen "Elvis -uutiset", mutta saa ilmoituksen "sivua ei voi näyttää". Käyttäjä siirtyy sivuston linkkikokoelmaan ja valitsee linkin www.elvisnews.com, josta palaa pian takaisin hakutuloksiin. Käyttäjä löytää tuloksista www.yle.fi -sivuston, ja yrittää löytää levyn nimeä levymyyntilistalta. Tehtävän ratkaisu on näytöllä, mutta käyttäjä ei huomaa sitä. Tämän jälkeen käyttäjä siirtyy takaisin hakutuloksiin ja ryhtyy ratkaisemaan kohtaa b). Käyttäjä valitsee hakusanakseen "Lisa Marie Presley" ja käy läpi tuloksista www.mtv3.fi - ja www.lisamariepresley.com -sivustot. Vastaus löytyy kuitenkin vasta www.elisa.net/viihde -sivustolta.

Tehtävä 4: Mikä teatteri Suomessa toi syksyllä 2003 ensi-iltaan näytelmän "My Fair Lady"?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "My Fair Lady"

Tehtävään käytetty aika: 2 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.pieksamaenseutuopisto.net

Käyttäjä suorittaa haun sanoilla "My Fair Lady", ja valitsee tuloksista Kulttuurin ajankohtaista - linkin. Tältä sivustolta löytyy maininta Poleenin teatterista. Käyttäjä mainitsee jo vastauksen kerran, mutta siirtyy tutkailemaan vielä hakutuloksia. Tämän jälkeen käyttäjä selaa hakutulosten toisen ja kolmannen sivun linkkejä. Lopulta käyttäjä palaa alkuun ensimmäiseksi tarkastelemalleen sivulle ja ilmoittaa vastaukseksi Poleenin teatterin.

Tehtävä 5: Mitkä sähkölaitokset Suomessa ovat ottaneet käyttöön sähkömittareiden kaukoluennan?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "sähkömittareiden kaukoluenta", haku suomi-sivuilla

Tehtävään käytetty aika: 1 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.turkuenergia.fi

Käyttäjä suorittaa haun sanoilla "sähkömittareiden kaukoluenta" ja lisää mukaan haun suomi-sivuilla. Ensimmäinen tulos näyttää sivuston

www.turkuenergia.fi, jonka kuvauksessa mainitaan sähkömittareiden kaukoluenta. Käyttäjä tarkastelee toista tulosta, joka on pdf-muotoinen tiedosto. Käyttäjä ilmoittaa, ettei halua avata kyseistä tyyppiä olevia tiedostoja, ja kertoo tehtävän vastaukseksi Turku Energia Oy:n.

Tehtävä 6: Kuka voi suorittaa Suomessa laillisen avioliittoon vihkimisen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "avioliittoon vihkiminen"

Tehtävään käytetty aika: 3 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.om.fi

Testihenkilö hakee sanoilla "avioliittoon vihkiminen" ja saa tulokseksi evankelis-luterilaisen kirkon sivut. Käyttäjä tutkailee kyseistä sivustoa, ja siirtyy avioliitto -linkin kautta oikeusministeriön sivuille, joissa selostetaan yksityiskohtaisesti vastaus kysymykseen.

Tehtävä 7: Montako ministeriä kuuluu Ruotsin hallitukseen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Ruotsin hallitus"

Tehtävään käytetty aika: 3 min.

Tehtävän suoritus: keskeneräinen

Sivusto, jolta vastaus löydetty: -

Käyttäjä suorittaa haun sanoilla "Ruotsin hallitus", mutta saa tulokseksi vain artikkeleita Ruotsin hallituksesta, ei varsinaista asiantietoa. Testihenkilö selaa linkkejä ja valitsee www.sverigedirekt.se -sivuston. Sivusto kertoo Ruotsin julkisen sektorin toiminnasta, mutta ministereiden lukumäärää ei mainita missään, joten testihenkilö jättää tehtävän suorittamisen kesken.

Tehtävä 8: Missä kolmessa EU-maassa sovelletaan poikkeavia säännöksiä Schengenin sopimuksen suhteen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Schengenin sopimus"

Tehtävään käytetty aika: 7 min.

Tehtävän suoritus: valmis, vastaus väärä

Testihenkilö suorittaa haun sanoilla "Schengenin sopimus" ja tutkailee saatuja tuloksia, aina toiselle hakutulossivulle asti. Täältä käyttäjä valitsee Faktatietoja

euroopasta -linkin, joka vie www.eurooppa-tiedotus.fi -sivustolle. Pitkän selailun jälkeen käyttäjä löytää maininnan kolmesta maasta, ja ilmoittaa vastauksikseen Suomen, Ruotsin ja Norjan.

Testihenkilön kommentit: Testihenkilö hakisi historiatietoa sisältävät asiat mieluummin kirjasta, tai jo ennestään tuntemaltaan sivustolta, joista hän mainitsee esimerkkinä sanomalehtien uutissivut. Testihenkilö kokee erilaisten hakuoptioiden käytön hankalaksi.

3.4.3. Testi 3

Käytetyt hakukoneet: Google

Tehtävien suorittamiseen kulunut aika: 19 min.

Tehtävä 1: Grace Hopper menestyi eräällä tieteenalalla. a) Mikä tämä ala oli? b) Mikä oli Mark ja miten se liittyi Grace Hopperiin?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "grace+hopper tieteenala", "Grace hopper", "Grace hopper Mark"

Tehtävään käytetty aika: 1 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.wic.org

Koehenkilö suorittaa haun sanoilla "grace+hopper tieteenala", ja valitsee saaduista tuloksista ensimmäisen linkin. Koehenkilö ei avaa pdf-tiedostoa, vaan suorittaa uuden haun jättämällä "tieteenala" sanan pois. Tämän jälkeen koehenkilö valitsee ensimmäisen linkin ja suorittaa selaimella sanahaun "mark" sivustolta. Vastaus molempiin kohtiin löytyy helposti.

Tehtävä 2: Mainitse jokin tapahtuma/tilaisuus, johon pääministeri Matti Vanhanen osallistuu kuluvalla viikolla?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Matti+Vanhanen", "Matti+Vanhanen vko 48"

Tehtävään käytetty aika: 2 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.keskusta.fi

Testihenkilö hakee sanoilla "Matti+Vanhanen" ja valitsee tuloksista pääministerin kotisivut. Tutkailtuaan sivua testihenkilö palaa takaisin tuloksiin ja lisää hakuun sanat "vko 48". Tämän haun tuloksista testihenkilö valitsee

Keskustan kotisivut, jonka ajankohtaisosuudessa mainitaan tietoyhteiskuntaseminaari, johon pääministeri osallistuu.

Tehtävä 3: Elvis Presleyn uusi albumi on julkaistu. a) Mikä on albumin nimi, ja mikä ennen julkaisematon levytys sisältyy kyseiseen albumiin? b) Elviksen tytär julkaisi myös äskettäin albuminsa. Mikä on hänen uusimman albuminsa nimi?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Elvis Presley new album", "Lisa+Marie+Presley new+album"

Tehtävään käytetty aika: 3 min.

Tehtävän suoritus: valmis, vastaus osittain oikea

Sivusto, jolta vastaus löydetty: www.dimatisah.com,
www.lisamariepresley.com

Koehenkilö hakee sanoilla "Elvis Presley new album", ja valitsee ensimmäisen hakutuloksen. Koehenkilö ilmoittaa vastaukseksi "Wants Some Booty" ja "Nookie". Toisessa kohdassa haku sanoilla "Lisa+Marie+Presley new+album" tuottaa ensimmäisenä hakutuloksena sivuston, jolta oikea vastaus löytyy.

Tehtävä 4: Mikä teatteri Suomessa toi syksyllä 2003 ensi-iltaan näytelmän "My Fair Lady"?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "my+fair+Lady ensi-ilta", haku suomi-sivuilta

Tehtävään käytetty aika: 1 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: koti.mbnet.fi/musicals

Käyttäjät suorittaa haun "my+fair+Lady ensi-ilta" ja lisää hakuasetukseksi haun suomi-sivuilta. Käyttäjät ohittaa ensimmäiset tulokset, valitsee linkin Musikaalit ja löytää oikean vastauksen nopeasti.

Tehtävä 5: Mitkä sähkölaitokset Suomessa ovat ottaneet käyttöön sähkömittareiden kaukoluennan?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "sähömittareiden kaukoluenta", "sähkölaitos kaukoluenta"

Tehtävään käytetty aika: 3 min.

Tehtävän suoritus: valmis, vastaus väärä

Sivusto, jolta vastaus löydetty: www.vtt.fi

Koehenkilö suorittaa haun sanoilla "sähömittareiden kaukoluenta", mutta virheellisen hakusanan vuoksi tuloksia ei löydy. Koehenkilö vaihtaa hakusanoiksi "sähkölaitos kaukoluenta", ja valitsee ensimmäisen www.energia.fi -linkin. Koehenkilö hakee selaimella sivustolta sanaa "kaukoluenta". Lopulta käyttäjä palaa hakutuloksiin ja valitsee mainosesitteen www.vtt.fi -sivustolta. Täältä löytyy listaus sähkölaitoksista, jotka koehenkilö ilmoittaa oikeaksi vastaukseksi.

Tehtävä 6: Kuka voi suorittaa Suomessa laillisen avioliittoon vihkimisen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "laillinen+avioliittoon+vihkiminen", "kirkko avioliittoon+vihkiminen", "maistraatti avioliittoon+vihkiminen"

Tehtävään käytetty aika: 4 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.maistraatti.fi, www.evl.fi

Testihenkilö suorittaa aluksi haun sanoilla "laillinen+avioliittoon vihkiminen". Tulokset liittyvät lähinnä historiallisiin tapahtumiin, kuten uskonpuhdistukseen. Koehenkilö selailee uskonpuhdistusta käsittelevää sivustoa, ja palaa suorittamaan uutta hakua, joka käsittää sanat "kirkko avioliittoon+vihkiminen". Näistä tuloksista käyttäjä löytää www.evl.fi -sivuston, jossa luetellaan ne kirkon virkamiehet, jotka vihkimisen voivat toimittaa. Tämän jälkeen käyttäjä suorittaa erillisen haun vaihtamalla "kirkko" -hakuehdon "maistraatti" -sanaksi. Tuloksista löytyy www.maistraatti.fi -sivusto, jossa mainitaan muut vihkimisoikeuden haltijat.

Tehtävä 7: Montako ministeriä kuuluu Ruotsin hallitukseen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "ruotsin+hallitus+ministerit"

Tehtävään käytetty aika: 1 min.

Tehtävän suoritus: keskeneräinen

Sivusto, jolta vastaus löydetty: -

Koehenkilö suorittaa haun sanoilla "ruotsin+hallitus+ministerit" ja selaa tuloksia. Koehenkilö vilkaisee EU-info -sivustoa, ja ilmoittaa ettei löydä vastausta kysymykseen.

Tehtävä 8: Missä kolmessa EU-maassa sovelletaan poikkeavia säännöksiä Schengenin sopimuksen suhteen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Schengenin sopimus maat"

Tehtävään käytetty aika: 4 min.

Tehtävän suoritus: valmis, vastaus väärä

Sivusto, jolta vastaus löydetty: www.uvi.fi

Käyttäjä hakee sanoilla "Schengenin sopimus maat" ja valitsee tuloksista sivuston www.viro.org. Pitkän selailun jälkeen käyttäjä siirtyy sivustolle www.uvi.fi, josta ilmoittaa vastaukseensa Suomen, Ruotsin ja Tanskan.

Testihenkilön kommentit: Testihenkilö koki vaikeimmaksi tehtäväksi Ruotsin ministerien lukumäärää koskevan tehtävän. Testihenkilö uskoo, että ruotsinkielisillä hakusanoilla vastaus olisi löytynyt helpommin. Kirjojen tai muiden lähteiden käyttö ei testihenkilön mielestä olisi ollut tarpeellista, lisäasetukset hän sitävastoin koki hyödyllisiksi. Testihenkilön mielestä www-hakukoneita käytettäessä informaation paljous on vaikeampi ongelma, kuin tiedon vähäisyys.

3.4.4. Testi 4

Käytetyt hakukoneet: Google

Tehtävien suorittamiseen kulunut aika: 17 min.

Tehtävä 1: Grace Hopper menestyi eräällä tieteenalalla. a) Mikä tämä ala oli? b) Mikä oli Mark ja miten se liittyi Grace Hopperiin?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Crace Hopper", "hopper", "Grace Hopper"

Tehtävään käytetty aika: 2 min.

Tehtävän suoritus: keskeneräinen, vastaus osittain oikea

Sivusto, jolta vastaus löydetty: www.sdsc.edu/ScienceWomen

Koehenkilö suorittaa haun sanoilla "Crace Hopper", ja tutkailee saatuja tuloksia. Kirjoitusvirheen vuoksi tulokset osoittavat Jim Crace -nimiseen henkilöön, eikä vastausta löydy. Käyttäjä vaihtaa hakusanaksi "hopper", mikä laajentaa tulosjoukkoa entisestään. Pian koehenkilö huomaa virheensä ja suorittaa uuden haun sanoilla "Grace Hopper". Vastaus ensimmäiseen kohtaan löytyy helposti, ja käyttäjä siirtyy suoraan seuraavaan tehtävään.

Tehtävä 2: Mainitse jokin tapahtuma/tilaisuus, johon pääministeri Matti Vanhanen osallistuu kuluvalle viikolla?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "matti vanhanen", "matti vanhanen 2003"

Tehtävään käytetty aika: 3 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydettiin: www.keskusta.fi

Testihenkilö hakee sanoilla "matti vanhanen" ja valitsee tuloksista pääministerin kotisivut. Käyttäjä tutkailee sivuston henkilötieto-osuutta, mutta palaa pian takaisin hakukoneeseen. Käyttäjä suorittaa uuden haun lisäämällä hakuehtoihin sanan "2003". Näistä tuloksista käyttäjä siirtyy Pravdan sivuille, ja edelleen wikipedia -sivustolle. Tulokset näyttävät asiaankuulumattomilta, joten käyttäjä palaa takaisin aikaisemman haun tuloksiin. Käyttäjä valitsee linkeistä www.vnk.fi -sivuston, jonka ajankohtaisuudesta löytyy merkintä pääministerin keskuskauppakamarilla pitämästä puheesta.

Tehtävä 3: Elvis Presleyn uusi albumi on julkaistu. a) Mikä on albumin nimi, ja mikä ennen julkaisematon levytys sisältyy kyseiseen albumiin? b) Elviksen tytär julkaisi myös äskettäin albuminsa. Mikä on hänen uusimman albuminsa nimi?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "elvis presley", "elvis presley julkaisematon", "elvis presley tytär", "elvis presley tytär", "lisa presley uusin albumi"

Tehtävään käytetty aika: 4 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydettiin: www.mtv3.fi/viihde, www.lisamariepresley.com

Koehenkilö hakee sanoilla "elvis presley julkaisematon", ja saa tuloksena linkin www.mtv3.fi/viihde -sivuille. Koehenkilö tutkailee sivustoa ja löytää vastauksen ensimmäiseen kohtaan nopeasti. Toisessa kohdassa käyttäjä hakee sanoilla "elvis presley tytär albumi", jonka tulokset ovat kaukana alkuperäisestä aiheesta. Käyttäjä tarkastelee pitkään erästä yksityistä kotisivua, jonka pääaiheena on Karen Wheeler, asiaan täysin kuulumaton henkilö. Käyttäjä suorittaa sanahaun "uusin albumi" sivustolta. Tulosta ei löydy ja käyttäjä palaa takaisin hakukoneeseen. Seuraavaksi käyttäjä tekee haun sanoilla "elvis presley tytär", jonka tuloksista käy ilmi nimi Lisa Marie Presley. Tämän jälkeen käyttäjä asettaa hakusanoiksi "lisa presley uusin albumi".

Näistä tuloksista löytyy myös linkki sivustolle www.lisamariepresley.com, josta oikea vastaus löytyy.

Tehtävä 4: Mikä teatteri Suomessa toi syksyllä 2003 ensi-iltaan näytelmän "My Fair Lady"?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "my fair lady", "my fair lady teatteri ensi-ilta"

Tehtävään käytetty aika: 1 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydettiin: www.pieksamaki.fi

Käyttäjä suorittaa haun "my fair lady". Tuloksissa ei näy suomalaisia teattereita, joten käyttäjä lisää hakuun sanat "teatteri ensi-ilta". Ensimmäinen näin saatu hakutulos sisältää oikean vastauksen.

Tehtävä 5: Mitkä sähkölaitokset Suomessa ovat ottaneet käyttöön sähkömittareiden kaukoluennan?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "kaukoluenta sähkömittari", "kaukoluenta sähkömittari sähkölaitos"

Tehtävään käytetty aika: 1 min.

Tehtävän suoritus: valmis, vastaus väärä

Sivusto, jolta vastaus löydettiin: www.vtt.fi

Testihenkilö tekee haun sanoilla "kaukoluenta sähkömittari". Näistä tuloksista käyttäjä tarkastelee mainosesitettä. Tämän jälkeen käyttäjä palaa hakuun ja lisää hakuehtoihin sanan "sähkölaitos". Käyttäjä selailee tuloksia ja vilkaisee nopeasti www.tampere.fi -sivustoa. Käyttäjä palaa takaisin tuloksiin ja mainosesitteeseen sivulle www.vtt.fi, ja nimeää vastaukseksi listauksen sähkölaitoksista.

Tehtävä 6: Kuka voi suorittaa Suomessa laillisen avioliittoon vihkimisen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "laillinen avioliittoon vihkiminen"

Tehtävään käytetty aika: 1 min.

Tehtävän suoritus: valmis, vastaus osittain oikea

Sivusto, jolta vastaus löydettiin: www.minedu.fi

Koehenkilö suorittaa aluksi haun sanoilla "laillinen avioliittoon vihkiminen". Tulokset hakeutuvat kauas alkuperäisestä aiheesta, ja käyttäjä siirtyykin tutkailemaan Uskonnonvapaus Euroopassa -artikkelia. Käyttäjä yrittää etsiä artikkelista sanahauulla termiä "laillinen", jota ei kuitenkaan löydy. Käyttäjä palaa tuloksiin ja tutkailee www.minedu.fi -sivustoa, josta hän ilmoittaa vastaukseksi uskontokunnan papin.

Tehtävä 7: Montako ministeriä kuuluu Ruotsin hallitukseen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "ruotsin eduskunta ministerit", "ruotsin hallitus ministerit", "ruotsin eduskunta ministerit määrä", "hallitus ruotsi", "hallitus ruotsi ministerit", "ruotsi eduskunta", "sverige minister"

Tehtävään käytetty aika: 3 min.

Tehtävän suoritus: keskeneräinen

Sivusto, jolta vastaus löydetty: -

Koehenkilö suorittaa haun sanoilla "ruotsin eduskunta ministerit", vaihtaa kuitenkin hakusanaksi "ruotsin hallitus ministerit". Saaduista tuloksista ei löydy haluttua, joten käyttäjä liittää hakuun sanan "määrä". Käyttäjä jatkaa näin määrittelemällä erilaisia hakusanayhdistelmiä, mutta ei löydä linkkiä, joka vastaisi tehtävänantoa. Tuloksina on sivustoja ja artikkeleja, joilla on paljon tekstiä, mutta joiden asiayhteys on epävarma. Käyttäjä yrittää hakea lopuksi ruotsinkielisellä hakuehdolla, mutta luovuttaa tarkastelematta tuloksia tarkemmin.

Tehtävä 8: Missä kolmessa EU-maassa sovelletaan poikkeavia säännöksiä Schengenin sopimuksen suhteen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Schengenin sopimus säännökset", "Schengenin sopimus poikkeavat säännökset", "Schengenin sopimus", "EU poikkeavat säännökset"

Tehtävään käytetty aika: 2 min.

Tehtävän suoritus: valmis, vastaus väärä

Sivusto, jolta vastaus löydetty: www.uvi.fi

Käyttäjä hakee sanoilla "Schengenin sopimus säännökset", tulosten ollessa epämääräisiä hän lisää sanan "poikkeavat", ja tutkailee www.finlex.fi -sivustoa. Tämän jälkeen käyttäjä suorittaa useita eri hakuja erilaisia hakusanoja käyttäen, ja löytää lopulta sivuston www.edlex.fi, jonka yhteydessä mainitaan Suomi, Islanti ja Norja. Tämän käyttäjä ilmoittaa vastaukseksi.

Testihenkilön kommentit: Testihenkilön mielestä testin vaikein osuus oli hakusanojen määrittäminen ja yhdisteleminen oikein. Vaikeimmaksi tehtäväksi koehenkilö mainitsee tehtävän Ruotsin ministereistä, jolloin haku olisi ollut tarkoituksenmukaista suorittaa ruotsin kielellä. Tästä huolimatta koehenkilö ei olisi halunnut käyttää painettuja julkaisuja tiedon etsimiseen. Testihenkilön mielestä viralliset työraportit, tutkimussivut, pdf-muotoiset tiedostot ja erilaiset seminaarityöt ovat luotettavia ja hyviä tietolähteitä. Käyttäjät kertoo myös välttelevänsä runsaasti mainoksia sisältäviä sivustoja.

3.4.5. Testi 5

Käytetyt hakukoneet: Google

Tehtävien suorittamiseen kulunut aika: 35 min.

Tehtävä 1: Grace Hopper menestyi eräällä tieteenalalla. a) Mikä tämä ala oli? b) Mikä oli Mark ja miten se liittyi Grace Hopperiin?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Grace Hopper", "Grace hopper Mark"

Tehtävään käytetty aika: 2 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.sdsc.edu/ScienceWomen

Koehenkilö suorittaa haun sanoilla "Grace Hopper" ja tutkailee ensimmäistä linkkiä. Sivustolta selviää heti vastaus ensimmäiseen kohtaan. Tämän jälkeen käyttäjä lisää hakuehtoihin sanan "Mark", ja siirtyy hakutuloksista sivulle www.cs.yale.edu, jossa vastaus on näkyvillä. Käyttäjät kuitenkin jättää kyseisen kohdan huomioimatta ja palaa aikaisemmalle sivustolle, josta eteenpäin selaamalla löytyy vastaus kysymykseen.

Tehtävä 2: Mainitse jokin tapahtuma/tilaisuus, johon pääministeri Matti Vanhanen osallistuu kuluvalle viikolle?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Matti Vanhanen viikko 49"

Tehtävään käytetty aika: 1 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.valtioneuvosto.fi

Testihenkilö hakee sanoilla "Matti Vanhanen viikko 49", ja ensimmäinen linkki hakutuloksista johtaa valtioneuvoston sivuille. Täältä löytyy linkki Viikko-

ohjelma, jossa on mainittu pääministerin viikon tapahtumat. Käyttäjä ilmoittaa vastaukseksi valtiopäivien puhemiehen tapaamisen.

Tehtävä 3: Elvis Presleyn uusi albumi on julkaistu. a) Mikä on albumin nimi, ja mikä ennen julkaisematon levytys sisältyy kyseiseen albumiin? b) Elviksen tytär julkaisi myös äskettäin albuminsa. Mikä on hänen uusimman albuminsa nimi?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Elvis Presley uusi albumi", "Lisa Marie Presley uusi albumi"

Tehtävään käytetty aika: 4 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.click2music.fi, www.mtv3.fi/viihde

Koehenkilö hakee sanoilla "Elvis Presley uusi albumi", ja valitsee tuloksista linkin BMG -levy-yhtiön uutuuksia käsittelevälle sivustolle. Vastaus ensimmäiseen kohtaan löytyy nopeasti. Seuraavan haun käyttäjä suorittaa lähes samoilla hakusanoilla, vaihtaen vain henkilön etunimeä. Tällöin tulokset ovat kirjavia; mukana on elvis-uutisia, erilaisia yksityisiä kotisivuja ja erilaisia kaupallisia sivustoja. Käyttäjä valitsee lopulta www.mtv3.fi/viihde -sivuston, ja valitsee haluamansa artistin nimen sivun sisäisestä valikosta, minkä jälkeen vastaus kysymykseen löytyy.

Tehtävä 4: Mikä teatteri Suomessa toi syksyllä 2003 ensi-iltaan näytelmän "My Fair Lady"?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "My Fair Lady ensi-ilta syksy 2003"

Tehtävään käytetty aika: 1 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.pieksamaki.fi

Käyttäjä suorittaa haun "My Fair Lady ensi-ilta syksy 2003", jonka tuloksien ensimmäisestä linkistä löytyy selkeä vastaus tehtävään.

Tehtävä 5: Mitkä sähkölaitokset Suomessa ovat ottaneet käyttöön sähkömittareiden kaukoluennan?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "sähkömittareiden kaukoluenta", "sähkömittareiden kaukoluenta Suomessa" "sähkömittarit etäluenta", "sähkömittareiden etäluenta", "sähkölaitokset kaukoluenta", haku suomi-sivuilta

Tehtävään käytetty aika: 5 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.turkuenergia.fi

Koehenkilö suorittaa haun sanoilla "sähkömittareiden kaukoluenta", ja tutkailee tuloksia. Tämän jälkeen käyttäjä kokeilee monenlaisia hakusanoja ja niiden erilaisia yhdistelmiä. Saadut tulokset eivät anna vastauksia suoraan. Lisäksi käyttäjä ei halua aukaista pdf-tiedostoja, eikä tutkailla pitkää seminaarityötä aiheesta. Käyttäjä vierailee energiamarkkinaviraston sivuilla, josta palaa kuitenkin alkuperäisen haun tuloksiin, mistä löytyy linkki www.turkuenergia.fi -sivustolle.

Tehtävä 6: Kuka voi suorittaa Suomessa laillisen avioliittoon vihkimisen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "avioliittoon vihkiminen Suomessa laillinen", "suorittaa avioliittoon vihkiminen Suomessa"

Tehtävään käytetty aika: 2 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.om.fi, www.evl.fi

Käyttäjän ensimmäinen haku on "avioliittoon vihkiminen Suomessa laillinen", jota hän kuitenkin muokkaa pian erilaiseen muotoon. Seuraavan haun tuloksista löytyy linkki www.helsinginseurakuntayhtyma.fi -sivustolle, jonka kautta käyttäjä löytää linkit sivuille www.om.fi ja www.evl.fi.

Tehtävä 7: Montako ministeriä kuuluu Ruotsin hallitukseen?

Käytetyt hakukoneet: Google, AskJeeves

Käytetyt hakusanat: "Ruotsin hallitus ministerit", "Ruotsin hallitus ministerien lukumäärä", "sverige regering minister", "sverige regering info", "Hur många minister finns det i Sverige?", "sverige regering minister"

Tehtävään käytetty aika: 10 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.regering.se

Koehenkilö suorittaa haun sanoilla "Ruotsin hallitus ministerit". Hakutulokset ovat sekavia, ja käyttäjä vaihtaa hakusanaa. Näistäkään tuloksista ei löydy

toivottua vastausta, joten käyttäjä vaihtaa hakusanan kieltä. Haku "sverige regering minister" tuottaa parempia tuloksia. Koehenkilö tutkailee tuloksia ja muokkaa hakusanaa moniin eri muotoihin. Lopulta koehenkilö löytää www.regering.se -sivustolle, jossa kaikki ministerit on listattu. Tästä huolimatta käyttäjä etsii edelleen sivustoa, jossa varsinainen lukumäärä on ilmoitettu. Koehenkilö siirtyy käyttämään toista, AskJeeves -hakukonetta. Useiden hakusanojen jälkeen käyttäjä palaa aikaisempiin tuloksiin, edellä mainitulle Ruotsin hallituksen sivustolle laskemaan mainittujen ministerien lukumäärän.

Tehtävä 8: Missä kolmessa EU-maassa sovelletaan poikkeavia säännöksiä Schengenin sopimuksen suhteen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Schengenin sopimus", "Schengenin sopimus poikkeavat säännökset", "EU Schengenin sopimus poikkeavat säännökset", "Schengenin sopimus", "Schengenin sopimus poikkeus", "Schengenin sopimus poikkeuksia", "Schengenin sopimus poikkeavia", "Schengen-maat"

Tehtävään käytetty aika: 10 min.

Tehtävän suoritus: valmis, vastaus osittain oikea

Sivusto, jolta vastaus löydetty: www.

Käyttäjät suorittaa useita hakuja, erilaisia sanamuotoja ja hakusanoja käyttäen. Käyttäjä selailee www.finlex.fi -sivustoa, mutta toteaa tekstien olevan liian laajoja. Tuloksissa on erilaisten ministeriöiden sivustoja, joita käyttäjä vilkaisee, muttei tutki tarkemmin. Käyttäjä tarkastelee myös Marjo Matikaisen henkilökohtaisia internetsivuja, verkkouutisia sekä työministeriön www-sivustoa tuloksetta. Lopulta käyttäjä valitsee www.eurooppa-tiedotus.fi -sivuston, jolla on mainittu Kreikka ja Tanska. Käyttäjä antaa vastaukseksi nämä maat.

Testihenkilön kommentit: Testihenkilön mielestä www-hakukoneet ovat ehdottomasti nopein ja helpoin tiedonhakukanava. Vaikeimpana koehenkilö piti viimeistä tehtävää, jonka hakutulokset olivat liian laajoja sivustoja. Hakusanan määrittely aiheutti käyttäjälle eniten ongelmia.

3.4.6. Testi 6

Käytetyt hakukoneet: AltaVista

Tehtävien suorittamiseen kulunut aika: 29 min.

Tehtävä 1: Grace Hopper menestyi eräällä tieteenalalla. a) Mikä tämä ala oli? b) Mikä oli Mark ja miten se liittyi Grace Hopperiin?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "Grace Hopper", "Grace hopper Mark"

Tehtävään käytetty aika: 2 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydettiin: www.sdsc.edu/ScienceWomen

Koehenkilö suorittaa haun sanoilla "Grace Hopper" ja tutkailee ensimmäistä linkkiä. Lisäämällä hakuun sanan "Mark", käyttäjä löytää vastauksen molempiin kohtiin.

Tehtävä 2: Mainitse jokin tapahtuma/tilaisuus, johon pääministeri Matti Vanhanen osallistuu kuluvalle viikolle?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "matti vanhanen vko 49", "pääministeri matti vanhanen viikko-ohjelma"

Tehtävään käytetty aika: 6 min.

Tehtävän suoritus: keskeneräinen

Sivusto, jolta vastaus löydettiin: -

Testihenkilö suorittaa haun sanoilla "matti vanhanen vko 49", mutta tuloksissa on täysin asiaankuulumattomia sivustoja, suunnistustuloksia ja yksittäisten henkilöiden kotisivuja. Tuloksia selattaessa pidemmälle löytyy linkki Keskustapuolueen sivuille. Käyttäjä suorittaa haun sivuston tapahtumakalenterista päivämäärän avulla, mutta ei löydä haluamaansa. Käyttäjä siirtyy seuraavaan tehtävään.

Tehtävä 3: Elvis Presleyn uusi albumi on julkaistu. a) Mikä on albumin nimi, ja mikä ennen julkaisematon levytys sisältyy kyseiseen albumiin? b) Elviksen tytär julkaisi myös äskettäin albuminsa. Mikä on hänen uusimman albuminsa nimi?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "elvis presley", "elvis presley unreleased", "lis marie presley"

Tehtävään käytetty aika: 6 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydettiin: www.kki.pl/elvisal/elvisnews.htm, www.lisamariepresley.com

Koehenkilö hakee sanoilla "elvis presley", ja valitsee tuloksista linkin www.elvis.com. Sivuston uutisosoudesta löytyy tietoa myynnissä olevista albumeista, mutta ei mainintaa siitä, mikä on äskettäin julkaistu. Käyttäjä palaa tuloksiin ja lisää hakusanaksi "unreleased". Näin saaduista tuloksista käyttäjä löytää puolalaisen Elvis -sivuston, josta vastaus löytyy. Toiseen kohtaan löytyy vastaus virheellisellä hakusanalla.

Tehtävä 4: Mikä teatteri Suomessa toi syksyllä 2003 ensi-iltaan näytelmän "My Fair Lady"?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "my fair lady ensi-ilta"

Tehtävään käytetty aika: 1 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.pieksamaki.fi

Käyttäjä suorittaa haun "my fair lady ensi-ilta", jonka tuloksien ensimmäinen linkki vie oikealle sivustolle.

Tehtävä 5: Mitkä sähkölaitokset Suomessa ovat ottaneet käyttöön sähkömittareiden kaukoluennan?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "sähkömittarit kaukoluenta"

Tehtävään käytetty aika: 2 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.turkuenergia.fi

Koehenkilö suorittaa haun sanoilla "sähkömittarit kaukoluenta" ja saa tuloksena yhden ainoan linkin sivustolle www.turkuenergia.fi. Käyttäjä ilmoittaa tämän vastaukseeseen.

Tehtävä 6: Kuka voi suorittaa Suomessa laillisen avioliittoon vihkimisen?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "avioliittoon vihkiminen", "siviilivihkiminen"

Tehtävään käytetty aika: 3 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.maistraatti.fi, www.evl.fi

Testihenkilö hakee sanoilla "avioliittoon vihkiminen". Tulokset sisältävät monenlaisia hääsivuja, eräs linkeistä on www.evl.fi, josta käyttäjä löytää osittaisen vastauksen. Tämän jälkeen koehenkilö suorittaa haun sanalla "siviilivihkiminen", jolla saadaan aikaisempaa hakua täydentävä tulos.

Tehtävä 7: Montako ministeriä kuuluu Ruotsin hallitukseen?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "svenska minister", "ruotsin ministerit", "Sverige minister"

Tehtävään käytetty aika: 3 min.

Tehtävän suoritus: keskeneräinen

Sivusto, jolta vastaus löydetty: -

Testihenkilö suorittaa hakuja sekä suomen että ruotsin kielellä. Molemmat haut tuottavat enimmäkseen kaupallisia sivustoja, joiden asiasisältö ei näytä lupaavalta. Käyttäjä selaa hakutulosten toista sivua, vaihtaa vielä hakusanaa toiseen muotoon ja siirtyy seuraavaan tehtävään.

Tehtävä 8: Missä kolmessa EU-maassa sovelletaan poikkeavia säännöksiä Schengenin sopimuksen suhteen?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "Schengenin sopimus poikkeavat säännökset", "EU Schengenin sopimus", "EU Schengenin sopimus poikkeava"

Tehtävään käytetty aika: 6 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.eurooppa.eu.int

Testihenkilö suorittaa hakuja erilaisilla sanamuodoilla ja tarkastelee useita pitkiä dokumentteja. Www.edilex.fi - lakikanavan sivuilta käyttäjä etsii selaimen haulla sanaa "poikkeava", mutta tuloksetta. Koehenkilö tutkailee myös eduskunnan sivustoa, jolta löytyy aihetta käsittelevä pdf-tiedosto. Koehenkilö toteaa tiedoston liian pitkäksi luettavaksi ja palaa hakutuloksiin. Käyttäjä vaihtaa taas hakusanan muotoa, ja valitsee ensimmäisen linkin www.eurooppa.eu.int -sivustolle. Osoitteesta löytyy oikea vastaus kysymykseen.

Testihenkilön kommentit: Testihenkilön mielestä vaikein tehtävä oli etsiä vastausta kysymykseen Ruotsin ministereiden lukumäärästä. Testihenkilö ei

olisi halunnut käyttää kirjaa tai muuta painettua materiaalia apunaan. Hänen mielestään myös sivustot, joilta hän tietoa etsi, olivat luotettavia.

3.4.7. Testi 7

Käytetyt hakukoneet: Google

Tehtävien suorittamiseen kulunut aika: 29 min.

Tehtävä 1: Grace Hopper menestyi eräällä tieteenalalla. a) Mikä tämä ala oli? b) Mikä oli Mark ja miten se liittyi Grace Hopperiin?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Grace Hopper", "Grace Hopper Mark"

Tehtävään käytetty aika: 4 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.sdsc.edu/ScienceWomen

Koehenkilö suorittaa haun sanoilla "Grace Hopper" ja tutkailee ensimmäistä linkkiä. Käyttäjä valitsee www.sdsc.edu -sivuston ja lukee tarkasti sen sisältöä. Käyttäjä lisää hakuun sanan "Mark", vilkaisee muutamaa linkkiä ja niiden sisältöjä, palaa ensimmäisen haun tuloksiin, ja löytää vastauksen alkuperäiseltä sivustolta.

Tehtävä 2: Mainitse jokin tapahtuma/tilaisuus, johon pääministeri Matti Vanhanen osallistuu kuluvalle viikolle?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Matti Vanhanen", "Matti Vanhanen vierailu"

Tehtävään käytetty aika: 5 min.

Tehtävän suoritus: keskeneräinen

Sivusto, jolta vastaus löydetty: -

Testihenkilö hakee aluksi sanoilla "Matti Vanhanen", ja valitsee tuloksista pääministerin kotisivut. Sivustolta löytyy linkki www.eduskunta.fi -sivustolle, josta käyttäjä kuitenkin palaa hakutuloksiin. Käyttäjä lisää hakuehtoihin sanan "vierailu". Tämän jälkeen tuloksiksi saadaan artikkeleita ja uutisjulkaisuja. Eräs niistä kertoo pääministerin vierailusta Kaarinassa "viime keskiviikkona", mutta artikkelin julkaisemispäivämäärä ei ole saatavilla. Käyttäjä tutkailee vielä muita hakutuloksia, mutta siirtyy seuraavaan tehtävään.

Tehtävä 3: Elvis Presleyn uusi albumi on julkaistu. a) Mikä on albumin nimi, ja mikä ennen julkaisematon levytys sisältyy kyseiseen albumiin? b) Elviksen

tytär julkaisi myös äskettäin albuminsa. Mikä on hänen uusimman albuminsa nimi?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Elvis Presley", "Lisa Marie Presley"

Tehtävään käytetty aika: 5 min.

Tehtävän suoritus: valmis, vastaus osittain oikea

Sivusto, jolta vastaus löydetty: www.elvis.com, www.lisamariepresley.com

Koehenkilö hakee sanoilla "Elvis Presley", ja valitsee tuloksista linkin www.elvis.com. Koehenkilö tutkailee uutisosuutta, ja selaa sisältöä eteenpäin. Sivuston ShopElvis -linkistä käyttäjä löytää kohtaan, jossa musiikkia on myytävänä. Ensimmäinen mainos koskee tulevaa julkaisua, Elvis Today, Tomorrow & Forever. Käyttäjä ilmoittaa tämän vastaukseksi. Toisen haun käyttäjä suorittaa sanoilla "Lisa Marie Presley", ja ensimmäinen linkki antaa oikean vastauksen.

Tehtävä 4: Mikä teatteri Suomessa toi syksyllä 2003 ensi-iltaan näytelmän "My Fair Lady"?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "2003 ensi-ilta My Fair Lady"

Tehtävään käytetty aika: 2 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: koti.mbnet.fi/musicals

Käyttäjä suorittaa haun "2003 ensi-ilta My Fair Lady", jonka tuloksista löytyy lyhyen selailun jälkeen Musikaalit -linkki. Täältä käyttäjä löytää ratkaisun tehtävään.

Tehtävä 5: Mitkä sähkölaitokset Suomessa ovat ottaneet käyttöön sähkömittareiden kaukoluennan?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "sähkömittareiden kaukoluenta"

Tehtävään käytetty aika: 3 min.

Tehtävän suoritus: keskeneräinen

Sivusto, jolta vastaus löydetty: -

Koehenkilö suorittaa haun sanoilla "sähkömittareiden kaukoluenta", ja tutkailee tuloksia. Käyttäjä yrittää avata pdf-muotoisen tiedoston, mutta tässä

esiintyy ongelmia. Käyttäjä tutkailee tuloksia, mutta siirtyy seuraavaan tehtävään.

Tehtävä 6: Kuka voi suorittaa Suomessa laillisen avioliittoon vihkimisen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "avioliittoon vihkiminen", "siviilivihkiminen"

Tehtävään käytetty aika: 2 min.

Tehtävän suoritus: keskeneräinen

Sivusto, jolta vastaus löydetty: -

Testihenkilö hakee sanoilla "laillinen avioliittoon vihkiminen". Jälleen tulokset liittyvät uskonnonvapauteen ja uskonpuhdistukseen. Käyttäjä avaa aihetta käsittelevän pdf-tiedoston, mutta ei halua selata pitkää tiedostoa eteenpäin. Tämän jälkeen käyttäjä tutkailee toista linkkiä, mutta palaa pian hakutuloksiin, ja siirtyy seuraavaan tehtävään.

Tehtävä 7: Montako ministeriä kuuluu Ruotsin hallitukseen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "ruotsin hallitus", "ruotsi hallitus", "ruotsin hallituksen ministerit"

Tehtävään käytetty aika: 4 min.

Tehtävän suoritus: keskeneräinen

Sivusto, jolta vastaus löydetty: -

Testihenkilö hakee tietoa ruotsin hallituksesta yhdistelemällä näitä sanoja erilaisin tavoin. Haut tuottavat lähinnä Euroopan Unionia koskevia tuloksia ja uutisarkistoja. Käyttäjä vierailee www.naviterm.com -sivustolla, jonka sisällöstä ja tarkoituksesta hän mainitsee olevansa epävarma. Käyttäjä huomauttaa, että haluaisi suorittaa haun ruotsiksi, mutta ei osaa asetella hakusanaa oikein. Lyhyen selailun jälkeen käyttäjä siirtyy seuraavaan tehtävään.

Tehtävä 8: Missä kolmessa EU-maassa sovelletaan poikkeavia säännöksiä Schengenin sopimuksen suhteen?

Käytetyt hakukoneet: Google

Käytetyt hakusanat: "Schengenin sopimus"

Tehtävään käytetty aika: 4 min.

Tehtävän suoritus: keskeneräinen

Sivusto, jolta vastaus löydetty: -

Koehenkilö suorittaa haun sanoilla "Schengenin sopimus", ja tutkailee tuloksista Baltic Guide Online -sivustoa. Seuraavaksi käyttäjä selailee www.virrankoski.net -sivustoa, jolta siirtyy www.finlex.fi -aineistoon. Tuloksettomana etsinnän jälkeen käyttäjä lopettaa tehtävän suorittamisen.

Testihenkilön kommentit: Testihenkilö koki viimeisimmät tehtävät ehdottomasti vaikeimpina. Painettuja julkaisuja, erityisesti sanomalehteä ja lakikirjaa, koehenkilö olisi halunnut käyttää kahden tehtävänannon kohdalla.

3.4.8. Testi 8

Käytetyt hakukoneet: AltaVista

Tehtävien suorittamiseen kulunut aika: 28 min.

Tehtävä 1: Grace Hopper menestyi eräällä tieteenalalla. a) Mikä tämä ala oli? b) Mikä oli Mark ja miten se liittyi Grace Hopperiin?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "Grae Hopper"

Tehtävään käytetty aika: 2 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.sdsc.edu/ScienceWomen

Koehenkilö suorittaa haun sanoilla "Grace Hopper" ja valitsee ensimmäisen linkin. Tältä sivustolta käyttäjä löytää vastauksen ensimmäiseen ja, hakemalla selaimen haulilla sanaa "mark", toiseen kohtaan.

Tehtävä 2: Mainitse jokin tapahtuma/tilaisuus, johon pääministeri Matti Vanhanen osallistuu kuluvalle viikolla?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "Matti Vanhanen", "Matti Vanhanen vko 49", "Matti Vanhanen tapahtuma", haku suomi-sivuilta

Tehtävään käytetty aika: 5 min.

Tehtävän suoritus: valmis, vastaus väärä

Sivusto, jolta vastaus löydetty: -

Hakiessaan sanoilla "Matti Vanhanen", koehenkilö saa tuloksiksi lähinnä erilaisia urheilutuloksia. Pidemmälle selailtaessa tuloksissa on mukana myös vanhasen kotisivu josta käyttäjä löytää valtioneuvoston sivulle. Käyttäjä vaihtaa hakusanaa, ja tuloksista löytyy uutinen Eero Järnefeltin näyttelyn

avajaisista, joiden avaajana toimii pääministeri Matti Vanhanen. Näyttelyn päivämäärä (8.9.-12.10) mainitaan sivustolla, mutta käyttäjä ilmoittaa tämän vastaukseen.

Tehtävä 3: Elvis Presleyn uusi albumi on julkaistu. a) Mikä on albumin nimi, ja mikä ennen julkaisematon levytys sisältyy kyseiseen albumiin? b) Elviksen tytär julkaisi myös äskettäin albuminsa. Mikä on hänen uusimman albuminsa nimi?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "Elvis Presley", "Elvis Presley new album", "Elvis Presley albums"

Tehtävään käytetty aika: 6 min.

Tehtävän suoritus: keskeneräinen

Sivusto, jolta vastaus löydetty: www.lisamariepresley.com

Koehenkilö hakee sanoilla "Elvis Presley" ja selaa tuloksista erilaisia epävirallisia kotisivuja, www.amazon.co.uk -kaupallista sivustoa sekä keräilyharvinaisuuksia esitteleviä sivuja. Käyttäjä lisää hakuun sanat "new album", mutta tulokset eivät lähene toivottua aihepiiriä. Käyttäjä siirtyy seuraavaan kohtaan, jonka ensimmäinen haku sanoilla "Lisa Marie Presley" tuottaa toivotun tuloksen.

Tehtävä 4: Mikä teatteri Suomessa toi syksyllä 2003 ensi-iltaan näytelmän "My Fair Lady"?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "My Fair Lady", haku suomi-sivuilta, "Suomen ensi-ilta My Fair Lady"

Tehtävään käytetty aika: 2 min.

Tehtävän suoritus: valmis, vastaus oikea

Sivusto, jolta vastaus löydetty: www.kaleva.fi/kulttuuri

Käyttäjä suorittaa haun "My Fair Lady", jonka tuloksista löytyy [kaleva.plus](http://www.kaleva.plus) -sivusto. Käyttäjä hakee vielä erilaisella hakusanalla, mutta palaa alkuperäisiin tuloksiin ja löytää tiedon.

Tehtävä 6: Kuka voi suorittaa Suomessa laillisen avioliittoon vihkimisen?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "laillinen avioliitto", "Suomen laki avioliitto", "Suomen laki avioliiton vihkiminen"

Tehtävään käytetty aika: 4 min.
Tehtävän suoritus: valmis, vastaus oikea
Sivusto, jolta vastaus löydetty: www.finlex.fi

Koehenkilö suorittaa haun sanoilla "laillinen avioliitto", ja löytää tuloksista Oikeusministeriön sivuston. Täältä käyttäjä etenee www.finlex.fi -säädöskokoelmaan, josta tehtävän ratkaisu löytyy.

Tehtävä 7: Montako ministeriä kuuluu Ruotsin hallitukseen?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "Ruotsin hallitus"

Tehtävään käytetty aika: 1 min.

Tehtävän suoritus: valmis, vastaus väärä

Sivusto, jolta vastaus löydetty: www.mtv3.fi/verkkouutiset

Haku sanoilla "Ruotsin hallitus" tuottaa tuloksiksi erilaisia artikkeleita, joita käyttäjä tutkailee. Eräs artikkeli listaa "Ruotsin hallituksen uuden kokoonpanon". Artikkelin julkaisuvuosi (1998) on mainittu, mutta käyttäjä ei huomioi tätä. Käyttäjä ilmoittaa vastaukseensa yhdeksäntoista, silloisen hallituksen mukaan.

Tehtävä 5: Mitkä sähkölaitokset Suomessa ovat ottaneet käyttöön sähkömittareiden kaukoluennan?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "sähkömittareiden kaukoluenta", "sähkölaitokset", "sähkölaitos kaukoluenta"

Tehtävään käytetty aika: 2 min.

Tehtävän suoritus: valmis, vastaus väärä

Sivusto, jolta vastaus löydetty: www.vtt.fi

Testihenkilö hakee erilaisilla hakusanayhdistelmillä, löytää linkin yritysinfo -sivustolle, jolta ei kuitenkaan löydy vastausta. Lopulta käyttäjä valitsee tuloksista mainosesitteen, jonka sisällöstä hän ilmoittaa löytäneensä vastauksen.

Tehtävä 8: Missä kolmessa EU-maassa sovelletaan poikkeavia säännöksiä Schengenin sopimuksen suhteen?

Käytetyt hakukoneet: AltaVista

Käytetyt hakusanat: "Schengenin sopimus", "poikkeava Schengenin sopimus"

Tehtävään käytetty aika: 6 min.

Tehtävän suoritus: valmis, vastaus väärä

Sivusto, jolta vastaus löydetty: -

Testihenkilö hakee sanoilla "Schengenin sopimus", mutta tulokset ovat hyvin hajanaisia. Käyttäjä lisää sanan hakuehtoihin, tutkailee sivustoja ja ilmoittaa vastaukseen Hollanti.

Testihenkilön kommentit: Testihenkilö koki hakusanan määrittelyn vaikeimmaksi tehtäväksi. Lisäksi hän etsisi tietoa jo tuntemiltansa sivustoilta hakukoneen käytön sijasta.

3.5. Haastattelut

Käytettävyydestien yhteydessä toteutettiin testihenkilöiden haastattelut, joissa selvitettiin käyttäjien asenteita ja ajatuksia www-hakukoneiden avulla tapahtuvasta tiedonhausta.

Kaikki koehenkilöt kertovat käyttävänsä hakukoneita opiskelujensa tiedonlähteenä vähintään silloin tällöin. Yhtä lukuunottamatta kaikki ovat sitä mieltä, että hakukoneiden avulla tapahtuva tiedonhaku on helpompaa, kuin tiedon etsiminen muista tietolähteistä. Lisäksi useimmat käyttäjistä uskovat löytävänsä tietoa nopeammin hakukoneen, kuin painettujen julkaisujen avulla.

Käyttäjiltä kysyttiin myös, mihin seikkoihin he kiinnittävät huomiota arvioidessaan sivustojen tietosisällön luotettavuutta. Vastauksiksi saatiin aineiston alkuperä, julkaisija, sivuston ylläpitäjä, tieteellinen tausta ja mainosten määrä. Aineiston ajankohtaisuutta tai sivuston/julkaisijan kansallisuutta ei pidetty erityisen tärkeinä seikkoina.

4. Tutkimustuloksia

4.1. Www-hakukoneista havaittua

Tarkasteltaessa www-hakukoneita yleisesti, huomataan että niiden ulkoasut poikkeavat keskinäisessä vertailussa toisistaan vain vähän. Hakusanojen syöttö tekstikenttään toimii AskJeeves -palvelua lukuunottamatta samalla tekniikalla. Myös sivustojen rakenne on yhtäläinen. Hakutulokset listataan sisäisten linkkien mukaan porrastetusti ja tulossivujen lukumäärä on esitetty sivun alaosassa. Myös kuva- ja uutishakuominaisuudet on toteutettu käyttäjälle näkyviltä osiltaan identtisesti kaikissa käsittelemissäni hakukoneissa.

Erot hakukoneiden ominaisuuksissa eivät siis ole ulkoisessa rakenteessa, vaan toiminnallisuudessa. Hakukoneiden käyttöliittymän tutkiminen ei näin

ollen anna vastauksia siihen, miten hakukoneita voi käyttää tiedonhaussa onnistuneesti. Tiedonhakuprosessin onnistuminen www-hakukoneella riippuu siitä, miten hyvin hakukone pystyy tunnistamaan käyttäjän tiedontarpeen.

4.2. Käytettävyydesteissä havaitut ongelmat

Tiedonhakuprosessi ei ollut täysin ongelmaton testihenkilöille. Hakukoneiden ominaisuuksissa, tiedonhaussa ja tiedon esittämisessä ilmenneet ongelmat olivat yhteneväisiä lähes kaikissa testeissä. Seuraavassa on listattu kaikki testissä ilmenneet, tiedonhakua vaikeuttaneet tapahtumat merkittävyysjärjestyksessä:

1. Testihenkilö suorittaa haun mielestään loogisella hakusanalla, mutta saadut tulokset ovat täysin asiaankuulumattomia.
2. Testihenkilö ei jaksaa selata pitkää tiedostoa.
3. Testihenkilö luottaa hakutulokseen, eikä tarkastele varsinaista sisältöä kriittisesti.
4. Sivuston julkaisupäivämäärä on epäselvä.
5. Testihenkilö ei osaa suorittaa hakua ruotsin kielellä.
6. Tehtävän suorittamista vaikeuttaa internet-yhteyden ongelmat.
7. Testihenkilö tekee kirjoitusvirheen.
8. Hakutuloksen kuvaus on harhaanjohtava.
9. Pdf-muotoinen tiedosto aiheuttaa ongelmia.
10. Testihenkilö käyttää vain yhtä hakusanaa.
11. Tehtävän suorittaminen edellyttää muuta tietämystä.
12. Testihenkilö ei onnistu suorittamaan hakua lisäasetuksin.
13. Mainossivustot häiritsevät tiedonhakua.

Ensimmäinen ja yleisin havaittu ongelma muodostui hakusanan valinnasta. Hakusana määriteltiin joko liian yleisesti, tai toisaalta liian tarkasti. Molemmissa tapauksissa informaation etsiminen vaikeutui, edellisessä tapauksessa liiallisen tietotulvan, jälkimmäisessä liian vähäisen tiedon vuoksi. Hakusanaa tai sen oikeaa muotoa ei osata nimetä, jolloin hakutulokset myös todennäköisesti käsittävät runsaasti epäoleellista informaatiota. Testihenkilön mielestä täysin looginen ja sopiva hakusana voi tuottaa hakukoneista yllättäviä tuloksia. Esimerkiksi AltaVistalla suoritettu haku sanoilla "matti vanhanen" antaa ensimmäisiksi tuloksiksi sivustoja suunnistuskisoista. Hakusanat "laillinen avioliittoon vihkiminen" taas antavat hakukoneesta riippumatta tulokseksi historiallisia dokumentteja uskonnonvapaudesta ja keskiajasta.

Toinen merkittävä ongelma aiheutui hakukoneiden antamien tulosten kuvauksista. Testihenkilö tulee "sokeaksi" hakutulosten suhteen, jolloin mikä tahansa hakutulos hyväksytään tietolähteenä, jos sieltä löytyy kriteerit täyttävä

tieto. Esimerkkinä mainittakoon kysymys sähkölaitoksista. Tällöin hakutuloksista löytyi mainosesitys, jossa kaukoluenta kyllä oli mainittu, mutta jonka varsinainen asiasisältö ei koskenut aihetta. Useat käyttäjät kuitenkin ilmoittivat vastauksensa juuri tästä esitteestä, sillä se löytyi hakutuloksena ja siinä oli lueteltu joitain sähkölaitoksia. Linkin kuvaus tuloslistassa voi myös osoittautua täysin erilaiseksi, kuin mitä itse sivuston tarjoama aineisto. Näin ollen hakutulosten tulkitseminen on aikaavievää ja vaikeaa.

Opetut käyttötavat ja tottumukset ovat myös osa hakukoneiden käyttöä. Käyttäjät valitsivat testeissä yleisimmin yhden, tutuimman hakukoneen ja käyttivät sitä läpi koko testin. Hyvin harva käytti tarkennetun haun mahdollisuutta tai kieliopioita. Käyttäjillä on selvästi oma yksilöllinen tapansa hakea tietoa. Eräät koehenkilöt suorittavat hakuja nopeasti monilla eri sanayhdistelmillä ja tutkailevat vain ensimmäisiä tarjottuja tuloksia päällisin puolin. Toiset taas suorittavat vain yhden sanahaun, ja tarkastelevat tämän jälkeen tuloksia perusteellisesti, useampia sivuja läpikäyden.

Muita ongelmia aiheuttivat esimerkiksi ruotsin kieli – englanniksi kaikki käyttäjät suorittivat hakuja sujuvasti ja hyvin tuloksin-, pdf-tiedostot, kirjoitusvirheet, internet-yhteyden ongelmat sekä haluttomuus selata pitkiä tiedostoja.

4.3. Tiedon oikeellisuus

Testitehtävien suoritusnopeutta ja vastausten tasoa seurattaessa ilmeni, että www-tiedonhaku onnistuu parhaiten selkeästi määritellyiltä alueilta. Tehtävä 4. osoittautui kaikkein helpoimmaksi ja nopeimmaksi ratkaista. Jokainen testihenkilö löysi oikean vastauksen ja tehtävän keskimääräinen suoritus-aika oli noin 1,6 minuuttia. Tehtävä 1. paljastui myös erittäin helpoksi ja kaikki koehenkilöt yhtä lukuunottamatta ratkaisivat sen täydellisesti, keskimääräisesti 2,4 minuutissa.

Vaikeimmiksi koehenkilöt kokivat tehtävät 7 ja 8. Vain harvat onnistuivat löytämään halutun tiedon suurimman osan jättäessä tehtävän suorittamisen kesken. Tehtäviin kulutettiin myös paljon aikaa, yhteensä keskimäärin 4,5 minuuttia.

Helpoksi koetut ja parhaiten suoritettut tehtävät olivat luonteeltaan hyvin yksiselitteisiä. Vaikeimmat taas olivat jo tehtävänannoltaan laajoja, ja edellyttivät käyttäjiltä jo aikaisempaa tietämystä aiheesta, tai vähintäänkin erikoisoptioiden ja vieraan kielen hallitsemista.

5. Yhteenveto

5.1. Www-hakukoneet tiedonhaun tukena

Tiedonhakuprosessin ongelmaksi ei tutkimuksessa muodostunut niinkään tiedon luotettavuus, vaan tarkoituksenmukaisen tiedon paikallistaminen. Tiedonhaku www-hakukoneen avulla onnistuu siis tarkasti rajatulta aihealueelta, hyvin määritellyllä hakusanalla erinomaisesti. Yksittäistä, tarkasti eriteltyä tietoa haettaessa www-hakukoneiden käyttö on hyvinkin tarkoituksenmukaista. Hakusanan muokkaaminen ja useiden hakujen suorittaminen erilaisilla hakusanoilla edesauttavat halutun tiedon löytämistä.

Hakukoneiden lisäominaisuudet ja erilaiset hakua kaventavat optiot epäilemättä toimivat tiedonhakuprosessin tukena, mutta toteutetuissa tutkimuksissa havaittiin, että vain harva hallitsee hakukoneiden erikoisominaisuuksien käytön. Tämän lisäksi havaittiin, että käyttäjä ei yleensä ole halukas käyttämään monia eri hakukoneita, vaan jatkaa huonojenkin tulosten jälkeen valitsemallaan, tutulla hakukoneella.

5.2. Hakukoneiden kehityksestä

Hakukoneita edelleen kehitettäessä on mielestäni jatkossa syytä huomioida erityisesti kielen monimuotoisuus, kaupallisten ja tieteellisten sivujen erottelu sekä linkkien kuvausten paikkansapitävyys. Ensisijaisesti kuitenkin juuri hakusanojen asettelun pulmallisuus vaikeuttaa muutoin helppoa ja käyttäjälle mieleistä tiedonhakutapaa oleellisesti. Mielestäni jatkossa hakukoneiden toimintaperiaatteiden parantamisessa tulisi keskittyä sanojen, asiayhteyksien ja aihekokonaisuuksien tunnistamisen problematiikkaan. Sekä tämä tutkimus, että aikaisemmat julkaisut aiheesta tukevat tätä johtopäätöstä.

Viiteluettelo

[Ackermann ja Hartman, 1999] Ernest Ackermann ja Karen Hartman, *The Information Specialist's Guide to Searching & Researching on the Internet & the World Wide Web*. Franklin, Beedle, London, 1999.

[AllTheWeb, 2003] *AllTheWeb-hakukoneen esittely* <http://www.alltheweb.com>, 2003.

[Basch, 1996] Reva Basch, *Secrets of the Super Net Searchers: the Reflections, Revelations and Hard-Won Wisdom of 35 of the World's Top Internet Researchers*. Wilton, 1996.

[BTJ, 2001] *Tiedonhakijan Opas*, BTJ Kirjastopalvelu, Helsinki 2001.

[Glossbrenner, 1998] Alfred Glossbrenner, *Search Engines for the World Wide Web*, Peachpit Press, Berkeley CA, 1998.

- [Google, 2003] *Google -hakukone*, <http://www.google.fi>, 2003.
- [HTTK, 2003] *Hämeen tietotekniikkakeskuksen tiedonhakuopas*, <http://www.htk.fi/kirjasto/inetopas/haku.htm>, 2003.
- [Kolari, 1999] Juha Kolari, *Tiedonhakijan Ominaisuudet Verkkotietoa Haettaessa ja Arvioitaessa*. Tampere, 1999.
- [Koski, 2003] *Huomioita Hakukoneista ja Internetistä*, <http://www.nicklaskoski.com>, 2003.
- [Morville, 1996] Peter Morville, *The Internet Searcher's Handbook: Locating Information, People & Software*. Neal-Schuman, New York, 1996.
- [Nikkanen, 1997] Kimmo Nikkanen, *Internet Tiedonhakukoneet - 2: Internet Oppijan Työkaluna*. Odin International, Jyväskylä, 1997.
- [OY, 2003] *Oulun yliopiston listaus hakukoneista ja -palveluista*, <http://www oulu.fi/pohti/haku.html>, 2003.
- [Sullivan, 2003] Dan Sullivan, *Search Engine Sizes*, <http://searchenginewatch.com>, 2003.
- [TAY, 2003] *Tampereen yliopisto - Tiedonhaku*, http://www.uta.fi/laitokset/kirjasto/opetus/kirjastokurssi/tiedonhaku_tiedhaunarviointi.html, 2003.
- [TAYK, 2003] *Tampereen yliopiston kirjasto - internet- aineiston arviointikriteerejä*, http://www.uta.fi/laitokset/kirjasto/oppimiskeskus/palvelut/arviointi_shtml, 2003.
- [YTK, 2003] *Yhdyskuntasuunnittelun tutkimus- ja koulutuskeskus*, <http://virtuaali.tkk.fi/yhdyskuntasuunnittelu/ytk-eri/Tiedonhaku/internet.html>, YTK, 2003.

Suorakäyttöisten kaavionmuokkaustyökalujen käytettävyys

Päivi Kärkkäinen

Tiivistelmä.

Tutkielman tarkoituksena on selvittää eräiden suorakäyttöisten kaavionmuokkaustyökalujen käytettävyyttä käytettävyystestien avulla. Työkalut on kehitetty Tampereen yliopiston tietojenkäsittelytieteiden laitoksella, ja niiden avulla käyttäjä voi tasata, kopioida, siirtää ja poistaa piirto-objekteja sekä muuttaa niiden kokoa. Testeihin osallistui yhdeksän koehenkilöä, ja tulosten perusteella työkaluja voidaan sanoa helppokäyttöisiksi sekä hyödyllisiksi kaavioiden muokkaamiseen liittyvissä tehtävissä, mutta testeissä tuli ilmi myös parannusehdotuksia niiden toiminnallisuuteen.

Avainsanat ja -sanonnat: kaavionmuokkaustyökalut, käytettävyyden arviointi, suorakäyttöisyys.

CR-luokat: D.2.2, H.1.2, H.5.2

1. Johdanto

Kaavioiden sekä muiden piirrosten piirtämiseen on olemassa runsaasti tietokoneohjelmia, joista osa bittikarttapohjaisia ja osa objektipohjaisia. Piirrosten muokkaamiseen on myös kehitetty monia työkaluja, joista yleisimmät liittyvät piirto-objektien siirtelyyn, poistamiseen kopioimiseen, tasaamiseen sekä ryhmittelyyn. Nämä työkalut esiintyvät lähes kaikissa piirtosovelluksissa.

Piirtosovellusten työkaluja käytetään usein komentopohjaisesti, mutta luonnollisempi tapa esimerkiksi muokkaustyökalujen käytölle on suorakäyttöisyys. Suorakäyttöisyys tietokoneohjelmissa tarkoittaa sitä, että käyttäjä saa välittömän palautteen toiminnoistaan. Tällöin sekä toiminnon kohteena olevat objektit että käyttäjän suorittamat toiminnot ovat selkeästi nähtävissä [Shneiderman, 1992].

Tutkimuksen kohteena olevat työkalut on kehitetty Tampereen yliopiston tietojenkäsittelytieteiden laitoksella, ja ne on tarkoitettu kaavioiden suorakäyttöiseen muokkaamiseen. Ne toimivat R2Java-piirtosovelluksessa, joka on Java-pohjainen kaavioeditori. R2Java-ohjelmalla voidaan piirtää suorakaiteen, ellipsin sekä pyöristetyn suorakaiteen muotoisia objekteja sekä yhdysviivoja näiden välille. Työkaluilla voidaan tasata, siirtää, kopioida sekä poistaa piirtoalueella olevia objekteja.

Luvussa kaksi kerrotaan piirto-ohjelmista sekä niissä käytettävistä yleisimmistä työkaluista, sillä monessa sovelluksessa käytetään samanlaisia työkaluja objektien muokkaamiseen. Luvussa kolme selvennetään suoraikäyttöisyyden käsitettä sekä kerrotaan joistain sovellusalueista. Luvussa neljä esitellään tutkimuksen kohteena olevat kaavionmuokkaustyökalut sekä R2Java-piirtosovellus, jossa työkalut toimivat. Luvussa viisi kerrotaan tutkimuksessa tehdyistä käytettävyydesteistä. Siinä esitellään testitehtävät sekä kerrotaan testijärjestelyistä. Luvussa kuusi esitellään käytettävyydestien tuloksia. Siinä käsitellään tehtävien suorittamista, esiintyneitä ongelmia sekä koehenkilöiden mielipiteitä. Luvussa seitsemän kerrataan tutkimuksen tulokset sekä esitetään ehdotuksia jatkotutkimuksien aiheiksi.

2. Piirto-ohjelmat

Kaavioiden piirtämiseen on olemassa monia sovelluksia, sekä bittikarttapohjaisia että objektipohjaisia. Tämän tutkimuksen kohteena olevat työkalut toimivat objektipohjaisessa piirto-ohjelmassa, joten tässä luvussa keskitytään esittelemään vain tämän tyyppisissä piirto-ohjelmissä käytettäviä työkaluja. Luvussa mainitut työkalut sekä muut piirto-ohjelmien piirteet perustuvat ArgoUML [2003], Fujaba [2003], SmartDraw [2003], CorelDraw [2003] sekä FreeHand [2003] – piirto-ohjelmiin.

Yleisin syöttölaite piirtämiseen tarkoitetuissa tietokoneohjelmissä on hiiri. Piirtäminen tapahtuu hiiren kursorin kautta, ja piirtojälgjen paksuutta voi yleensä vaihdella. Meyerin ja muiden [1994] tekemässä tutkimuksessa hiiri osoittautui parhaimmaksi piirto-ohjelman syöttölaitteeksi sekä testien tulosten että käyttäjien mielipiteiden perusteella. Testeissä vertailtiin hiiren, hiirikynän (mouse pen), ohjauspallon, piirtimen (stylus) sekä kosketusnäytön käyttöä graafisen käyttöliittymän piirto-ohjelmassa.

2.1. Yleisimmät muokkaustyökalut

Piirto-ohjelmissä yleisin työkalu on niin sanottu nuolityökalu, jolla voidaan valita objekteja sekä siirtää niitä. Usein työkalulla voi myös muuttaa objektin kokoa painamalla objektin kulmassa hiiren vasenta painiketta ja raahaamalla hiirellä. SmartDraw - ohjelmassa [SmartDraw, 2003] objektien kokoa voidaan muuttaa myös valikkokomennolla, jolloin ensin valitaan objektit nuolityökalulla ja sen jälkeen toiminto valikosta. Ohjelmassa voidaan muuttaa objekteja pienemmäksi, suuremmaksi tai useampi objekti samansuuruisiksi.

Myös objektien kopiointi on yleinen ja hyödyllinen toiminto piirto-ohjelmissä. Varsinaista kopiointityökalua piirto-ohjelmissä ei yleensä ole, vaan valittujen objektien kopiointi tapahtuu muiden yleisimpien tietokoneohjelmien ta-

voin näppäimistön ctrl-c -komennolla tai valitsemalla toiminto valikosta. Kopioidun objektin liittäminen tapahtuu vastaavasti ctrl-v -komennolla tai vastaavalla valikkokomennolla. Myös objektien poistaminen tapahtuu yleensä valitsemalla objekti ja painamalla näppäimistön delete-painiketta tai vastaavaa painiketta työkalurivillä.

Piirtosovelluksissa on usein jokin toiminto objektien tasaamiseen. Objektit voidaan tasata yläreunan, alareunan, vasemman sekä oikean reunan suhteen samalle tasolle. Tasaaminen tapahtuu yleensä valikkokomennolla. Esimerkkihjelmista vain ArgoUML-ohjelmassa [ArgoUML, 2003] oli mahdollista käyttää suoraikäyttöistä työkalua objektien tasaamiseen. Yleisiä toimintoja piirto-ohjelmissa ovat myös objektien asettelu toistensa päälle tai alle, objektien pyörittäminen sekä ryhmittäminen.

2.2. Työkalujen käyttäminen

Piirtämiseen tarkoitetuissa tietokoneohjelmissa joitain muokkaustoimintoja käytetään niin, että ensin valitaan muokattava objekti, sitten haluttu toiminto ja vasta sen jälkeen käyttäjä näkee muokkauksensa lopputuloksen. Käyttäjän kannalta tämä ei ole paras mahdollinen tapa, sillä Robbinsin ja muiden [1999] mukaan tällöin käyttäjälle aiheutuu huomattavaa kognitiivista kuormitusta. Tämä johtuu siitä, että tällöin käyttäjä joutuu työkalua valitessaan vertaamaan mielessään ollutta haluttua lopputulosta vain työkalun nimen tai sitä kuvaavan ikonin kanssa. Muokkaustoimintojen siirtyminen suoraikäyttöisempään suuntaan olisi siis tämän perusteella käyttäjälle eduksi.

St. Amant ja Horton [2002] esittelevät uuden tavan käyttää erilaisia työkaluja piirto-ohjelmassa. Heidän mukaansa työkalujen käyttö tietokoneohjelmissa poikkeaa liikaa niiden käytöstä reaali maailmassa. He esittelevät tutkimuksessaan HabisDraw-ohjelman, jonka työkalujen käyttö on hyvin samanlaista kuin reaali maailmassa. Esimerkiksi piirtoalueella voi olla useita kynätyökaluja, jotka käyttäjä saa halutessaan käyttöön ja kynän väri vaihdetaan kastamalla sitä eri värisiin mustepulloihin. Ohjelmassa käytetään myös viivainta, jonka avulla suorat viivat saadaan piirrettyä ja nuppineulan tavoin toimivia neulatyökaluja, joilla piirretty kuvio saadaan kiinnitettyä tiettyyn paikkaan piirtoalueella. Reaali maailman työkalujen käytön mallintaminen piirto-ohjelmaan tuntuu ainakin tässä tapauksessa toimivan hyvin ja voisi kuvitella sen helpottavat tietokoneohjelman käytön oppimista vasta-alkajilla.

2.3. Kaavioiden muokkaaminen

Raisamon [2002] mukaan nykyisissä piirtosovelluksissa kaavioiden muokkaukseen on olemassa seuraavia menetelmiä: sommittelualgoritmit, komentopohjaiset työkalut, rajoitteet sekä suora-asemointi. Tämän luvun esimerkihjelmissä

kaavioiden muokkaaminen tapahtuu pääosin komentopohjaisten työkalujen sekä suora-asemoinnin avulla.

Ryall ja muut [1997] esittelevät tutkimuksessaan GLIDE (Graph Layout Interactive Diagram Editor)-piirto-ohjelman, jossa kaavioiden muokkaus tapahtuu vuorovaikutteisten rajoitteiden avulla. Käyttäjä määrittelee itse, mitä kahdeksasta muokkausperiaatteesta käyttää tietyille objekteille, ja ohjelma huolehtii siitä, että objektit ovat näiden periaatteiden mukaisesti aseteltuja, vaikka niitä muokattaisiin. Yksi näistä muokkausperiaatteista on objektien tasaaminen. Rajoitepohjaiset piirto-ohjelmat eivät ole kovin yleisiä, ja tutkimuksessa todetaankin että käyttäjillä on usein vaikeuksia ymmärtää miten ohjelma toimii ja miten sen ominaisuuksia voi hyödyntää.

Gleicher [1992] on tutkimuksessaan yhdistänyt rajoitteiden käyttämisen ja suorakäyttöisyyden. Tutkimuksessa esitellään Briar-piirto-ohjelma, jossa käytetään suorakäyttöisiä asetelutekniikoita yhdessä objektien väliset suhteet määrittävien rajoitteiden kanssa. Tutkimuksen mukaan nämä kaksi yhdistämällä saadaan käyttäjän kannalta miellyttävä käyttöliittymä, ja suhteiden ilmaisu pysyy selkeänä.

2.4. Objektien tasaaminen

Kaavioon piirrettävien objektien tasaustapa voi Robbinsin ja muiden [1999] mukaan olla yksi selventävä tekijä objektien välisten suhteiden käsittämiseen. Objektien tasaustapa voi muun muassa selventää niiden ryhmitystä ja loogista rakennetta. Yleisesti kaaviot ovat usein selkeämmän ja siistimmän näköisiä, jos ne on tasattu järkevästi ja symmetrisesti. Symmetrisyys tuo mihin tahansa kuvaan järjestelmällisemmän vaikutelman.

Joissain objektipohjaisissa piirto-ohjelmissa objektien tasaamiseen ei ole varsinaista työkalua, vaan niiden järjestäminen tapahtuu valitsemalla objekti valintatyökalulla ja raahaamalla se haluttuun paikkaan. Objektien tasaaminen tarkasti samaan linjaan on näin ollen vaikeaa. Joissain sovelluksissa, kuten FreeHand [2003] sekä CorelDraw [2003], käytetään tasaamisen apuna viivaimia piirtoalueen reunoilla tai piirtoalueelle ilmestyviä apuviivoja, joihin objektit saadaan tarttumaan. Nämä helpottavat tarkan lopputuloksen aikaansaamista.

Raisamo ja Räihä [1996, 2000] esittelivät tutkimuksessaan tasauspuikon, josta tämän tutkimuksen kohteena olevien työkalujen kehittäminen on lähtenyt liikkeelle. Alkuperäistä tasauspuikkoa käytettiin kahdella ohjaimella, hiirellä sekä ohjauspallolla, jolla pystyi säätämään puikon kokoa ja kulmaa sekä muuttamaan kursorin väliaikaisesti valintatyökaluksi. Tasauspuikko toimii siten, että sitä raahataan kohti objekteja, ja kun se koskettaa objektin reunaa tai sen tasauspisteitä, objekti tarttuu siihen ja työntyy sen mukana haluttuun paikkaan.

Myös Robbins ja muut [1999] ovat esitelleet tasaustyökalun, jonka toimintaperiaate on hyvin samankaltainen tasauspuikon kanssa.

Voisi luulla, että suoraikäyttöisyys lisääisi käyttäjän hiirellä tekemiä liikkeitä, ja näin lisääisi ylimääräistä räsitusta hiirikädelle, mutta esimerkiksi Robbinsin ja muiden [1999] tutkimuksessa todettiin, että suoraikäyttöisen tasaustyökalun käyttäminen jopa vähensi hiirellä tehtäviä raahauksia. Tässä tapauksessa käyttäjän ei tarvinnut valita erikseen objektia tasauksen kohteeksi piirtämällä valintasuorakaide sen ympärille.

3. Suoraikäyttöisyys piirtosovelluksissa

Tietokoneohjelmat ovat perinteisesti perustuneet dialogi-metaforaan, eli interaktio tapahtuu siten, että käyttäjä antaa komennon ja tietokoneohjelma antaa siihen vastauksen johon käyttäjä taas reagoi. Tämä metafora sopii hyvin komentorivipohjaisiin käyttöliittymiin, mutta uudemmantyyppiset käyttöliittymät ovat alkaneet pohjautua suoraikäyttöisyyden metaforaan [Hudson and King, 1986].

Tämän tutkimuksen kohteena olevat kaavionmuokkaustyökalut ovat suoraikäyttöisiä, eli käyttäjä saa välitöntä visuaalista palautetta toiminnoistaan käyttäessään työkaluja. Tässä luvussa tarkastellaan suoraikäyttöisyyden käsitettä sekä sen hyödyntämistä piirtosovelluksissa.

3.1. Suoraikäyttöisyyden käsite

Shneiderman [1992] selventää suoraikäyttöisyyden määritelmää esimerkillä auton ajamisesta. Ajaessaan autoa kuljettaja saa välitöntä palautetta toiminnoistaan kun hän esimerkiksi kääntää ohjauspyörää. Jos ohjauspyörää kääntää vasemmalle, kuljettaja näkee toimintonsa seuraukset välittömästi katsoessaan ulos kun auto kääntyy. Autolla ajaminen on siis varsin suoraikäyttöistä toimintaa. On vaikea kuvitella, että autoa voisi ohjata antamalla ensin kääntymiskomento ja sen jälkeen toinen komento, joka päivittäisi näkymän vastaamaan kääntymisen jälkeistä tilaa. Suoraikäyttöisyys (direct manipulation) tarkoittaa siis sitä, että käyttäjälle annetaan välitöntä visuaalista palautetta suorittamistaan toiminnoista, jolloin sekä objektit että käyttäjän suorittamat toiminnot ovat molemmat nähtävissä [Shneiderman, 1992]. Tämä tarkoittaa siis sitä, että toimintojen suorittaminen ohjelmassa tapahtuu mahdollisimman yhteneväisesti todellisessa maailmassa suoritettavien samankaltaisten toimintojen kanssa.

Tämän tutkimuksen työkaluissa suoraikäyttöisyys tarkoittaa sitä, että työkalut toimivat ikään kuin hiiren kursorina, jolla voidaan muokata piirrettyjä objekteja. Kun työkalu valitaan, hiiren kursori muuttuu valituksi työkaluksi.

Objektien muokkaus tapahtuu koskettamalla tai raahaamalla niitä käytössä olevalla työkalulla.

Shneiderman [1992] on listannut suorakäyttöisyyden käyttämisestä mahdollisesti aiheutuvia ongelmia. Niitä ovat muun muassa se, että järjestelmä vaatii enemmän resursseja sekä sen toiminnot muuttuvat kömpelöiksi. Lisäksi visualisointien havaitseminen voi olla vaikeaa näkörajoitteisille käyttäjille. Suorakäyttöisyys saa kuitenkin aikaan myös etuja. Käyttäjät oppivat ohjelman käytön nopeasti, virhetilanteet ovat paremmin ennakoitavissa ja käyttäjän toiminnot ja näytön välittämä kuva vastaavat toisiaan.

Kaikissa tapauksissa suorakäyttöisyys ei siis tee ohjelmasta parempaa. Käyttäjän näkökulmasta sen soveltaminen voi kuitenkin tuoda selviä etuja ja tehdä ohjelmasta helppokäyttöisemmän.

3.2. Suorakäyttöiset sovellukset

Shneiderman [1992] esittää että onnistuneen suorakäyttöisen ohjelman toteuttaminen edellyttää, että suunnittelun pohjana käytetään kunnollista mallia todellisesta maailmasta. Esimerkiksi tekstieditorien kehitys viime vuosikymmeninä on muuttunut selkeästi suorakäyttöisempään suuntaan. Komentopohjaisuudesta on siirrytty WYSIWYG (What You See Is What You Get)-periaatteella toimiviin ohjelmiin, mikä tarkoittaa sitä, että käyttäjä näkee jo kirjoittaessaan tekstin tulostettavassa muodossa. WYSIWYG-periaate on sittemmin myös muodostunut tekstieditorien standardiksi [Shneiderman, 1992].

Tekstin kirjoittamiseen ja muokkaamiseen tarkoitettujen tietokoneohjelmien lisäksi Shneiderman [1992] mainitsee suorakäyttöisten ohjelmien sovellusalueista erityisesti virtuaalitodellisuuden (VR) sovellukset. Suorakäyttöisyyden käyttäminen VR-sovelluksissa on hyvin perusteltua, sillä virtuaalitodellisuus pyrkii jäljittelemään todellista maailmaa, joten ihmisen vuorovaikutus sovelluksen kanssa tulisi olla mahdollisimman luonnollista. Myös videopelit ovat yksi kaupallisesti menestyneimmistä suorakäyttöisyyden sovellusalueista [Shneiderman, 1992].

4. Suorakäyttöiset kaavionmuokkaustyökalut

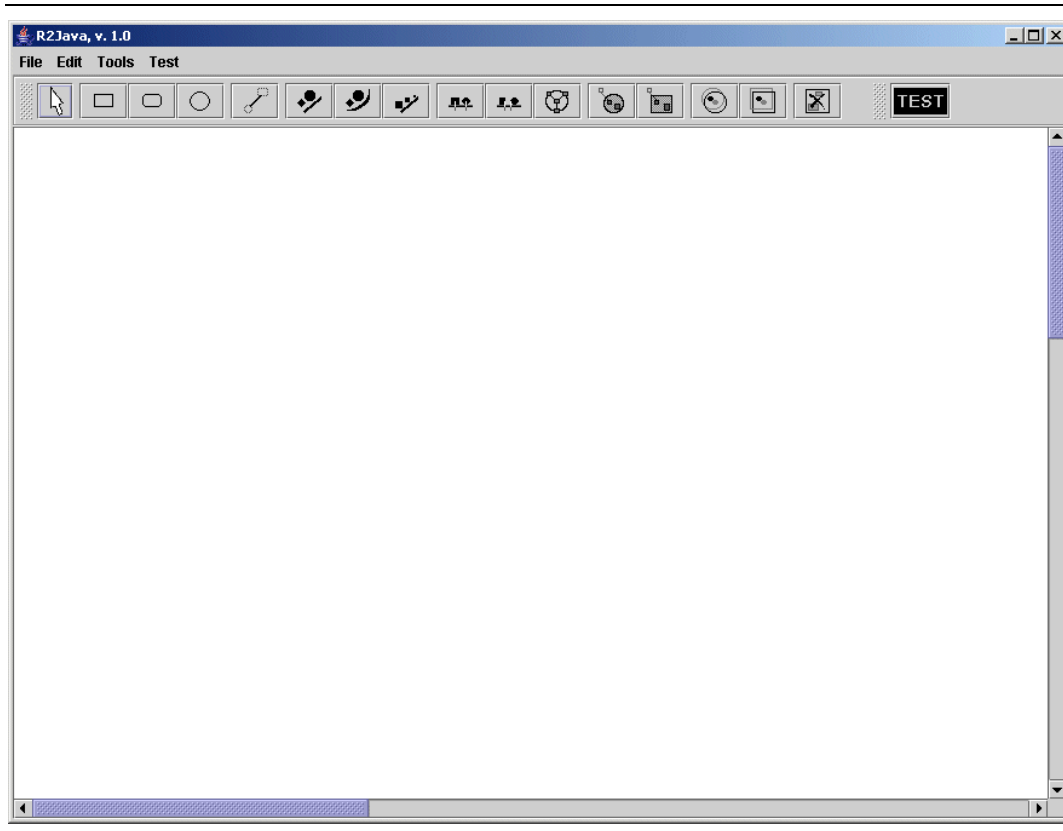
Tässä luvussa tarkastellaan tutkimuksen kohteena olevia suorakäyttöisiä kaavionmuokkaustyökaluja sekä R2Java - kaavionpiirto-sovellusta, jossa ne toimivat.

4.1. R2Java-sovellus

R2Java on piirto-ohjelma, joka on tarkoitettu erilaisten kaavioiden piirtämiseen. Se sisältää tavanomaisia piirtämistyökaluja sekä suorakäyttöisesti toimivia

kaavionmuokkaustyökaluja. Ohjelma on kehitetty Tampereen yliopiston tietojenkäsittelytieteiden laitoksella, ja se on toteutettu Java-ohjelmointikielellä.

R2Java on tarkoitettu erityisesti kaavioiden piirtämiseen ja niiden käsittelyyn. Ohjelmalla voi piirtää suorakaiteen, ellipsin sekä pyöristetyn suorakaiteen muotoisia objekteja ja yhdysviivoja näiden välille. Piirtäminen tapahtuu piirto-ohjelmille tavanomaisella tavalla, eli valitsemalla haluttu piirtotyökalu ja raa- haamalla hiirtä piirtoalustalla. Piirrosten muokkaaminen tapahtuu suorakäyt- töisillä muokkaustyökaluilla. Kuvassa 1 esitellään R2Java-ohjelman käyttöliit- tymä.



Kuva 1. R2Java-ohjelman käyttöliittymä

4.2. Kaavionmuokkaustyökalujen toiminnallisuus

Tutkimuksen kohteena olevat työkalut on jaettu tasaustyökaluihin (alignment tools) ja käsittelytyökaluihin (managing tools). Tasaustyökaluja ovat tasaus- puikko (alignment stick), tasauskaarri (alignment arc), tasausellipsi (alignment ellipse), koonsäätöpuikko (resizing stick) ja kaksi tasavälipuikkoa (distribution stick). Käsittelytyökaluihin kuuluvat ellipsin sekä suorakaiteen muotoiset kaappaustyökalut (grabbing tools) ja kopiointityökalut (copy tools) sekä poisto-

työkalu. Työkaluihin on lisätty tunto- sekä voimapalautetta, mutta niitä ei testattu tässä tutkimuksessa.

Tasaustyökalujen kehittäminen on lähtenyt liikkeelle Raisamon ja Rähän [1996, 2000] esittelemästä tasauspuikosta. Koonsäätöpuikko pohjautuu kutistuspuikkoon (shrinking stick), joka on esitelty Raisamon [1999] tutkimuksessa. Muista kaavionmuokkaustyökaluista kaappausellipsi, tasausellipsi sekä tasavälipuikot on esitelty Raisamon [2002] tutkimuksessa.

Kaavionmuokkaustyökalut toimivat siten, että kun työkalu valitaan työkaluriviltä, hiiren kursori muuttuu täksi työkaluksi. Hiiren oikeata painiketta painamalla käyttäjä voi muuttaa työkalun kokoa ja keskimmaisella painikkeella vaihtaa työkalun asentoa. Työkalua käytetään vasemmalla hiiren painikkeella. Tasauspuikossa sekä -kaareissa työkalun asentoa voi muuttaa vapaasti. Tasaväli- sekä koonsäätöpuikot ovat aina joko pysty- tai vaakasuorassa.

Tasaustyökaluja tasausellipsiä lukuun ottamatta käytetään raahaamalla työkalua ja koskettamalla objekteja. Tasaus- sekä tasavälipuikkoja käytettäessä objektiin tulee koskea koko sen reunan pituudelta, jotta se tasautuu. Kun objektia tasataan, sen väri muuttuu tummemmaksi. Tasavälipuikot tasaavat objektien väliin jäävät tai objektien keskipisteiden väliset etäisyydet saman suuruisiksi. Tasauspuikko sekä -kaari tasaavat objektit työkalun reunaan vasten. Koonsäätöpuikolla objektien kokoa voidaan muuttaa. Tasausellipsiä käytetään siten, että ellipsin keskikohta asetetaan keskimmäisen objektin päälle, jolloin työkalun keskelle ilmestyy käsikursori. Hiiren vasenta painiketta painamalla keskimmäisen objektin ympärillä olevat objektit tasataan ellipsin kehälle.

Kopiointi- ja kaappaustyökalut ovat ellipsin tai suorakaiteen muotoisia ja poistotyökalu suorakaiteen muotoinen. Kopiointityökalu on vihertävä, ja sitä käytetään siten, että työkalu viedään kopioitavien objektien päälle ja painetaan hiiren vasenta painiketta. Tällöin objektit, jotka ovat kokonaan työkalun reunojen sisäpuolella kopioituvat. Kopioiden väri muuttuu vihreäksi, ja kopiot voidaan raahata työkalulla uuteen paikkaan. Kaappaustyökalu toimii samalla periaatteella, ja siirtää objekteja sitä raahaamalla. Yhdysviivat pysyvät normaalisti kiinni objekteissa, mutta jos työkalua käytettäessä painetaan näppäimistön ctrl-näppäintä, ulkopuoliset viivat poistuvat ja objektit voidaan leikata pois kaaviosta. Poistotyökalu on punertava, ja se muuttaa poistettavien objektien värin punaiseksi, kun se viedään objektien päälle. Poistettavat objektit poistetaan painamalla hiiren vasenta painiketta.

5. Käytettävyydestit

Tutkimuksessa arvioidaan suorakäyttöisten kaavionmuokkaustyökalujen käytettävyyttä. Preece ja muut [1994] toteavat että käytettävyyden arviointi on yksi

käyttäjakeskeisen ohjelmistosuunnittelun keskeisistä osa-alueista. Käyttäjakeskeisen suunnittelun tarkoituksena on saada tietokoneohjelmasta helppokäyttöinen ja tehokas järjestelmä, jonka käyttäminen kohderyhmäkäyttäjien on helppo oppia.

Tässä tutkimuksessa käytettävyyttä selvitetään käytettävyydestien avulla. Savage [1996] vertaili kolmea käytettävyyden arviointitekniikkaa: asiantuntijoiden tekemää arviointia, käyttäjien tekemää arviointia sekä käytettävyydestausta. Tutkimuksen mukaan käyttöliittymään liittyvät ongelmat tulivat parhaiten ilmi käytettävyydestesteissä. Asiantuntijoiden tekemissä käytettävyyssarvioinneissa tuli ilmi lähinnä vain uusia käytettävyydestutkimusta vaativia kohtia ohjelmassa. Tämän perusteella käytettävyydestaus vaikuttaa sopivimmalta tavalta arvioida näiden työkalujen käytettävyyttä. Tässä luvussa kerrotaan tarkemmin tässä tutkimuksessa suoritetuista käytettävyydestesteistä.

5.1. Testijärjestelyt

Käytettävyydesteihin osallistui yhdeksän koehenkilöä, joista kuusi naisia ja kolme miestä. Iältään koehenkilöt olivat 22–24 -vuotiaita, ja he olivat kaikki tottuneita tietokoneohjelmien käyttäjiä. Kaikki koehenkilöt olivat aikaisemmin käyttäneet jotain piirto-ohjelmaa, mutta keskimäärin vain harvoin. Yksi koehenkilö oli aikaisemmin käyttänyt joitain suorakäyttöisiä työkaluja. Testit järjestettiin Tampereen yliopiston tietojenkäsittelytieteiden laitoksen käytettävyysslaboratoriossa, ja ne videoitiin tarkempaa analysointia varten.

Ennen käytettävyydestiä koehenkilölle esiteltiin R2Java-ohjelmaa sekä näytettiin, miten testattavia muokkaustyökaluja käytetään ja mitä niillä voi tehdä. Koehenkilö sai halutessaan myös itse kokeilla työkaluja jonkin aikaa, jotta hän tottuisi hieman niiden käyttöön. Työkalujen ja ohjelman toimintaan tutustumiseen kului keskimäärin viisi minuuttia. Koehenkilöille kerrottiin, että heidän tulisi testitehtävissä käyttää suorakäyttöisiä muokkaustyökaluja. Nuolityökalun toimintaa ei esitelty eikä sitä mainittu.

Testitehtävissä koehenkilön tuli muokkaustyökaluja käyttämällä muokata alkupiiirros vastaamaan paperilla annettua lopputulosta. Jokaiseen tehtävään liittyvä alkupiiirros oli tallennettu aikaisemmin, joten käyttäjän ei tarvinnut luoda itse piirroksia. Koehenkilö sai itse päättää, koska piirros on valmis, ja se tallennettiin tehtävän suorittamisen jälkeen.

Testitehtävien suorittamisen jälkeen koehenkilöitä pyydettiin vielä täyttämään loppukyselylomake, jossa kysyttiin mielipiteitä työkaluista sekä selvitetiin heidän kokemustaan tietokoneiden käytöstä. Koehenkilöiltä pyydettiin myös vapaamuotoisia kommentteja ohjelmasta sekä parannusehdotuksia.

5.2. Testitehtävät

Kaikki testitehtävät olivat samantyyppisiä. Koehenkilön tuli muokata alkupiirros näyttämään samalta kuin paperilla annettu lopputulos. Alkupään tehtävissä tarkoituksena oli käyttää vain yhtä työkalua, ja vähitellen käytettävien työkalujen määrää lisättiin ja näin vaikeutettiin tehtäviä. Testitehtäviä oli yhteensä 11, ja kaikkien tehtävien suorittamiseen koehenkilöiltä kului keskimäärin 20 minuuttia.

Testitehtävien tarkoituksena oli selvittää, pitävätkö koekäyttäjät työkaluja helppokäyttöisinä ja miten nopeasti he oppivat työkalujen käytön. Testitehtäviä suunniteltaessa oli mietitty, mitä työkalua tehtävässä olisi tarkoitus käyttää, joten tarkoituksena oli myös saada selville, miten käyttäjät osaavat hyödyntää työkalujen ominaisuuksia ja valita sopivan työkalun tehtävää varten. Taulukossa 1 näkyy, mitä työkaluja tehtävissä oli tarkoitus käyttää ja mitä muokaus työkaluja koehenkilöt yleisimmin (vähintään kolme henkilöä) käyttivät.

Tehtävä	Testatut työkalut	Yleisimmin käytetyt työkalut
1	tasauspuikko	tasavälipuikko, tasauspuikko
2	tasavälipuikko	tasavälipuikko, tasauspuikko
3	koonsäätöpuikko	tasavälipuikko, tasauspuikko, koonsäätöpuikko
4	tasausellipsi	tasausellipsi
5	kopiointityökalu	kopiointityökalu, tasauspuikko, tasavälipuikko
6	kaappaustyökalu	tasauspuikko, tasavälipuikko, kaappaustyökalu
7	kopiointityökalu, tasauspuikko	tasauspuikko, kopiointityökalu, kaappaustyökalu, tasavälipuikko
8	tasauskaari, koonsäätöpuikko, tasauspuikko	tasauskaari, kaappaustyökalu, tasauspuikko, koonsäätöpuikko, tasausellipsi
9	tasausellipsi, kopiointityökalu, tasavälipuikko	kopiointityökalu, tasauspuikko, tasavälipuikko, tasausellipsi
10	kopiointityökalu, kaappaustyökalu	kopiointityökalu, poistotyökalu, tasavälipuikko, kaappaustyökalu, tasauspuikko
11	ei määritelty	poistotyökalu, tasauspuikko, kopiointityökalu, tasavälipuikko

Taulukko 1. Testitehtävissä käytetyt työkalut

6. Käytettävyydestien tulokset

Käytettävyydestejä analysoitaessa keskityttiin erityisesti käyttäjien valitsemiin työkaluihin sekä tehtävien tekemiseen kuluneeseen aikaan. Keskimääräisesti vähiten aikaa koehenkilöiltä kului tehtävään 5 (47s) ja eniten tehtävään 10 (3min 58s).

6.1. Tehtävien suorittaminen

Käytettävyydesteissä koehenkilöiden oli tarkoitus käyttää suoraikäyttöisiä kaavionmuokkaustyökaluja annettujen piirrosten muokkaamiseen. Ohjelmassa oli myös perinteinen nuolityökalu, jolla objekteja voi siirtää sekä niiden kokoa muuttaa. Koehenkilöitä ei erikseen kielletty käyttämästä nuolityökalua, mutta sitä ei esitelty eikä mainittu ohjelman muita työkaluja esiteltäessä, ja heille kerrottiin, että tarkoitus olisi käyttää suoraikäyttöisiä työkaluja.

Kolme koehenkilöä yhdeksästä ei käyttänyt nuolityökalua missään vaiheessa, vaan suoritti kaikki muokkaukset suoraikäyttöisillä työkaluilla. Kuusi koehenkilöä käytti nuolityökalua ainakin yhdessä tehtävässä objektien siirtämiseen ja heistä kaksi myös objektien koon muuttamiseen. Alkuperäinen tarkoitus oli, että nuolityökalun työkaluikoni olisi poistettu ohjelman käyttöliittymästä käytettävyydestien ajaksi, mutta se päätettiin jättää paikoilleen, sillä sitä tarvitaan, jos piirroksista halutaan poistaa yhdysviivoja.

Viidessä tehtävässä ne koehenkilöt, jotka käyttivät tehtävään tarkoitettuja työkaluja, suorittivat tehtävän selvästi nopeammin kuin ne, jotka käyttivät nuolityökalua. Muissa tehtävissä selkeätä eroa koehenkilöiden käyttämien työkalujen välillä ei syntynyt. Käyttäessään nuolityökalua yksi koehenkilöistä sanoi käyttävänsä mieluummin vanhoja tuttuja menetelmiä kuin uusia työkaluja.

Koehenkilöt osasivat lähes aina valita tehtävään sopivan työkalun, lukuun ottamatta koonsäätöpuikkoa ja kaappaustyökalua, joita ei juurikaan käytetty tehtävissä, joissa niitä oli tarkoitettu käytettävän. Objektien koon muuttamisen sijasta muutamat koehenkilöt poistivat objektin ja ottivat sopivamman kokoisesta objektista kopion sen tilalle. Kaappaustyökalun sijasta käytettiin usein kopiointityökalua ja poistotyökalua. Koehenkilöt, jotka käyttivät nuolityökalua, suorittivat objektien siirtämisen yleensä sen avulla.

Koehenkilöt pystyivät valitsemaan oikean työkalun yleensä nopeasti. Vain muutamassa tapauksessa koehenkilö valitsi vahingossa väärän työkalun, mutta tällöinkin osasi korjata virheensä. Vääriä valintoja tapahtui erityisesti kaappaus- ja kopiointityökalujen kohdalla, ja koehenkilöt korjasivat virheensä pikaisesti huomattuaan, että työkalu on väärän värinen. Kaksi koehenkilöä päätyivät kahdessa eri tehtävässä poistamaan alkupiirroksen kokonaan ja piirtämään halutun lopputuloksen mukaisen piirroksen ohjelman piirtotyökaluilla. Näistä

toinen koehenkilö ei muistanut koonsäätöpuikon olemassaoloa, ja piirsi alkuperäisiä pienemmät objektit. Toisessa tapauksessa taas koehenkilö oli muokannut alkupiirrosta jo kauan, eikä ollut päässyt haluttuun lopputulokseen, joten hän totesi saavansa kuvasta halutunnäköisen nopeammin piirtämällä koko kuvan itse.

6.2. Työkalujen käyttäminen

Eniten koehenkilöt (vähintään kolme henkilöä) käyttivät tasauspuikkoa (10 tehtävässä) ja tasavälipuikkoja (9 tehtävässä). Kahdesta tasavälipuikosta enemmän käytettiin objektien väliset etäisyydet tasaavaa puikkoa. Vähiten käytettiin tasauskaarta, sillä vain yhdessä tehtävässä sille oli järkevää käyttöä. Muita työkaluja käytettiin 2-5 tehtävässä.

Tauspuikon sekä -kaaren käyttö sujui koehenkilöiltä sujuvimmin. Tausellipsiä käytettiin osittain väärissä yhteyksissä, mutta sen käyttöä varsinaisesti testaavassa tehtävässä kaikki koehenkilöt osasivat käyttää sitä. Koonsäätöpuikkoa käytettiin vain vähän. Osa koehenkilöistä ei käyttänyt sitä yhdessäkään tehtävässä, vaikka olivat etukäteen kokeilleet sitä. Yksi koehenkilö sanoi, ettei enää testitilanteessa muistanut, että koonsäätöpuikolla olisi voinut muuttaa objektien kokoa.

Käsittelytyökaluista koehenkilöt käyttivät eniten kopiointityökaluja. Kaappaustryökaluja käytettiin lähinnä vain yksittäisten objektien siirtämiseen. Työkalun mahdollistamaa monen objektin yhtäaikaista siirtämistä ei juuri hyödynnetty eikä yksikään koehenkilö käyttänyt sitä objektien leikkaamiseen. Koehenkilöt käyttivät tasaisesti sekä suorakaiteen että ellipsin muotoisia työkaluja. Poistotyökalun käyttäminen oli sujuvaa, sillä turhia objektien poistoja ei tapahtunut. Yksi koehenkilö yritti poistaa poistotyökalulla myös yhdysviivoja, mutta se ei ollut mahdollista.

Tehtävien suoritusaikoihin niissä käytetyillä työkaluilla ei ollut merkittävää vaikutusta. Tehtäviin tarkoitetuilla työkaluilla ei suoriuduttu tehtävistä nopeammin kuin muita suorakäyttöisiä työkaluja käyttämällä. Alkupään tehtävissä koehenkilöt käyttivät lähes aina samoja työkaluja, mutta myöhemmissä tehtävissä ilmeni enemmän eroja koehenkilöiden käyttämien työkalujen välillä.

6.3. Ongelmat ohjelman käyttämisessä

Ohjelman käyttäminen ei aiheuttanut koehenkilöille suuria ongelmia, sillä he olivat tottuneita tietokoneohjelmien käyttäjiä. Joitain pieniä tehtävien suorittamista haittaavia hankaluuksia kuitenkin esiintyi.

Muutamille koehenkilöille aiheutti hämmennystä se, että uuden tehtävän alussa mikään ohjelman työkalu ei ollut käytössä, vaikka viimeisenä käytetyn työkalun ikoni työkalurivillä näyttikin olevan valittuna. Työkalu piti valita uu-

delleen, jotta sitä pystyi käyttämään. Tämä ongelma ilmeni vain niissä tapauksissa, joissa koehenkilö halusi käyttää tehtävän alussa samaa työkalua kuin edellisen tehtävän lopuksi, joka näytti olevan valittuna.

Ohjelman piirtoalueen päivitys oli ajoittain todella hidasta. Joitain koehenkilöitä tämä häiritsi, ja he ihmettelivät piirtoalueelle esimerkiksi objektien rauhauksen yhteydessä tulevia ylimääräisiä viivoja, jotka katosivat kun piirtoaluetta päivitettiin. Vierityspalkin päivittymisessä oli myös ongelmia yhden koehenkilön kohdalla. Hän ei päässyt muokkaamaan piirtoalueen yläosassa olevaa objektia, josta osa oli piirtoalueen ulkopuolella, sillä vierityspalkkia ei pystynyt vierittämään ylöspäin.

Lähes jokaisella koehenkilöllä oli ainakin jossain vaiheessa ongelmia tasavälipuikkojen asennon muuttamisessa. Hiiren keskimäinen painike ei aina vaihtanut puikon asentoa, ja joissain tapauksissa koehenkilö vaihtoi työkalua, kun ei saanut puikkoa käännettyä haluamaansa suuntaan. Muilla työkaluilla tätä ongelmaa ei esiintynyt, joten ongelma johtui luultavasti tasavälityökalujen toteutuksesta.

Ohjelman käytössä esiintyi myös muutamia yksittäisiä ongelmia: yksi objekti hävisi hieman sen piirtämisen jälkeen, tasauskaari ei tasannut yhtä objektia, johon osui ja tasauspuikko tasasi objektin väärältä puoleltaan. Näitä ongelmia ei analysoida tarkemmin, sillä tutkimuksessa keskityttiin pääasiassa suora-käyttöisiin työkaluihin.

6.4. Ongelmat työkalujen toiminnallisuudessa

Ennen testejä koehenkilöille oli esitelty kaikki suora-käyttöiset työkalut, ja he olivat saaneet itse kokeilla niitä, joten heillä kaikilla oli hieman kokemusta työkalujen käyttämisestä. Tästä syystä ylitsepääsemättömiä ongelmia työkalujen käyttämisessä ei ollut, ja koehenkilöt osasivat lähes jokaisessa tapauksessa käyttää työkaluja oikein.

Eniten ongelmia koehenkilöillä oli tasavälipuikkojen käyttämisessä. Ongelmat olivat kahdenlaisia: objektien paikat vaihtuivat eri järjestykseen sen mukaan miten puikko osui niihin ja objekteja yritettiin tasata liian leveällä puikolla, jolloin ne hajautuivat liian laajalle. Ensimmäinen ongelma ilmeni erityisesti puurakenteiden objekteja tasattaessa, kun koehenkilö tasasi kahden solmun lehtisolmuja yhtäaikaan. Jos koehenkilö osui puikolla objekteihin epätasaisesti, niiden paikat saattoivat vaihtua ja toisen solmun lehdet siirtyä toisen solmun alle. Virhettä yritettiin korjata yleensä siirtämällä solmut takaisin alkuperäisille paikoilleen ja yrittämällä uudestaan. Useimmat koehenkilöt päätyivät lopulta tasaamaan vain yhden solmun lehtisolmut kerrallaan, jolloin ongelmaa ei esiintynyt. Toinen ongelma oli, että koehenkilöt yrittivät tasata objekteja liian leveällä puikolla, jolloin objektit levisivät koko puikon leveydelle. Virhettä yri-

tettiin yleensä korjata siten, että he siirsivät objekteja lähemmäksi toisiaan, mutta pitivät puikon edelleen leveänä. Erehdyksien kautta kaikki koehenkilöt lopulta keksivät, että objektien tasauksen välimatkat riippuvat puikon leveydestä.

Kopiointityökalun yhteydessä esiintyi myös ongelmia useammassa tehtävässä. Ongelmia oli kahdenlaisia: koehenkilöt kopioivat vahingossa ylimääräisiä objekteja ja joskus työkalu oli liian pieni, eikä objekti kopioitunut. Ensimmäistä ongelmaa esiintyi silloin, kun koehenkilö kopioi objektin, eikä siirtänyt sitä heti pois alkuperäisen objektin läheisyydestä. Tällöin, kun koehenkilö halusi kopioida toisen samanlaisen objektin, myös kopioitu objekti oli edelleen työkalun vaikutuspiirissä, joten sekä alkuperäinen että kopioitu objekti kopioituivat. Toinen ongelma johtui siitä, että kopiointityökalun tuli olla jonkin verran suurempi kuin kopioitava objekti, jotta objekti kopioitui. Molemmat ongelmat voitaisiin poistaa siten, että työkaluun lisättäisiin poistotyökalun kaltainen ominaisuus, eli kun työkalun siirtäisi objektien päälle, sen vaikutuspiirissä olevien objektien väri muuttuisi vihreäksi, jotta käyttäjä tietäisi, mitkä objektit tulevat kopioitumaan hiiren vasenta painiketta painettaessa.

Puikonmuotoisten työkalujen eriävä toimintatapa aiheutti myös jonkin verran ongelmia. Koehenkilöt yrittivät muuttaa koonsäätö- sekä tasavälipuikkojen kulmaa vapaasti hiiren keskimmaisella painikkeella, sillä tasauspuikossa on kyseinen toiminto. Puikkojen käytössä aiheutti hämmennystä myös se, että koonsäätöpuikko vaikuttaa objekteihin, vaikka se ei osuisi niihin kokonaan, kun taas muiden puikkojen tulee osua koko objektiin. Epäselvyyksiä aiheutti myös se, että koonsäätöpuikko vaikuttaa objekteihin molemmilta puoliltaan, kun muissa puikoissa vain toinen puoli suorittaa toiminnon. Kaikkien puikonmuotoisten työkalujen käyttötavan yhtenäistäminen poistaisi nämä ongelmat.

Tasausellipsiä ensimmäistä kertaa käytettäessä kahdeksan koehenkilöä muutti ellipsin koon ensin liian isoksi, sillä he eivät hahmottaneet sitä, että objektit tasataan ellipsin kehälle eikä sen sisälle. Kaikki osasivat kuitenkin korjata virheensä ja pienentää työkalua. Viisi koehenkilöä yritti käyttää tasausellipsiä myös puurakenteen yhteydessä yrittäessään asettaa solmun lehtisolmuja tasaisesti yhtä kauaksi solmusta ja toisistaan. Työkalun käyttökohde ei siis ollut täysin selvä kaikille.

6.5. Koehenkilöiden mielipiteet

Koehenkilöiltä kysyttiin mielipiteitä työkaluista loppukyselylomakkeessa. Lomakkeessa oli 14 väittämää, joihin vastattiin ympyröimällä omaa mielipidettä vastaava vaihtoehto asteikolla 1 (vahvasti eri mieltä) - 5 (vahvasti samaa mieltä). Koehenkilöiden mielipiteissä esiintyi paljon hajontaa. Lähes jokainen väittämä sai arvoja välillä 2 - 5.

Lomakkeen vastauksista kävi ilmi, että koehenkilöt voisivat kuvitella käyttävänsä testattuja työkaluja kaavioiden laatimiseen (keskimääräinen arvosana 4,6) ja työkalut sopivat heidän mielestään hyvin kaavionkäsittelyyn (4,4). Ohjelma (4,2) ja työkalut (3,9) olivat koehenkilöiden mielestä kohtuullisen helppokäyttöisiä ja työkalujen toimintaidea ymmärrettiin myös hyvin (4,2). Tyytymättömmimpiä koehenkilöt olivat työkaluja kuvaavien ikonien havainnollisuuteen (3,1), mutta työkalukursorit kuvasivat työkalujen toimintaa hieman paremmin (3,6). Kyselyn mukaan koehenkilöiden oli melko helppo valita tehtävään sopiva työkalu (3,3) ja haluttu työkalukin löytyi työkalurivistä kohtuullisen helposti (3,2).

Koehenkilöiltä kysyttiin loppukyselylomakkeessa myös vapaamuotoisia kommentteja. He mainitsivat siitä, että tasavälipuikon käyttäminen ja työkalujen koon muuttaminen hiiren oikealla painikkeella on vaikeaa, ja että ohjelmassa olisi hyvä olla peruutustoiminto. Eräs koehenkilö ehdotti, että työkalut olisi hyvä saada palautettua alkuperäiseen asentoonsa sekä kokoonsa. Positiivista kommenttia kirjoitettiin tasaustyökalujen hyödyllisyydestä.

7. Lopuksi

Tutkielman tarkoituksena oli selvittää suoraikäyttöisten kaavionmuokkaustyökalujen käytettävyyttä käytettävyydestien avulla. Tässä luvussa kerrotaan tutkimuksen tulokset sekä ehdotetaan näiden perusteella parannuksia työkaluihin ja aiheita jatkotutkimuksiin.

Suoritettujen käytettävyydestien perusteella kaavionmuokkaustyökalut osoittautuivat kohtuullisen helposti käytettäviksi. Koekäyttäjät ymmärsivät nopeasti työkalujen käyttötavan ja osasivat lähes jokaisessa tehtävässä käyttää niitä oikein. Joidenkin työkalujen toiminnallisuudessa ilmeni pieniä käytettävyysongelmia, mutta ylitsepääsemättömiä vaikeuksia niiden käyttämisessä ei ollut. Tehtävään sopivan työkalun valinta sujui koehenkilöiltä myös nopeasti.

Tutkimuksen perusteella ilmeni ainakin kaksi seikkaa, jotka parantaisivat työkalujen käytettävyyttä. Kaappaus- ja kopiointityökaluihin tulisi lisätä poistotyökalun kaltainen ominaisuus, jolloin muokkauksen kohteena olevat objektit vaihtaisivat väriä jo silloin, kun työkalu tuodaan objektien päälle. Tällöin käyttäjä näkisi selvemmin, mihin objekteihin työkalu vaikuttaa jo ennen sen käyttämistä, mikä vähentäisi virheellisiä muokkauksia. Toinen muutos kohdistuu puikonmuotoisiin työkaluihin. Niiden käyttötapaa tulisi yhtenäistää, sillä käyttäjät eivät muista, mikä puikko toimii milläkin tavalla. Kaikkien puikkojen tulisi toimia vain toiselta reunalta ja mahdollisesti vapaata kulmansäätöäkin pitäisi harkita lisättäväksi kaikkiin puikkoihin.

Tasavälipuikon käyttämiseen liittyvään objektien paikkojen vaihtumiseen on vaikea keksiä ratkaisua. Luultavasti ongelma poistuisi vain harjoittelemalla enemmän työkalun käyttöä, jolloin käyttäjä osaisi tarttua objekteihin oikeassa järjestyksessä. Työkalun toimintatapaa ei kannata tämän takia lähteä muuttamaan. Itse ohjelmaan voisi tehdä kaksi muutosta: työkaluja kuvaavia ikoneita voisi yrittää muokata hieman kuvaavammiksi sekä lisätä painikkeen, joka palauttaisi työkalun alkuperäiseen asentoon ja kokoon.

Tämän tutkimuksen pohjalta työkalujen testaamista kannattaa jatkaa ainakin vertailemalla niiden käyttöä muiden piirto-ohjelmien työkalujen kanssa. Työkalut ovat osoittautuneet hyödyllisiksi, ja olisi mielenkiintoista selvittää, lyhentääkö niiden käyttäminen suoritusajoja muiden piirto-ohjelmien työkaluihin verrattuna. Työkalujen käyttöä kannattaisi testata myös siinä tilanteessa, että nuolityökalu olisi poistettu ohjelmasta. Tällöin kaavioita voisi muokata vain testattavilla työkaluilla. Myös työkaluihin liitettyä tunto- ja voimapaletta voisi testata muiden testien yhteydessä.

Viiteluettelo

- [ArgoUML, 2003] ArgoUML 0.12, <http://argouml.tigris.org/>, tarkastettu 25.11.2003.
- [CorelDraw, 2003] CorelDraw 9, <http://www.coreldraw.com>, tarkastettu 25.11.2003.
- [FreeHand, 2003] Macromedia FreeHand 8, <http://www.macromedia.com/>, tarkastettu 25.11.2003.
- [Fujaba, 2003] Fujaba Tool Suite 4.0, <http://www.uni-paderborn.de/cs/fujaba/>, tarkastettu 25.11.2003.
- [Gleicher, 1992] Michel Gleicher, Integrating constraints and direct manipulation. *Proceedings of SI3D'92, symposium on Interactive 3D graphics*, ACM Press, 1992, 171-174.
- [Hudson and King, 1986] Scott E. Hudson and Roger King, A generator of direct manipulation office systems. *ACM Transactions on Information Systems* 4 (2), 1986, 132-163.
- [Meyer et. al., 1994] Shawna Meyer, Oryx Cohen and Erik Nilsen, Devise comparisons for goal-directed drawing tasks. *Conference companion of CHI'94, conference on Human factors in computing systems*, ACM Press, 1994, 251-252.
- [Preece et. al., 1994] Jenny Preece, Yvonne Rogers, Helen Sharp, David Benyon, Simon Holland and Tom Carey, *Human-computer interaction*. Addison-Wesley, 1994, osa VI.

- [Raisamo, 2002] Jukka Raisamo, Tuntopalautteen tukemat suorakäyttöiset kaavionmuokkaustyökalut. Pro Gradu -tutkielma, Tampereen yliopisto, kesäkuu 2002. Saatavissa myös http://www.cs.uta.fi/research/theses/masters/Raisamo_Jukka.pdf, tarkastettu 25.11.2003.
- [Raisamo, 1999] Roope Raisamo, An alternative way of drawing. *Proceedings of CHI'99, conference on Human factors in computing systems*, ACM Press, 1999, 175-182.
- [Raisamo and Räihä, 1996] Roope Raisamo and Kari-Jouko Räihä, A new direct manipulation technique for aligning objects in drawing programs. *Proceedings of UIST'96, ACM symposium on User interface software and technology*, ACM Press, 1996, 157-164.
- [Raisamo and Räihä, 2000] Roope Raisamo and Kari-Jouko Räihä, Design and evaluation of the alignment stick. *Interacting with computers* **12**, 2000, 483-506.
- [Robbins et. al., 1999] Jason Robbins, Michael Kantor and David Redmiles, Sweeping away disorder with the broom alignment tool. *Extended Abstracts of CHI'99, conference on Human factors in computing systems*, ACM Press, 1999, 250-251.
- [Ryall et. al., 1997] Kathy Ryall, Joe Marks and Stuart Shieber, An interactive constraint-based system for drawing graphs. *Proceedings of UIST'97, ACM symposium on User interface software and technology*, ACM Press, 1997, 97-104.
- [Savage, 1996] Pamela Savage, User interface evaluation in an iterative design process: a comparison of three techniques. *Proceedings of CHI'96, conference on Human factors in computing systems*, ACM Press, 1996, 307-308.
- [Shneiderman, 1992] Ben Shneiderman, *Designing the user interface, 2nd edition*. Addison-Wesley, 1992, luku 5 Direct Manipulation.
- [SmartDraw, 2003] SmartDraw 6, <http://www.smartdraw.com/>, tarkastettu 26.11.2003.
- [St. Amant and Horton, 2002] Robert St. Amant and Thomas E. Horton, Characterizing tool use in an interactive drawing environment. *Proceedings of the symposium on Smart graphics*, ACM Press, 2002, 86-93.

Dynaamisesti ladattavan koodin tietoturva

Timo Linden

Tiivistelmä.

Tutkielmassa käsitellään dynaamisesti ladattavaan koodin liittyviä riskejä, esitellään ja arvioidaan dynaamisesti ladattavan koodin eri toteutustekniikoita ja toteutetaan tietoturvaratkaisu esimerkkiarkkitehtuuriin. Tutkielmassa tullaan pääosin paneutumaan Java-kielen tarjoamiin mahdollisuuksiin.

Avainsanat ja -sanonnat: dynaaminen luokanlataus, mobiilikoodi, turvallisuus.
CR-luokka: D.2.4

1. Johdanto

Dynaamisesti ladattava koodi tarkoittaa koodia, joka ladataan vasta pääohjelman ajon aikana. Koodi voidaan ladata joko paikalliselta levyltä tai verkosta. Siitä käytetään myös nimiä: mobiilikoodi, ladattava koodi ja kaukokoodi. Koodi voi olla lähdekoodia, binäärikoodia tai jotain näiden kahden väliltä.

Dynaamisesti ladattava koodi mahdollistaa ohjelmiston hajauttamisen ja tarjoaa näin monia etuja ja mahdollisuuksia. Ohjelmiston toimintaa voidaan muuttella ladattavilla moduuleilla, jolloin esimerkiksi saman ohjelmiston eri käyttäjien toiveet voidaan ottaa paremmin huomioon. Dynaamisesti ladattavaa koodia ovat esimerkiksi Java-apletit, JavaScript-ohjelmat ja ActiveX-komponentit.

Dynaamisesti ladattavaan koodiin, varsinkin jos se on tuntemattoman ihmisen tekemä, liittyy selvä riski. Jos koodissa on virus, se voi saastuttaa isäntäkoneeni. Jos kyseessä on puolestaan Troijan hevonen, se voi ottaa koneesi haltuun näytellen samalla viatonta. WWW:n myötä olemme entistäkin haavoittuvammiksi hyökkäyksille haitallisen koodin kautta. Viimeaikoina riehuneet blaster-madot ovat hyvä esimerkki aiheen vakavuudesta. Käyttäjien toimia vakoilevat ohjelmat ovat myös vakava tietoturva-aukko. Myös luotettavasta lähteestä ladatussa koodissa saattaa olla tahattomasti tehtyjä virheitä, jotka voivat aiheuttaa ongelmia.

Nykyisissä käyttöjärjestelmissä ajettavien ohjelmien turvallisuutta vaarantavat toiminnot käyttävät aina käyttöjärjestelmän palveluita. Turvallisuuslinjauksilla pyritään määrittelemään mitä palveluita ja resursseja kullekin ohjelmalle suodaan. Turvallisuuslinjauksiin kuuluvat mm. ohjelmalle

suotava suoritusaika, ohjelmalle suotava muistimäärä, tiedostojen ja hakemistojen käyttöoikeudet ja verkkotoiminnot. Nykyisissä käyttöjärjestelmissä ei ole kuitenkaan valmista ratkaisua tuntemattoman koodin turvalliseen suoritukseen. Ilman kunnollista suoritussympäristöä, käyttäjä joutuu aina valitsemaan ajaako hän ohjelman riskillä vai torjuuko ehkä hyödyllisenkin ohjelman toiminnan.

Seuraavassa luvussa esitellään erilaisia ratkaisumalleja dynaamisesti ladatun koodin turvalliseen suoritukseen. Kolmannessa luvussa käydään läpi dynaamisen luokanlataamisen mahdollistavia ohjelmointikieliä ja neljännessä luvussa paneudutaan esimerkkiarkkitehtuuriin.

2. Ratkaisumalleista

2.1. Yleistä

Dynaamisesti ladatun koodin turvallisuutta on tutkittu paljon ja useita ratkaisumalleja on esitetty. Yleensä on jouduttu tinkimään tehosta ja käyttömukavuudesta turvallisuuden takaamiseksi. Ratkaisumallit voidaan jakaa kahteen osaan; joko staattiseen käynnöksen aikana tai dynaamisesti ajon aikana tehtävään tarkistukseen.

Dynaamisessa tarkistuksessa ohjelman suoritusta tarkkaillaan ja sen suoritus voidaan lopettaa jos se käyttäytyy haitallisesti. Tämä voi olla käyttäjälle varsin epämukavaa, varsinkin jos ohjelma lopetetaan kesken tärkeän työn. Tekniikka mahdollistaa koodin suorittajan itse määrittellä mihin seikkoihin koodin suorituksessa puututaan. [Sekar *et al.*, 2003]

Staattisessa tarkistuksen läpäisseen koodin ajoon ei puututa vaan tarkistus tehdään tutkimalla ohjelman koodia. Tehokkaaseen tarkistukseen tarvitaan kuitenkin aina ohjelman lähdekoodi, koska binäärimuotoista koodia on vaikeita tarkistaa. Dynaamisesti ladattu koodi on kuitenkin usein saatavilla vain binääri- tai muissa ei-lähdekoodimuodossa, joten staattinen tarkistus ei tarjoa kovin käytännöllistä ratkaisua. Tarkistus voidaan suorittaa sekä koodin tekijän että koodin suorittajan toimesta. Ensimmäisessä vaihtoehdossa haittapuolena on se, että koodin suorittaja ei voi vaikuttaa tärkeinä pitämiinsä turvallisuusseikkoihin. Jälkimmäinen vaihtoehto voi olla liian hidasta ja vaivalloista. Parhaimmat ratkaisumallit yhdistävätkin nämä kaksi tekniikkaa. [Sekar *et al.*, 2003]

2.2. Sähköinen allekirjoitus

Koodin sähköisellä allekirjoituksella voidaan varmistua koodin alkuperästä. Ladattavaan koodin mukaan liitetään todistus, josta selviää kuka koodin on

kirjoittanut ja koska se on kirjoitettu. Sähköisen allekirjoituksen jälkeen koodia ei voida muuttaa.

Sähköinen allekirjoitus on hyödyllinen vain jos koodi on peräisin luotettavalta taholta. Tuntemattoman tahon tekemään koodiin ei ole luottamista vaikka se olisikin sähköisesti allekirjoitettu.

Eräs sähköisen allekirjoitukseen perustuva malli on ottaa mukaan kolmas osapuoli, jolle koodin tekijä lähettää tuotoksensa. Tämä kolmas osapuoli tarkistaa koodin ja merkitsee sen suorituskelpoiseksi. Yleiseen käyttöön tullakseen tämä vaatisi kuitenkin liian paljon resursseja. [Kim Tan ja Moreau, 2002]

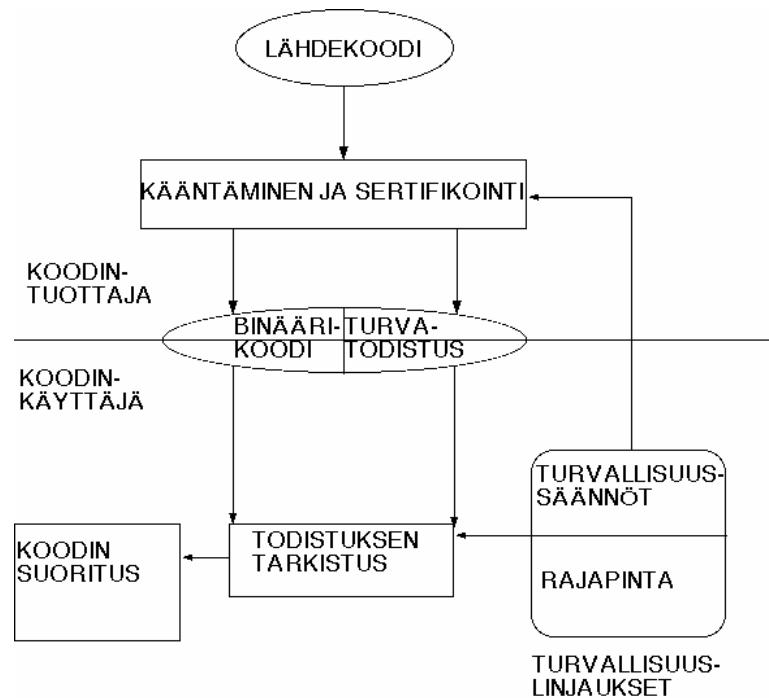
2.3. PPC

PPC eli Proof-carrying code [Necula, 1997] on mekanismi jonka avulla voidaan suorittaa jopa binäärimuotoista koodia turvallisesti. Turvallisuus perustuu koodin mukaan liitettyyn todistukseen, joka todistaa koodin noudattavan todistuksessa mainittuja turvallisuuslinjauksia. Käyttäjä voi nopeasti tarkistaa mitä koodin ajo vaatii ja joko hyväksyä tai hylätä koodin suorituksen.

PCC-arkkitehtuuri jakautuu sekä koodin tekijän että koodin käyttäjän puolelle. Koko prosessi keskittyy koodin käyttäjän ilmoittamaan turvallisuuslinjaukseen. Turvallisuuslinjauksen avulla koodin käyttäjä ilmaisee mitä palveluita vieraan koodin ajossa sallitaan. Turvallisuuslinjaus koostuu turvallisuussäännöistä ja rajapinnasta. Turvallisuussäännöt sisältävät vieraalle koodille sallitut palvelut joita kutsutaan määritellyn rajapinnan kautta. PCC-koodin käyttö voidaan jakaa kolmeen vaiheeseen: koodin kääntö ja sertifiointi, todistuksen tarkistaminen ja koodin suoritus. Kuva 1 selventää PCC-arkkitehtuuria.

Ohjelman lähdekoodi käännetään erityisellä PCC-kääntäjällä erityiseen PPC-binäärimuotoon, joka sisältää sekä kohdealustalle tarkoitetun binäärikoodin että todistuksen koodin turvallisuudesta. Käännöksen jälkeen binäärikoodiin tai turvallisuustodistukseen ei voida tehdä muutoksia. PCC-binäärimuotoinen koodi on valmis välitettäväksi koodin käyttäjälle.

Ladattuaan PCC-binäärikoodin koneelleen käyttäjä vertaa koodin turvallisuustodistusta omaan turvallisuuslinjaukseensa. Tarkistus on nopeaa ja helppoa. Jos todistus läpäisee tarkistuksen, koodi voidaan ladata suoritettavaksi. Tarkistus tarvitsee tehdä samalle koodille vain kerran. Koodin suoritus voidaan tehdä nyt ilman erityisiä ajonaikaisia tarkistuksia.



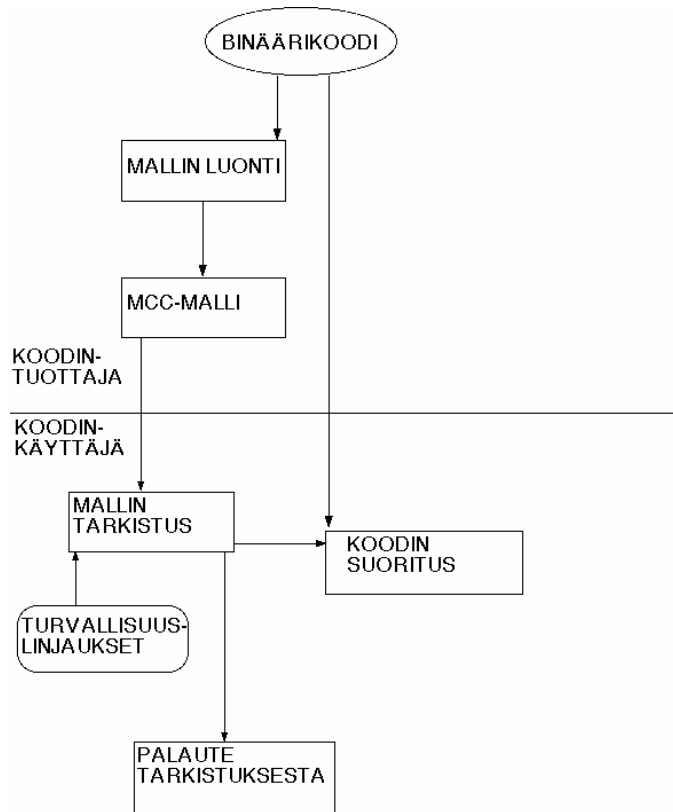
Kuva 1. PCC-arkkitehtuuri

PPC-koodin etu on tarkistuksen vaivattomuus. Koodintuottaja on tietoinen koodinkäyttäjän määrittelemistä turvallisuuslinjauksista, joten hän pystyy tekemään juuri tietylle käyttäjälle suunnatun version ohjelmastaan. Turvallisuudesta kertovan todistuksen luominen on kuitenkin melko raskasta eikä PPC-mekanismia ole testattu korkeammilla ohjelmointikielillä. PPC-mekanismiin haittapuolia ovat myös vain staattisen tarkistuksen käyttö ja tarkistuksen jättäminen koodin tuottajan tehtäväksi.

2.4. MCC

MCC eli Model-carrying code [Sekar *et al.*, 2003] on ohjelmistokehys jossa yhdistetään staattinen ja dynaaminen tarkistus. Turvallisuus perustuu koodin mukana tulevaan malliin joka toimii siltana binäärikoodin ja turvallisuuslinjausten välillä.

Mallissa on kuvattu kaikki koodin turvallisuuteen liittyvät asiat. Mallin avulla koodintuottajat pystyvät kertomaan koodin ajoon tarvittavat resurssit koodin käyttäjille. Tehdäkseen päätöksen koodin turvallisuudesta käyttäjän tarvitsee vain verrata mallia omaan turvallisuuslinjaukseensa. Kun ohjelma ajetaan, MCC-ohjelmistokehys varmistaa että ohjelma käyttäytyy mallinsa mukaisesti. Näin estetään koodintuottajaa antamasta virheellistä mallia haitalliselle koodille. MCC-arkkitehtuuri voidaan jakaa kolmeen osaan: mallin luominen, mallin tarkistaminen ja koodin suoritus. Kuva 2 kuvaa MCC-mekanismiin toimintaa.



Kuva 2. MCC-mekanismin toiminta

MCC-malliin on tarkoitus sisältää kaikki tiedot koodin turvallisuuteen liittyvistä asioista. Malli luodaan käyttäen erityistä turvallisuuskieltä käyttäen. Turvallisuuskieli perustuu äärelliseen tila-automaattiin, jonka avulla voidaan kuvata mm. tiedostojen käsittelyoikeudet, resurssien käyttörajoitukset ja verkkoyhteyksien rajoittaminen. Tila-automaatin avulla pystymme myös varmistamaan, että kirjoitus tapahtuu juuri siihen tiedostoon joka avattiin kirjoitusta varten.

MCC-malli luodaan ohjelman ajonaikaisen käyttäytymisen pohjalta. Komponentti joka vastaa mallin luomisesta koostuu kahdesta osasta. Toinen osista koostuu koodin ajoympäristöstä ja lokikirjasta, johon talletetaan koodin tekemät palvelupyynnöt. Toinen osa koostuu tila-automaatteja tuottavasta algoritmista ja lokikirjan lukijasta.

MMC-ohjelmistokehyksen verifiointikomponentti vertaa koodin mallia isäntäkoneen turvallisuuslinjauksesta tehtyyn MMC-malliin. Jos malli ei mene läpi käyttäjälle ilmoitetaan ne asiat, jotka rikkovat isäntäkoneen turvallisuuslinjauksia. Käyttäjä voi nyt valita ajaako hän koodin vai ei. Jos käyttäjä hyväksyy mallin tai malli menee tarkistuksesta läpi, koodi välitetään koodin ajoa valvovalle komponentille.

Koodin ajonaikainen käyttäytyminen pitää olla sen MMC-mallin mukainen. Käyttäytymisen tarkistus voitaisiin periaatteessa tehdä staattisesti binäärikoodia tutkimalla, mutta se olisi liian vaivalloista. Tämän vuoksi MCC käyttää ajonaikaista dynaamista tarkistusta.

MCC-arkkitehtuurin ajokomponentti pakottaa koodin toimimaan mallinsa mukaisesti. Jokainen käyttöjärjestelmälle tehty palvelupyyntö otetaan kiinni ja sitä verrataan koodin malliin. Jos palvelupyyntö eroaa koodin mallissa olevaan pyyntöön, ohjelman suoritus keskeytetään. Ohjelman keskeytys voi siis johtua joko siitä että koodintuottaja on tahallaan antanut väärän mallin koodilleen tai koodin malliin ei ole onnistuttu tallettamaan kaikkia koodin tekemiä toimintoja.

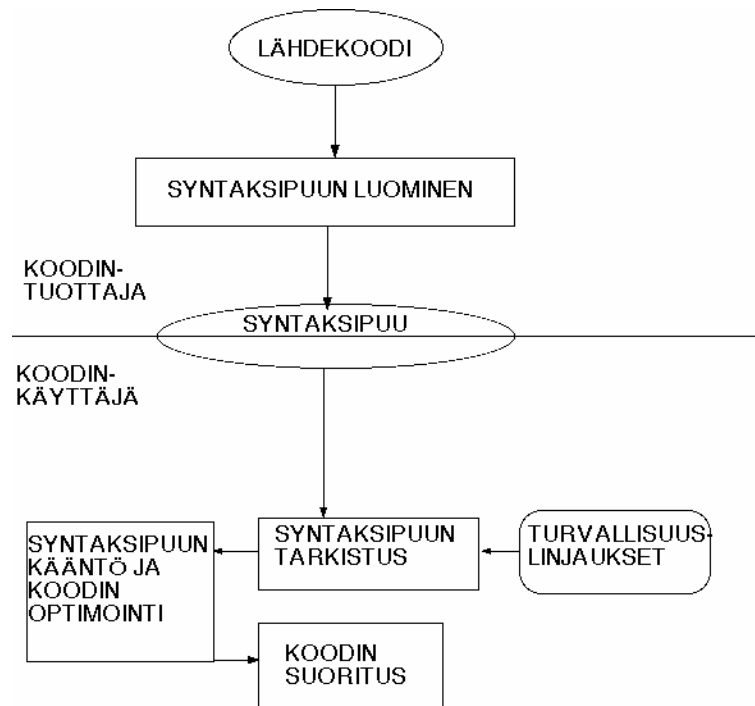
MCC mekanismin avulla koodintuottaja voi vapaasti tuottaa koodia miettimättä koodinkäyttäjän mahdollisia turvallisuusrajoituksia. Koodinkäyttäjän ei puolestaan tarvitse arvailla koodin tarvitsemia resursseja eikä paljastaa turvallisuuslinjauksiaan ulkopuolisille. MCC-mallin tarkistus ja koodin ajonaikainen tarkkailu voidaan täysin automatisoida. MCC-mallin tuottaminen on kuitenkin monimutkainen prosessi. Käytössä oleva ajonaikaiseen toimintaan perustuva mallin luonti ei välttämättä pysty taltioimaan kaikkia mahdollisia ohjelman toimintoja. Vaikkakin ajonaikaisia ohjelman keskeytyksiä saattaa tulla, ei turvallisuusasioita kuitenkaan laiminlyödä.

2.5. WELL

WELL eli Wellformedmed Encoding at the Language-Level [Haldar *et al.*, 2002] on malli, jonka perusideana on se, että vain lähdekoodista voidaan päätellä mitä koodi todella tekee. WELL-malli esittää tekniikan jonka avulla ohjelman lähdekoodissa oleva tieto sen toiminnasta välitetään käyttäjälle. Käyttäjä voi näin päättää rikkooko ladattu koodi hänen turvallisuuslinjauksiaan vai ei. WELL-arkkitehtuuri voidaan jakaa karkeasti kahteen osaan: syntaksipuun muodostamiseen ja syntaksipuun tarkistamiseen.

Lähdekoodista muodostetaan ns. syntaksipuu, joka on eräänlainen riisuttu versio ohjelman rakenteesta. Syntaksipuu muodostetaan määrätyin ehdoin, joten vain kunnollisesta koodista voidaan muodostaa syntaksipuu.

Koodinkäyttäjä lataa syntaksipuun, toteaa sen turvallisuuden ja kääntää sen ajettavaksi. Käännösvaiheessa koodia voidaan muuttaa soveltumaan paremmin kohdealustalle. Kuva 3 esittää WELL-tekniikan toiminnan.



Kuva 3. WELL-tekniikan toiminta

WELL-mallin etu on se, että lähdekoodin ja käyttäjälle välitettävän syntaksipuun ero on pieni. Käyttäjä voi näin varmentua koodin turvallisuudesta. Käyttäjä voi myös optimoida koodin toiminnan käänösvaiheessa. Käänökseen kuluu kuitenkin aikaa ja resursseja, mikä voi muodostaa ongelman joissakin tapauksissa.

2.6. Naccio

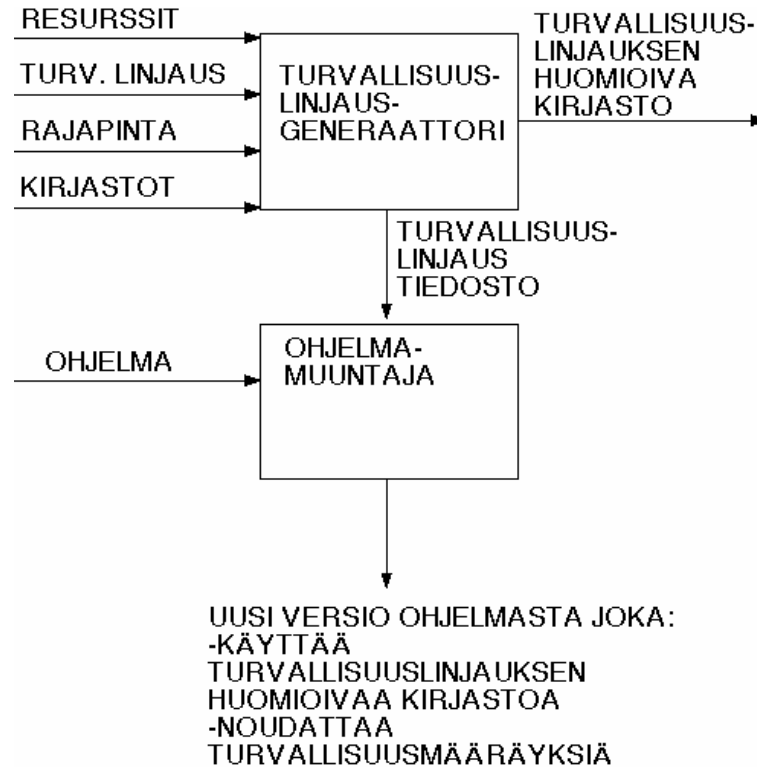
Naccio [Evans ja Tywman, 1999] on arkkitehtuuri joka yhdistää turvallisuuslinjauksen ja ohjelman tuottaen uuden ohjelman. Uusi ohjelma toimii kuten vanha ohjelma mutta noudattaa määriteltyä turvallisuuslinjausta. Naccio-arkkitehtuuri muodostuu kahdesta osasta: turvallisuuslinjausgeneraattorista ja ohjelmamuuntajasta.

Turvallisuuslinjausgeneraattorin luoman tiedoston pohjalta ohjelmamuuntaja muuttaa tuntemattoman ohjelman toiseksi ohjelmaksi, joka noudattaa määriteltyjä ehtoja. Turvallisuuslinjausgeneraattoriin syötetään kuvaus järjestelmän resursseista (mm. levyasemat, muistin määrä ja prosessointinopeus), ohjelmalle asetettavat rajoitteet (mm. sallittava muistin määrä ja prosessoriaika) ja järjestelmän rajapinnat (kuinka käyttöjärjestelmän palveluita kutsutaan) ja alustan luokkakirjastot (esim. Java API luokat tai Windowsin DLL:t).

Turvallisuuslinjausgeneraattori luo turvallisuuslinjauksen huomioivan luokkakirjaston ja turvallisuuslinjaustiedoston. Turvallisuuslinjauksen

huomioiva kirjastossa mahdolliset vahinkoa aiheuttavat metodit on kapseloitu jonkin toisen metodin sisään. Näin koodin tekemiä palvelupyynnöitä pystytään valvomaan. Turvallisuuslinjaustiedosto sisältää säännöt kuinka ohjelma on muutettava vastaamaan vallitsevaa turvallisuuslinjausta. Turvallisuuslinjaustiedostosta ilmenee mm. missä turvallisuuslinjauksen huomioiva kirjasto on ja mitkä ovat kapseloitujen metodien uudet nimet.

Ohjelmamuuntajan tehtävänä on lukea turvallisuuslinjaustiedosto ja muuttaa ohjelma sen mukaisesti. Ohjelmamuuntajan täytyy varmistaa että ohjelman binäärikoodi todella toimii turvallisuuslinjauksen mukaisesti. Ohjelmamuuntaja tärkein tehtävä on vaihtaa ohjelman käyttämät luokkakirjasto turvallisuuslinjauksen huomioivaan luokkakirjastoon. Ohjelmamuuntaja on alustariippuvainen. Kuva 4 esittää Naccio-arkkitehtuurin.



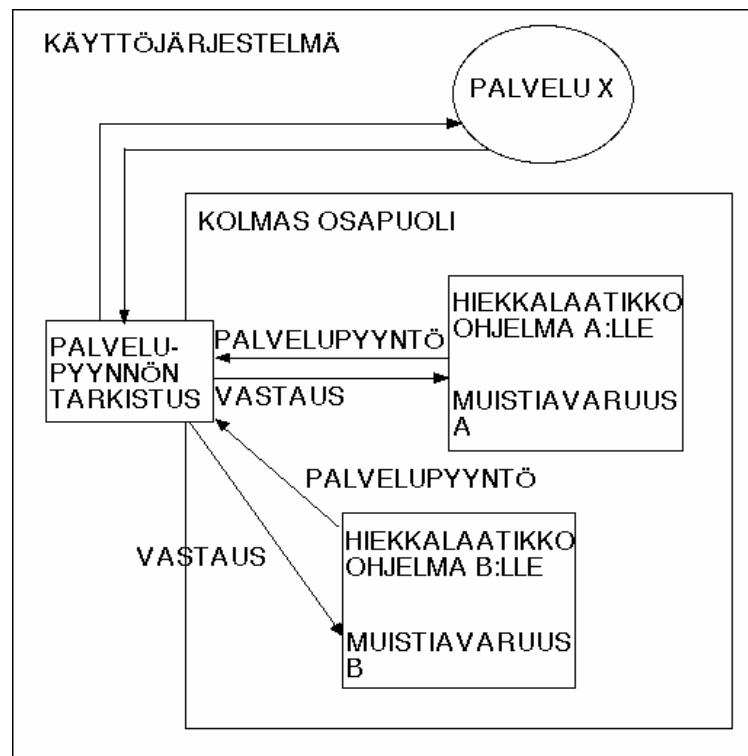
Kuva 4. Naccio-arkkitehtuuri

Naccio-arkkitehtuurin avulla voidaan varmistaa ohjelman turvallinen toiminta. Turvallisuuslinjausgeneraattorin luomaa tiedostoa voidaan käyttää useaan kertaan eri ohjelmille. Ohjelman muuntaminen syö kuitenkin resursseja, joten varsinkin dynaamisesti ladatun koodin toiminta voi hidastua. Myös valmiiksi turvallisten ohjelmien toimintaan voi tulla epätoivottuja rajoituksia.

2.7. Hiekkalaatikko

Hiekkalaatikkomallissa ladattu koodi ajetaan omassa "hiekkalaatikossa" jossa sen toimintaa pystytään valvomaan ja rajoittamaan. Näin ollen koodi ei pääse mielinmäärin mellastamaan isäntäkoneessa.

Ladattu koodi sijoitetaan omaan muistiavaruuteen eli se ei pääse käsiksi muiden ohjelmien tietoihin. Ajettava ohjelma joutuu myös pyytämään käyttöjärjestelmän palveluita jonkin kolmannen tahon kautta. Tämä kolmas taho sitten päättää isäntäkoneen turvallisuuslinjauksiin vedoten hyväksytäänkö palvelupyynnö vai ei. Kuva 5 kuvaa hiekkalaatikkomallin toiminnan.



Kuva 5. Hiekkalaatikkomallin toiminta

Hiekkalaatikkomalli on tehokas ratkaisu dynaamisesti ladatun koodin turvalliseen suoritukseen. Käyttäjä voi itse määrittellä minkälaisia oikeuksia koodille suodaan. Haittapuolena on se että tuntemattomalle koodille ei oikeastaan voida antaa minkäänlaisia oikeuksia. Tämä voi estää hyödyllistenkin ohjelmien kunnollisen toiminnan. [Prevelakis ja Spinellis, 2001]

3. Ohjelmointikielistä

3.1. yleistä

Java on suosionsa, ilmaisuvoimansa ja alustariippumattomuutensa ansiosta paras valinta dynaamista luokanlatausta tarvitsevaan ohjelmaan. Muut

ohjelmointikieliet eivät tarjoa niin mutkattomia keinoja dynaamiseen luokanlataamiseen.

3.2. Java

Java teknologia kehitettiin alun perin tuottamaan turvallinen ohjelmointiympäristö. Etenkin internetin yleistymisen lisäsi tarvetta ympäristöön jossa dynaamisesti ladattavia ohjelmia voitiin suorittaa turvallisesti. Javan turvallisuusmekanismit tarjosivat yrityksille mahdollisuuden tehdä vaivatta turvallisia ohjelmia, joita ihmiset uskalsivat käyttää.

3.2.1. Tyyppiturvallisuus ja tavukoodin varmennus

Java on tyyppiturvallinen kieli. Käännöksen aikana tarkistetaan, että muuttujiin talletettavat arvot täsmäävät muuttujien tietotyyppien kanssa ja metodit palauttavat mitä niiden kuuluukin. Ajon aikana tarkistetaan esimerkiksi taulukoiden rajat. Tyyppiturvallisuus estää viittaukset mm. vieraisiin muistialueisiin ja suojattuihin muuttujiin. Javan muistinhallinta on automaattista eli se antaa ohjelmalle tietyn määrän muistia ja putsaa järjestelmään jääneet roskat. Koodi ei voi varata loputtomasti muistia eikä jättää muistialueita vapauttamatta.

Tyyppiturvallisuuden lisäksi Java käyttää tavukoodin varmennusta takaamaan, että virtuaalikoneen suorittama koodi on laillista Java-tavukoodia. Tavukoodinvarmentaja ja virtuaalikone varmistavat tyyppiturvallisuuden ajon aikana. [Gong, 1998]

3.2.2. Ohjelmien lataus

Ohjelman lataamisen yhteydessä annetaan ohjelman luokille oma muistiavaruus. Tällä eristetään ohjelma muista samassa virtuaalikoneessa olevista ohjelmista. Java teknologiassa virtuaalikone toimii käyttöjärjestelmän ja ohjelman välissä. Kun ohjelma haluaa käyttöjärjestelmältä esimerkiksi I/O-palvelua, palvelu hoidetaan virtuaalikoneen kautta. Tämä on yksi Javan turvallisuusmekanismi peruspilareista. Ohjelman saatavilla olevia palveluita voidaan helposti rajoittaa virtuaalikoneen avulla.

Jokaiselle ohjelmalla voidaan asettaa oma turvallisuuslinjaus, joka pitää sisällään ohjelmalle sallitut palvelut. Turvallisuuslinjauksesta voi muokata mieleisensä, koska se on tekstimuotoinen tiedosto. Ohjelman turvallisuuslinjausta voidaan muuttaa myös ajon aikana. Tällöin käytetään erityistä oliota, joka pitää turvallisuuslinjauksen sisällään. Kun ohjelma pyytää jotain turvallisuusherkkää palvelua, erityinen turvallisuusvalvoja-luokka tarkistaa ohjelman turvallisuuslinjauksesta sallitaanko kyseinen palvelupyyntö.

Javan virtuaalikoneen toiminta ohjelman latauksen yhteydessä voidaan jakaa seuraaviin vaiheisiin:

1. Virtuaalikone vastaanottaa luokkatiedoston ja hyväksyy sen ajoon jos se läpäisee tavukoodinvarmentajan.
2. Virtuaalikone luo luokalle turvallisuuslinjauksen.
3. Virtuaalikone lataa luokan hiekkalaatikkoon.
4. Virtuaalikone suorittaa koodin ajonaikaisen tyyppiturvallisuuden huomioiden.
5. Jos ohjelma pyytää jotakin turvallisuusherkkää palvelua, niin turvallisuusvalvoja-luokka tarkistaa ohjelman turvallisuuslinjauksesta voidaanko pyyntö sallia.
6. Kun luokkatiedostoa ja siitä tehtyjä olioita ei enää käytetä Javan virtuaalikone putsaa ne pois järjestelmästä.

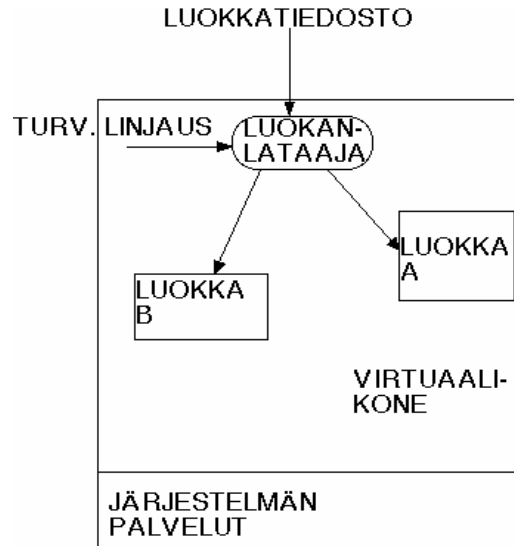
Kuva 6 esittää kahden eri luokan latauksen ja niiden sijoituksen Javan virtuaalikoneessa. [Lindholm ja Yellin, 1999], [Tozawa ja Hagiya, 2002]

3.2.3. Javan dynaaminen luokanlataus

Javassa dynaaminen luokanlataus voi tapahtua vasta kun luokkaa todella tarvitaan. Tyyppiturvallisuus taataan linkkauksen aikana tapahtuvien tarkistuksien avulla. Java sallii omien luokanlataajien tekemisen, joiden avulla luokkia voidaan ladata mistä tahansa.

Luokanlataajan pienin ladattavissa oleva yksikkö on luokka. Luokka on talletettu ns. luokkatiedostoon, joka ei nimestään huolimatta tarvitse olla tiedosto. Luokan tiedot voidaan lukea myös tietokannasta tai suoraan verkosta.

Ladattu luokka saa luokanlataajansa oikeudet eli sen ladanneen ohjelman oikeudet. Ohjelman käyttämään turvallisuuslinjaustiedostoon voidaan kuitenkin määrittää mitä oikeuksia eri paikoista ladatuille luokille suodaan. [Gong, 1998], [Liang ja Bracha, 1998]



Kuva 6. Kahden luokan lataus ja sijoitus virtuaalikoneessa

3.3. Muista kielistä

Muut ohjelmointikieliet eivät tarjoa niin mutkattomia keinoja dynaamiseen luokanlataamiseen. Seuraavaksi on esitelty kaksi mobiilikoodimaailmassa käytettyä kieltä, sekä ylivoimaisesti tehokkain ohjelmointikieli C++.

3.3.1. Safe-Tcl

Safe-Tcl on erityisesti selaimiin tarkoitettu ympäristö, jonka turvallisuus perustuu koodin tulkkaukseen ja eristykseen. Safe-Tcl käyttää ns. solua, jonka sisällä epäluotettava koodi suoritetaan. Solun sisältä ei ole suoraa yhteyttä vaaralliseksi pidettyihin palveluihin vaan palvelut välitetään ns. aliaksen kautta. Aliakset ovat hyvin määriteltäviä rajapintoja, jotka pitävät kirjaa siitä mitkä palvelut ovat sallittuja kullekin solun sisällä olevalle ohjelmalle. Safe-Tcl:n suurin ongelma on kielen rajoitteisuus. [Gritzalis ja Aggelis, 1998]

3.3.2. Telescript

Telescript kehitettiin agenttisysteemejä varten. Sen turvallisuus perustuu prosessien mukana kulkevaan listaan, johon on merkitty prosessin oikeudet. Oikeuksiin kuuluvat prosessille sallittava muistinmäärä, prosessille sallittava suoritusaika ja prosessin suoritusprioriteetti. [Moore, 1998]

3.3.3. C++

C++:n dynaamisesti ladatuilla kirjastoilla eri metodien toimintaa voidaan vaihdella, mutta turvallisuusasioita ei ole erikseen otettu huomioon.

4. TIA-arkkitehtuuri

4.1. yleistä

Esimerkki arkkitehtuurina käytetään Poika Isokosken [Isokoski ja Raisamo, 2003] kehittämään TIA-tekstinsyöttöarkkitehtuuria. TIA-tekstinsyöttöarkkitehtuurin idea on helpottaa tekstin syöttöä. Arkkitehtuuri mahdollistaa käyttäjän käyttä kerran opettelemaansa kirjoitusmetodia aina ja kaikkialla. Kirjoitusmetodi voi olla esimerkiksi softanäppäimistö tai käsialan tunnistin.

4.2. Arkkitehtuuri

TIA-arkkitehtuuri koostuu kahdesta pääkomponentista palvelimesta ja moduulista. Kun käyttäjä saapuu TIA-arkkitehtuuria tukevalle laitteelle, hänen henkilökohtaisen moduulin sijainti luetaan esimerkiksi käyttäjän sirukortilta. Moduulin kuuluvat luokat ja tiedostot on pakattu yhteen JAR-pakettiin. TIA-palvelin lataa moduulin käyttövalmiiksi ja mahdollistaa näin käyttäjän syöttää tekstiä laitteelle tutulla tavalla. TIA-arkkitehtuuria tukevalle laite on varusteltu kosketusnäytöllä. TIA-arkkitehtuurin runko on kirjoitettu Javalla, joten se on alustariippumaton. Myös käyttäjien moduulit tulee olla Javalla kirjoitettuja.

4.3. Ongelma

TIA-tekstinsyöttöarkkitehtuurin toiminta vaatii siis, että käyttäjä voi ladata verkosta oman henkilökohtaisen moduulin. Moduuli sisältää ohjeet käyttäjän tekstinsyöttötavan esittämisestä ja tulkkauksesta. Koska moduuli ladataan käyttäjän määrittelemästä paikasta, ei mikään estä käyttäjää kirjoittamasta järjestelmälle haitallista moduulia. Myös moduulien kykyä tarkkailla muiden käyttäjien moduuleita voidaan pitää tietoturva-aukkona.

4.4. Ratkaisusta

Nykyisessä ratkaisussa moduulilla on kaikki oikeudet mikä on selkeä riski. Aiemmin esitellyistä turvallisuusratkaisuista on luontevinta käyttää Java-kielen tarjoamia mahdollisuuksia.

Ideaali ratkaisu olisi luoda ohjelman sisään oma rajoitetumpi hiekkalaatikko, jossa ladatun JAR-paketin sisältämät luokat voitaisiin suorittaa. Tämä on kuitenkin helpommin sanottu kuin tehty. Java ei näet kuitenkaan suoraan tarjoa mahdollisuutta rajoittaa dynaamisesti ladatun JAR-paketin oikeuksia. Yksittäisesti ladattu luokka voidaan latausvaiheessa laittaa omaan turvallisuuspiiriinsä, jossa sen oikeuksia voidaan rajoittaa. Jos JAR-paketin pääluokka toteuttaisi luokan tiedot tavu-taulukossa palauttavan metodin, voisi

olla mahdollista saada kaikki JAR-paketissa olevat luokat saataisiin saman turvallisuuspiirin sisään.

Eräs ratkaisu olisi toteuttaa oma turvallisuusvalvoja-luokka, joka estäisi turvallisuusherät toiminnot. Turvallisuusvalvoja-luokka toimii kuitenkin koko ohjelman laajuisesti, joten TIA-palvelin jouduttaisiin käynnistämään aina uudestaan jokaisen käyttökerran jälkeen.

Helpoin keino estää ladattujen luokkien toimintoja on kirjoittaa itse oma turvallisuuslinjaustiedosto, jossa eri paikoista ladattujen luokkien oikeudet on esitelty. Kun TIA-palvelin käynnistetään otetaan turvallisuusvalvoja-luokka käyttöön ja kerrotaan mitä turvallisuuslinjaustiedostoa ohjelma käyttää. Ohjelman ajonaikana voidaan ottaa käyttöön myös dynaaminen turvallisuuslinjausolio, jonka kautta eri paikoista ladattujen luokkien oikeuksia voidaan säädellä.

5. Päätelmät

Dynaamisesti ladattavan koodin käyttö tulee hyvin todennäköisesti lähivuosina lisääntymään. Kuitenkin erilaisten turvallisuustekniikoiden yleistyminen ei ole yhtä todennäköistä, vaikka useat aiemmin esitellyt tekniikat vaikuttavat varsin päteviltä ratkaisuilta. Teollisuuden käyttämä Java tulee olemaan tulevinakin vuosina se ainut oikea ratkaisu dynaamisesti ladattua koodia käyttäviin ohjelmistoihin. Olisi toivottavaa että tulevaisuuden käyttöjärjestelmät ja ohjelmointiympäristöt ottaisivat dynaamisesti ladattavaan koodin turvallisuuteen paremmin huomioon. Pitäisi olla mahdollisuus luoda omia hiekkalaatikkomalliin perustuvia ajoympäristöjä pääohjelman sisään.

Olin hieman yllätynyt Javan sekavasta ja joustamattomasta turvallisuusarkkitehtuurista. Moni asia vaikutti selkeämmältä mitä se todellisuudessa oli. Javan olioparadigma näyttää hieman ampuneen itseään jalkaan turvallisuusseikoissa, sillä useaa yksinkertaista asiaa vaivasi "Kumpi tuli ennen? Muna vai kana?"-ongelma. Javan 1.5 versioon on onneksi luvassa parempi tuki ohjelman sisäisten hiekkalaatikkojen tekoon.

6. Lähteet

[Evans ja Tywman, 1999] David Evans ja Andrew Tywman, Flexible policy directed code safety. *IEEE Symposium on Security and Privacy*, 1999.

[Gong, 1998] Li Gong, Secure java class loading. *IEEE Internet Computing*, pages, marras- ja joulukuu, 1998. <http://citeseer.nj.nec.com/gong98secure.html>

[Gritzalis ja Aggelis, 1998] S. Gritzalis and G. Aggelis, Security Issues Surrounding Programming Languages for Mobile Code: Java vs Safe-Tcl. *ACM Operating Systems Review*, **32(2)**, Huhtikuu 1998, 16-32 .

- [Haldar *et al.*, 2002] Vivek Haldar, Christian H. Stork ja Michael Franz, The source is the proof. *Proceedings of the 2002 workshop on New security paradigms*, Virginia Beach, Virginia, 2002, 69-73.
- [Isokoski ja Raisamo, 2003] Poika Isokoski ja Roope Raisamo. Architecture for personal text entry methods. In Morten Borup Harning and Jean Vanderdonckt, editors, *Closing the Gaps: Software Engineering and Human-Computer Interaction*, 2003, 1-8.
- [Kim Tan ja Moreau, 2002] Hock Kim Tan ja Luc Moreau, Certificates for mobile code security. *Proceedings of the 2002 ACM symposium on Applied computing*, Madrid, Espanja, 2002, 76-81.
- [Liang ja Bracha, 1998] Sheng Liang ja Gilad Bracha, Dynamic class loading in the Java virtual machine. *Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*. Vancouver, British Columbia, Canada, 1998, 36-44.
- [Lindholm ja Yellin, 1999] Tim Lindholm ja Frank Yellin, *The Java(TM) Virtual Machine Specification (2nd Edition)*. Addison-Wesley Pub Co, 1999.
- [Moore, 1998] Jonathan T. Moore, Mobile Code Security Techniques. Technical Report, University of Pennsylvania, Department of Computer and Information Science, MS-CIS-98-28, Toukokuu 1998.
- [Necula, 1997] G. Necula, Proof-carrying code. *ACM Symposium on Principles of Programming Languages (POPL)*, 1997.
- [Prevelakis ja Spinellis, 2001] Vassilis Prevelaikis ja Diomidis Spinellis, Sandboxing Applications. *USENIX 2001 Technical Conference Proceedings: FreeNIX Track*, 2001.
- [Sekar *et al.*, 2003] R. Sekar, V.N. Venkatakrishnan, Samik Basu, Sandeep Bhatkar ja Daniel C. DuVarney, Model-carrying code: a practical approach for safe execution of untrusted applications. *Proceedings of the nineteenth ACM symposium on Operating systems principles*, 2003.
- [Tozawa ja Hagiya, 2002] Akihiko Tozawa ja Masami Hagiya, Formalization and Analysis of Class Loading in Java. *Higher-Order and Symbolic Computation*, **Vol. 15**, Issue 1, maaliskuu 2002, 7-55.

Esimaksettu GSM-puhelu ja sen vaikutus käyttäjiin

Jani Lundan

Tiivistelmä

Tässä tutkimuksessa perehdytään GSM-verkon rakenteen kautta GSM-puhelun kulkuun ja vertaillaan sitä esimaksetun GSM-puhelun kulkuun. Lisäksi tarkastellaan hyötyjä ja haittoja, joita esimaksettu tekniikka tuo käyttäjille.

Avainsanat -ja sanonnat: GSM-puhelu, esimaksettu puhelu, esimaksetun puhelun vaikutus käyttäjiin.

CR-luokka: C.2.1

1. Johdanto

Nokian GSM System Training -kirjassa sivulla 11 kerrotaan, että GSM-tekniikkaa alettiin kehittämään vuonna 1982, jolloin CEPT (The European Conference of Postal and Telecommunications Administrations) varasi 900MHz taajuuden GSM-tekniikka varten. Vuoden 1982 jälkeen CEPT jatkoi tutkimuksia ja suunnitelmien sekä aikataulujen tekemistä. Vuonna 1991 soitettiin varsinaisesti ensimmäinen GSM-puhelu. Ensimmäinen varsinainen GSM-verkko pystytettiin suomessa vuonna 1992, jonka jälkeen Suomessa oli nopeasti 13 operaattoria, jotka tarjosivat palveluja. Australia oli ensimmäinen Euroopan ulkopuolinen verkon pystyttävä. Tämä tapahtui myös vuonna 1992. Tämän jälkeen GSM-tekniikka laajeni ympäri maailmaa, ja vuoden 1997 lopulla GSM-tekniikan käyttäjiä oli maailmanlaajuisesti yli 100 miljoonaa.

Vuonna 2003 GSM-tekniikka on arkipäivää monille ihmisille ympäri maailman. Ainakin Suomessa GSM-tekniikka on monille jokapäiväinen asia. GSM-tekniikka on myös syrjäyttämässä lankapuhelimia monessa maassa. Harva kuitenkaan tietää, miten GSM-tekniikka toimii, ja miten puhelu kulkee GSM-verkossa. Tässä tutkimuksessa tutkitaan normaalin (postpaid), ja esimaksetun (prepaid) puhelun kulun eroja GSM-verkossa.

Jotta ymmärtäisi puhelun kulkua verkossa, on myös ymmärrettävä GSM-verkon rakenne. Kun GSM-verkon rakenne ja eri osa-alueet ymmärretään, voidaan GSM-puhelun kulku kuvata GSM-verkossa kohtuullisen helposti. Tässä tutkimuksessa puhelun kulkua tutkitaan ensin normaalin puhelun avulla, jonka jälkeen tutkimusta laajennetaan esimaksetun puhelun kulkuun.

Esimaksettu puhelu on osa älyverkkoja (Intelligent Network), ja kasvava osa GSM-tekniikkaa. Esimerkiksi Suomessa operaattorit laajentavat palvelujaan älyverkkoihin, joka mahdollistaa esimaksetut puhelut.

Vaikka tutkimuksen pääpaino on puhelun kulun kuvaamisessa ja verkon rakenteessa, perehtyy tutkimus myös esimaksetun puhelun vaikutuksiin loppukäyttäjien kannalta. Tutkimuksessa pyritään selvittämään mitkä asiat ovat käyttäjän kannalta huonommin, ja mitkä paremmin esimaksetussa puhelussa, kuin normaalissa puhelussa.

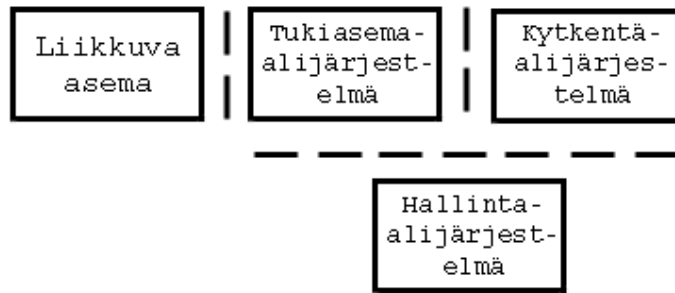
2. GSM-verkon rakenne

2.1. Yleistä rakenteesta

Kaj Granlund kertoo teoksen *Tietoliikenne* sivulla 299 seuraavaa GSM-verkon arkkitehtuurista. GSM-verkko koostuu neljästä eri kokonaisuudesta, jotka ovat yhteydessä toisiinsa. Loppukäyttäjille näkyvä osa on matkapuhelin, ja matkapuhelimen sisällä oleva SIM-kortti (Subscriber Identity Module). Tätä yhdistelmää kutsutaan liikkuvaksi asemaksi (MS, Mobile Station). Liikkuva asema on yhteydessä tukiasema-alijärjestelmään (BSS, Base Station Sub-system). Tukiasema-alijärjestelmä puolestaan on yhteydessä verkon kytkentäalijärjestelmään (NSS, Network Switching Sub-System). Hallinta-alijärjestelmä (OSS, Operating Sub-system) edustaa verkossa operaattorin osaa. Hallinta-alijärjestelmä antaa verkko-operaattorille mahdollisuuden valvoa ja hallita GSM-verkon toimintaa [Granlund, 2001, s. 120]. Hallinta-alijärjestelmän osaa verkosta ei käsitellä tässä tutkimuksessa tämän enempää.

On myös hyvä tietää, miten arkkitehtuurin eri osat ovat yhteydessä toisiinsa. Jokainen eri arkkitehtuurin osa yhdistyy toisiinsa rajapinnoilla. Rajapinnat sovittavat osat toisiinsa, ja jokaisella osalla on oma rajapintansa suhteessa toiseen. Tässä tutkimuksessa todetaan vain, että liikkuvan aseman ja tukiasema-alijärjestelmän välillä on ilmarajapinta ja muut osat arkkitehtuurissa on yhdistetty kiinteää tekniikkaa käyttäen.

Kuva 1 havainnollistaa GSM-verkon arkkitehtuuria. Kuvassa 1 olevat katkoviivat ovat erilaisia rajapintoja, joita tämä tutkimus ei kuitenkaan tarkemmin käsittele. Nokian GSM System Training kirjassa, sivulla 14, on kuvaa 1 vastaava kuva. Kuva 1 on kuitenkin mukailtu versio Nokian GSM System Training -kirjan kuvasta.



Kuva 1. GSM-verkon arkkitehtuuri. Katkoviivat edustavat erilaisia rajapintoja.

Kaj Granlundin *Langaton Tiedonsiirto* -kirjassa, sivulla 114 kerrotaan, että GSM-verkko rakentuu soluista. Yleensä jokaista solua GSM-verkossa vastaa yksi tukiasema, joka pitää yllä tietyn taajuista aluetta. Tärkeintä solujen ylläpitämässä taajuudessa on, etteivät solujen taajuudet sekoitu keskenään. Lähekkäisten solujen taajuudet tulee olla riittävän eritaajuisia. Taajuuksia voidaan kuitenkin käyttää uudestaan, kunhan samantaajuuksisten solujen välimatka on tarpeeksi suuri. Solujen kokoon vaikuttaa eniten tarvittavien kanavien määrä. Esimerkiksi kaupungeissa solujen koot ovat pieniä, koska käyttäjiä on luonnollisesti enemmän. Pienet solukoot puolestaan tuovat ongelmia mm. käyttäjän paikantamiseen. Tietyn alueen solut muodostavat ns. solujoukon. Tällaista solujoukkoa kutsutaan paikka-alueeksi (Location Area).

2.2. Liikkuva asema

Käyttäjille kaikkein näkyvin osa GSM-verkosta on liikkuva asema. Liikkuvana asemana voi toimia muukin laite kuin puhelin, mutta SIM-kortti on aina osa liikkuvaa asemaa. Tässä tutkimuksessa liikkuvalla asemalla tarkoitetaan puhelimen ja SIM-kortin yhdistelmää.

Liikkuva aseman toinen osa, SIM-kortti, on pieni älykortti, joka sisältää pienen tietokannan [Scourias, 1999]. Tietokanta sisältää mm. puhelinnumeron, joka koostuu maakoodista (Country Code), operaattorin tunnuksesta (National Destination Code) ja käyttäjänumerosta (Subscriber Number) [Nokia, 1998, s. 28]. SIM-kortti sisältää myös puhelimeen liittyviä asetuksia, ja muuta tietoa. Koska SIM-kortti on henkilökohtainen, on se suojattu käyttäjän salasanalla, eli PIN-koodilla (Personal Identity Number).

GSM-verkon kannalta on tärkeää tietää, missä tilassa liikkuva asema kulloinkin on. Tällaisia tiloja on kolme [Granlund, 1999, s. 299]: Suljetussa tilassa oleva liikkuva asema ei ole kytkettyneenä verkkoon, eli käytännössä puhelin on pois päältä. Odottavassa tilassa liikkuva asema seuraa verkon

toimintaa, mutta ei tee muuta. Aktiivisessa tilassa liikkuva asema varaa kaksisuuntaisen kanavan tiedonsiirtoa varten.

2.3. Tukiasema-alijärjestelmä

Nokia GSM System Training kirjan sivulla 41 kerrotaan, että tukiasema-alijärjestelmä koostuu kolmesta osasta. Tukiasemasta (BTS, Base Transceiver Station), tukiasemaohjaimesta (BSC, Base Station Controller), sekä nopeudensovittimesta (TC, TransCoder). Nopeudensovittimen tärkein tehtävä on sovittaa nopeudet kiinteän ja langattoman verkon välillä.

John Scourias esittää internetartikkelissaan, että tukiasema-alijärjestelmä koostuisi kahdesta osasta: tukiasemasta, ja tukiasemaohjaimesta. Vaikka nopeudensovitin voidaan asentaa minkä tahansa tukiasema-alijärjestelmän yhteyteen, on nopeudensovitin kuitenkin tärkeä osa tukiasema-alijärjestelmää sen tehtävän takia. Mielestäni sitä ei tulisi jättää pois tukiasema-alijärjestelmän kuvauksesta.

Tukiasema on suoraan yhteydessä liikkuvaan asemaan, sekä tukiasemaohjaimen [Honkanen, 2003; Kielenniva, 2003]. Tukiaseman tärkeimpiä tehtäviä ovat: digitaalisen tiedon modulointi radioteitä varten, tietoliikennekanavien tiedon koodaus ja purku, sekä puhelujen salaaminen ja purkaminen [Grandlund, 2001, s, 121]. Tukiasema vahvistaa myös GSM-verkon ns. kuuluvuusalueita. Kuuluvuusalue on se alue, jonka verkko kattaa, eli millä alueella liikkuva asema voi olla yhteydessä GSM-verkkoon. Tukiasemia voidaan liittää GSM-verkkoon erillisen radiolinkin avulla, tai kiinteällä yhteydellä. Langattomia tukiasemia suositetaan yleensä taajamissa ja kaupungeissa [Grandlund, 2001, s.121] niiden helpon asettelun takia.

Tukiasemaohjaimen tärkeimpiä tehtäviä on varata tiedonsiirtokanavia radioteiltä, ja vastata siirtotien käytöstä, sekä valvoa mm. signaalin laatua ja lähetystehoja [Grandlund, 2001, s. 121]. Lisäksi tukiasemaohjaimen tärkeimpiä tehtäviä on huolehtia liikkuvan aseman siirtymisestä tukiaseman solusta toiseen. Lisää liikkuvan aseman liikkumisesta GSM-verkossa, kappaleessa 3.3.

2.4. Verkon kytkentäalijärjestelmä

Sekä Nokian GSM System Training -, että Grandlundin Langaton Tiedonsiirto -kirja kertovat, että kytkentäalijärjestelmään on sijoitettu verkon päätehtävät. Tärkeimmät näistä tehtävistä ovat mm. puhelujen yhdistäminen GSM-verkon sisällä, sekä lanka- ja GSM-verkon välillä. Myös laskutuksen hoitaminen normaalissa puhelussa, ja käyttäjien paikallistaminen ovat kytkentäalijärjestelmän tärkeitä tehtäviä [Nokia, 1998, s. 37]. Verkon kytkentäalijärjestelmä sisältää matkapuhelinkeskuksen (MSC, Mobile

Switching Center), sekä tekstiviestikeskuksen (Short Message Service Center) [Granlund, 2001, s. 123].

Granlundin *Langaton tiedonsiirto* -kirjassa kerrotaan, että matkapuhelinkeskuksen tärkein tehtävä on puhelujen muodostaminen ja niiden ylläpitäminen. Lisäksi matkapuhelinkeskuksen tehtäviä ovat puhelujen kytkennät ja reititykset, sekä liikkuvan aseman seuranta, ja salaustekniikan välitys. Tekstiviestikeskuksen toimintaan tässä tutkimuksessa ei perehdytä.

Juha Lundan kertoi haastattelussaan, että kytkentäalijärjestelmä sisältää myös erilaisia tietokantoja. Näistä oleellimmat puhelun kulun kannalta ovat vierailijarekisteri (VLR, Visitor Location Register), joka pitää kirjaa niistä liikkuvista asemista, jotka ovat kulloinkin kyseisen vierailijarekisterin alueella. Kytkentäalijärjestelmä sisältää myös kotirekisterin (HLR, Home Location Register), joka pitää kirjaa käyttäjän pysyvistä tiedoista. Vierailijarekisteri ja kotirekisteri liittyvät olennaisesti matkapuhelinkeskukseen ja puhelun kulkuun GSM-verkossa. Kytkentäalijärjestelmä sisältää myös tunnistuskeskuksen (AuC, Authentication Center), joka on myös tietokanta [Granlund, 2001, s. 124]. Tunnistuskeskus säilyttää tietoturvallisia tietoja. Lisäksi kytkentäalijärjestelmä sisältää laitetunnistusrekisterin (EIR, Equipment Identity Register), jolla voidaan kieltää palvelut mm. sellaisilta laitteilta, jotka ovat varastettuja, tai jotka toimivat vastoin sopimuksia.

Juha Lundan kertoi haastattelussaan, että vierailijarekisteri on matkapuhelinkeskuksen yhteydessä, ja sen tiedot ovat väliaikaisia, toisin kuin kotirekisterin. Kun liikkuva asema tulee vierailijarekisterin toiminta-alueelle, kirjaa vierailijarekisteri liikkuvan aseman tiedot rekisteriinsä. Kun liikkuva asema poistuu vierailijarekisterin alueelta, poistetaan liikkuvan aseman tiedot vierailijarekisteristä. Yleensä tällaisessa tapauksessa liikkuva asema siirtyy jonkin toisen vierailijarekisterin alueelle, joka puolestaan kirjaa liikkuvan aseman omaan rekisteriinsä.

Haastattelussaan Juha Lundan kertoi, että kotirekisteri pitää kirjaa käyttäjän pysyvistä tiedoista, sekä siitä, minkä matkapuhelinkeskuksen alueella liikkuva asema kulloinkin on. Kotirekisteri huolehtii myös, että edellisessä kappaleessa kuvatus siirtymisen tiedot päivitetään molempiin vierailijarekistereihin.

3. Normaalin puhelun kulku GSM-verkossa

GSM-verkon rakenteen ollessa tiedossa, voidaan GSM-puhelun kulku havainnollistaa melko yksinkertaisesti. Seuraavissa kappaleissa kerrotaan miten normaali puhelu kulkee GSM-verkossa.

3.1. Puhelun muodostaminen GSM-verkossa

Kun liikkuva asema laitetaan päälle, kytkeytyy se verkkoon. Liikkuva asema ottaa tällöin yhteyden lähimpään tukiasemaan, joka lähettää tiedon matkapuhelinkeskuksen vierailijarekisteriin. Vierailijarekisteriin talletetaan käyttäjän sijainnin. Myös kotirekisteriin päivitetään tieto, missä solussa käyttäjä on.

Kun käyttäjä haluaa soittaa toiselle käyttäjälle, näppäilee hän sen käyttäjän numeron jolle on soittamassa. Verkon kannalta käyttäjä tekee palvelupyynnön, jotta saisi yhteyden toiseen käyttäjään [Nokia, 1998, s. 47]. Liikkuva asema ottaa yhteyden lähimpään tukiasemaan, joka toimittaa pyynnön matkapuhelinkeskukselle. Matkapuhelinkeskus tarkistaa, onko käyttäjällä oikeutta soittaa kyseiselle henkilölle [Nokia, 1998, s. 47]. Jos käyttäjällä on oikeus soittaa, tekee matkapuhelinkeskus tiedustelun kotirekisteriin käyttäjän sijainnista. Kotirekisteri palauttaa soittajaa palvelevalle matkapuhelinkeskukselle sen matkapuhelinkeskuksen sijainnin, joka palvelee käyttäjää jolle ollaan soittamassa [Nokia, 1998, s. 35].

3.2. Käyttäjän löytäminen

Soittajaa palvelevan matkapuhelinkeskuksen vierailijarekisteri tietää, minkä solun alueella käyttäjä on. Koska jokainen solu pitää yllä tietyntaajuisia kuuluvuusalueita, lähetetään solun kaikille tukiasemille hakupyynnö, joka sisältää liikkuvan aseman tunnistenumeron [Honkanen A, 2003]. Jokainen tukiasema lähettää tätä tunnistenumeroa omalla alueellaan. Kun liikkuva asema, joka tarkkailee verkkoa, huomaa oman tunnistenumeronsa, vastaa se verkon hakuun. Nyt käyttäjä on löydetty, ja puhelu voidaan yhdistää.

Käyttäjän hakemista GSM-verkosta voitaisiin havainnollistaa esimerkiksi. Otetaan esimerkiksi asiakkaan paikannus ravintolassa. Asiakas on varannut etukäteen pöydän ravintolasta, ja tullessaan sisään on käytännössä rekisteröitynyt ravintolaan. Ravintolan vierailijarekisteri pitää tietoa siitä, missä pöydässä henkilö istuu. Kun ravintolaan saapuu puhelu, jossa haetaan tiettyä henkilöä, tietää puhelimeen vastaava henkilö tavoiteltavan henkilön pöydän sijainnin vierailijarekisterin perusteella. Puhelimeen vastannut henkilö pyytää hovimestaria hakemaan kyseisen henkilön puhelimeen. Nyt kun hovimestari tietää pöydän sijainnin, kävelee hovimestari pöydän luo. Pöydässä istuu viisi ihmistä, joten hovimestari sanoo halutun henkilön nimen. Etsitty henkilö vastaa hovimestarin kyselyyn. Tämän jälkeen hovimestari johdattaa halutun henkilön puhelimelle.

Esimerkissä ravintolaan saapuminen vastaa rekisteröitymistä jonkin vierailijarekisterin alueelle. Hovimestari puolestaan vastaa tukiasemaa, joka

välittää hakupyynnön tietyn pöydän alueelle. Pöytä vastaa esimerkiksi GSM-verkon solua.

3.3. Käyttäjän liikkuminen GSM-verkon sisällä

Nokian GSM System Training -kirjassa sivuilla 53-58 kuvataan liikkuvan käyttäjän puhelun ylläpitoa. Liikkuvan käyttäjän puhelun ylläpidon mahdollistaa luovutus (handover) -operaatio. Luovutusoperaatioon voi olla kaksi syytä: tukiaseman signaalin vahvuus voi pudota liian alhaiseksi, tai solun käyttäjämäärä lähestyy maksimia. Luovutusoperaatio voidaan jaotella neljään eri luovutustyyppiin:

- Solun sisäinen (Intra Cell),
- Solujen välinen (Inter Cell),
- Solujen välinen, mutta eri tukiasemaohjain (Inter Cell-Inter BSC), sekä
- Eri matkapuhelinkeskusten välinen (Inter MSC)

Solun sisäinen luovutus on pienin luovutustyypeistä, ja siinä käyttäjän puhelu luovutetaan toiselle kanavalle, samassa solussa. Päätöksen luovutuksen tarpeesta tekee tukiasemaohjain, joka myös toteuttaa luovutuksen.

Solujen välisessä luovutuksessa käyttäjä liikkuu tukiaseman alueelta toisen tukiaseman alueelle. Tällaisessa tapauksessa muodostetaan yhteys uuden tukiaseman, ja liikkuvan aseman välille. Uuden yhteyden luomisen jälkeen katkaistaan yhteys vanhan tukiaseman ja liikkuvan aseman väliltä. Tässäkin tapauksessa tukiasemaohjain tekee päätöksen ja toteuttaa luovutuksen.

Solujen välisessä, mutta eri tukiasemaohjaimisessa luovutuksessa luovutuksen hoitaa matkapuhelinkeskus, vaikka päätöksen tekee tukiasemaohjain. Luovutus tapahtuu samalla tavalla, kuin solujen välisessä luovutuksessa.

Eri matkapuhelinkeskusten välinen luovutus on monimutkaisin. Käyttäjää palveleva matkapuhelinkeskus (ankkurimatkapuhelinkeskus), ottaa yhteyden toiseen matkapuhelinkeskukseen. Kohteena olevaan matkapuhelinkeskuksen vierailijarekisteriin päivitetään sijaintitiedot, ja yhteyden ylläpitäminen siirretään kohdematkapuhelinkeskukselle. On huomioitava, että matkapuhelinkeskuksset ovat käytännössä kaksi eri puhelinvaihdetta. Tämän takia puhelu voidaan siirtää matkapuhelinkeskusten välillä vain, jos on olemassa numero, joka sisältää kohdematkapuhelinkeskuksen tunnusteen. Tällaisia tapauksia varten on olemassa luovutusnumero (HON, HandOver Number). Luovutusnumero sisältää reititystietoa, jonka avulla puhelu voidaan ohjata toisen matkapuhelinkeskuksen alaisuuteen.

4. Esimaksettu puhelu ja kulku GSM-verkossa

Esimaksetun puhelun kulku eroaa hieman normaalin puhelun kulusta. Puheluilla on kuitenkin paljon yhteistä, ja voidaankin sanoa, että esimaksettu puhelu on laajennus GSM-verkoissa. Tarkkaan ottaen esimaksetut puhelut ovat osa älyverkkoja, joita tässä tutkimuksessa ei kuitenkaan tarkemmin tutkita. Jotta ymmärrettäisiin esimaksetun puhelun kulku, täytyy tietää mitä esimaksettu puhelu tarkoittaa.

4.1. Esimaksettu puhelu

Nokian MSG/IN prepaid concept -kirjassa kerrotaan, että esimaksetulla puhelulla tarkoitetaan puhelua, jossa käyttäjä maksaa käyttämistään palveluista etukäteen. Tämä eroaa normaalista puhelusta siten, että normaalissa puhelussa tietyn ajanjakson puhelut lasketaan yhteen. Yhteenlaskettu ajanjakso muutetaan rahaksi, ja tästä summasta lähetetään lasku käyttäjälle.

Nokian MSG/IN prepaid concept -kirjassa kerrotaan myös, että esimaksetussa puhelussa laskutus tapahtuu samaan aikaan, kun puhelu on käynnissä. Jokaisella käyttäjällä, tai tilaajalla, on käytössään tili (Account), jonne käyttäjä tallettaa rahaa palveluita varten. Tältä tililtä vähennetään rahaa sitä mukaa, kun käyttäjä käyttää palveluita. Jokaisella tilillä on myös tiliraja (Account Limit), joka kertoo kuinka kauan käyttäjä voi soittaa, tai kuinka paljon rahaa käyttäjällä on.

Nokian MSG/IN prepaid concept -kirja selvittää myös yhtä olennaista osaa esimaksetuista puheluista, rahan loppumista. Rahan loppuessa tililtä puhelun aikana, voidaan puheluille tehdä kolme erilaista toimintoa:

- Puhelu voidaan estää (Barred),
- puhelu voidaan ohjata ennalta määrättyyn numeroon,
- tai puhelua voidaan jatkaa normaalisti.

Saldon loppumista seuraavan toiminnon määrää palvelu, jota käyttäjä on parhaillaan käyttänyt. Käyttäjälle voidaan tehdä monta erilaista palvelua. Jokainen käyttäjälle tehty palvelu on erilainen, mutta eroksi riittää jo esimerkiksi pelkästään eri rahamäärä, joka käyttäjältä peritään käyttäjän soittaessa.

Esimaksettuun puheluun kuuluu myös olennaisesti Nokian MSG/IN prepaid concept -kirjassa kuvatut tiedotteet (Notifications). Tiedotteella tarkoitetaan useimmiten ilmoitusta, jonka käyttäjä saa tilin saldon lähestyessä nollaa. Tiedote voi olla myös esimerkiksi käyttäjän tilin lataushistoria. Tiedote voi olla joko äänitiedote tai tekstiviestitiedote.

4.2. Esimaksetun puhelun kulku GSM-verkossa

Esimaksettu puhelu on kulultaan hyvin saman tyyppinen kuin normaali puhelukin. Näillä kahdella tyyppillä on kuitenkin eroja. Aivan kuten normaalissa puhelussa, käyttäjä näppäilee numeron, liikkuva asema ottaa yhteyden tukiasemaan, ja tukiasema välittää pyynnön matkapuhelinkeskukselle. Normaalissa puhelussa palveleva matkapuhelinkeskus yhdistäisi puhelun toisen matkapuhelinkeskuksen kanssa, perustuen kotirekisterin tietoihin. Esimaksettu puhelu käyttäytyy kuitenkin tässä tapauksessa eri lailla.

Nokian MSG/IN 3 steps to IN: First Step -kirjassa kerrotaan, että matkapuhelinkeskuksessa on osa, joka hoitaa esimaksettujen puheluiden reitityksen. Tällaista osaa kutsumme tässä tutkimuksessa palveluvaihteeksi (Service Switching Point). Kun puhelu tulee palveluvaihteeseen, saa palveluvaihde tietoonsa mm. soittajan numeron, ja sen henkilön numeron, jolle ollaan soittamassa. Palveluvaihde tarkastaa matkapuhelinkeskuksen yhteydessä olevasta vierailijarekisteristä, että soittaja on älyverkkopalveluiden tilaaja.

Nokian MSG/IN 3 steps to IN: First Step -kirjassa kerrotaan, että palveluvaihde yhdistää puhelun palvelunhallintapisteeseen (SCP, Service Control Point). Palvelunhallintapiste tunnistaa käyttäjän, ja suorittaa käyttäjän palvelut, jos niitä käyttäjällä on. Kun palvelut on suoritettu, lähettää palvelunhallintapiste palveluvaihteelle ohjeet puhelun jatkamisesta [Nokia B, 2000]. Ohje voi olla esimerkiksi käsky jatkaa puhelua, estää puhelu tai ohjata puhelu toiselle henkilölle [Honkanen B, 2003]. Palveluvaihde voi nyt jatkaa puhelua näiden ohjeiden perusteella.

Nokian MSG/IN 3 steps to IN: First Step -kirjassa kerrotaan, että kun palvelut on suoritettu, ja ohje jatkamisesta on lähetetty palveluvaihteelle, tarkistaa palveluvaihde kotirekisteristä, minkä matkapuhelinkeskuksen alueella henkilö jolle ollaan soittamassa on. Kun henkilö on löydetty, yhdistetään puhelu sille matkapuhelinkeskukselle, jonka alueella henkilö on.

Mikäli henkilö, jolle ollaan soittamassa ei ole älyverkkopalveluiden tilaaja, yhdistää matkapuhelinkeskus puhelun samalla tavalla, kuin normaalin puhelun kulussakin. On kuitenkin mahdollista, että henkilö jolle ollaan soittamassa on älyverkkopalveluiden tilaaja. Tällaisissa tapauksissa matkapuhelinkeskuksen palveluvaihde tunnistaa käyttäjän älyverkkopalveluiden käyttäjäksi, ja ohjaa puhelun palvelunhallintapisteelle. Aivan kuten soittajankin kohdalla, tarkastetaan soitettavan henkilön palvelut ja suoritetaan ne, mikäli käyttäjällä on aktiivisia palveluja. Tämän jälkeen palvelunhallintapiste lähettää ohjeet puhelun jatkamisesta palveluvaihteelle, ja

palveluvaihte jatkaa puhelua näiden ohjeiden perusteella. Jos palvelut eivät estä puhelua, yhdistyy puhelu soitettavalle henkilölle.

5. Esimaksetun tekniikan vaikutus käyttäjiin

Esimaksettu puhelutekniikka on ollut käytössä monissa maissa, ja mm. Taiwanissa esimaksetut puhelut ovat saaneet suuren suosion[Yi-Bing, 1999]. Yi-Bingin tutkimuksesta selviää, että yli 40% Taiwanin 1,2 miljoonasta matkapuhelinkäyttäjistä on esimaksetun tekniikan käyttäjiä. Mikä esimaksetussa puhelussa viehättää? Miksi monet vaihtavat esimaksettuihin puheluihin? Mitä hyötyä esimaksetusta tekniikasta on? Entä mitä haittaa?

Nokian MSG/IN 3 steps to IN: First step -kirjassa kerrotaan esimaksettujen puheluiden suurimmista eduista. Kirjassa kerrotaan, että käyttäjän ei tarvitse sitoutua minkään operaattorin asiakkaaksi pitkäksi aikaa. Halutessaan käyttäjä voi ostaa liittymän, käyttää sen loppuun ja siirtyä seuraavan operaattorin asiakkaaksi. Myös Yi-Bing [Yi-Bing, 1999] pitää tätä syytä ehkä tärkeimpänä syynä esimaksettujen puheluiden suosioon. Myös liittymien helppo saatavuus on esimaksettujen puheluiden etu. Käyttäjä voi ostaa liittymänsä vaikka lähimmästä supermarketista, samalla kun on tekemässä ruokaostoksia. Yi-Bing [Yi-Bing, 1999] kertoo tutkimuksessaan myös, että monet nuoret suosivat esimaksettuja puheluita juuri sen helpon saatavuuden takia. Nuoret eivät halua täyttää kaavakkeita, joihin vaaditaan täysi-ikäisen allekirjoitus. Tämän takia monet nuoret päätyvät ostamaan esimaksetun liittymän.

Sekä Yi-Bingin [Yi-Bing, 1999] tutkimus, että Nokian MSG/IN 3 steps to IN: first step -kirja kertovat, että esimaksettujen puheluiden suurimpia etuja ovat myös puhelumenojen kontrollointi. Käyttäjän ostaessa etukäteen tietyn summan tililleen rahaa, ei voi puhua yli varojensa, kuten normaalissa GSM-puhelussa. Tätä seikkaa tukee myös säännölliset saldoilmoitukset. Jokaisen puhelun alussa tai lopussa tuleva saldoilmoitus pitää käyttäjän tietoisena siitä, kuinka paljon rahaa hänen tilillään vielä on.

Kun käyttäjältä on loppumassa raha, voi hän ladata rahaa tililleen ostamalla latauslipukkeen (voucher). Latauslipuke sisältää koodin, joka lähetetään matkapuhelinkeskukseen, joko soittamalla tai tekstiviestillä. Tämän jälkeen käyttäjän tilillä on latauslipukkeen arvoa vastaava summa rahaa. Vaikka tässä tekniikassa on paljon hyvää, on mielestäni latauslipukkeissa parantamisen varaa. Suurin ongelma latauslipukkeissa on, että ne ovat tietyn summan suuruisia. Jos esimerkiksi suurin latauslipun koko on 50 euroa, joutuu käyttäjä ostamaan neljä tällaista lipuketta saadakseen tililleen 200 euroa. Tällöin käyttäjä joutuu myös lähettämään jokaisen latauslipukkeen koodin

matkapuhelinkeskukseen yksitellen. Tämä ei välttämättä ole mielekästä esimaksettujen puheluiden suurkuluttajille.

Vaikka latauslipukkeet ovat nopea ja hyvä keino ladata rahaa tililleen, pitäisi tarjota mahdollisuus ladata enemmän rahaa kerralla. Tällaisen palvelun voisi tehdä esimerkiksi internettiin, jossa käyttäjä voisi siirtää suoraan omalta tililtään rahaa puhelutililleen, aivan kuten maksaisi normaalin puhelun puhelinlaskua. Toinen mahdollisuus olisi, että ostaessa latauslipuketta, voisi päättää sen ladattavan summan, jonka haluaa puhelutililleen. Myyntipaikoissa voisi olla jonkinlainen laite, johon myyjän syöttäessä tietyn rahasumman, tulostaisi kone koodin, jonka asiakas voisi ladata normaalin lipukkeen tavoin tililleen. Ainakaan Suomessa tällaiset keinot eivät ole vielä käytössä.

Esimaksetun tekniikan etuihin voidaan laskea myös helposti muokattavat palvelut [Nokia A, 2000]. Nokian MSG/IN three steps to IN: first step kirjassa on hyvä esimerkki muokatuista palveluista: Perheen lapsi haluaa kännykän. Vanhemmat ei kuitenkaan uskalla ostaa normaalia liittymää suurten puhelinlaskujen pelossa. Vanhemmat voivat hankkia esimaksetun liittymän, johon ladataan kuukausittain rahaa tietty summa. Kun lapsi käyttää puhelinta, vähenee raha tililtä kunnes tili on tyhjä. Monet vanhemmat kuitenkin haluavat, että lapsi voi soittaa hätätilanteessa kotiin. Lapsen esimaksettuun liittymään voidaan tehdä palvelu, jossa saldon loppumisen jälkeen vanhemmille soitettu puhelut laskutetaan vanhempien liittymästä.

Ei voida kuitenkaan sanoa, että esimaksetuista puheluista olisi pelkästään hyötyä. Mielestäni suurin haitta on, että rahan loppuessa ei voi enää soittaa muihin numeroihin kuin hätänumeroon ja rahanlatausnumeroon. Käyttäjän on siis huolehdittava, että rahaa on tilillä jos haluaa soittaa. Rahan lataaminen ei myöskään ole ainoa ongelma. Kun tililleen lataa rahaa, vanhenee se tietyn ajan kuluttua. Yleensä tällainen aika on puoli vuotta [Nokia C, 2000], vaikka operaattori voikin määritellä vanhenemisajan. Tämä on suuri ongelma käyttäjille, jotka puhuvat vähän. Toisaalta tämä on myös ongelma henkilöille, jotka lataavat tililleen suuria summia kerralla. Tosin suurien summien lataaminen on vaikeaa vakiokokoisten latauslipukkeiden takia. Ehkä suurin syy suurien summien lataamisen ongelmalle onkin juuri rahavarojen vanheneminen.

Mielestäni monista ongelmista päästäisiin, jos voitaisiin yhdistää sekä esimaksettu liittymä, ja normaali GSM-liittymä. Liittymää tulisi ensisijaisesti käyttää esimaksettuna liittymänä, mutta rahan loppuessa esimerkiksi paikassa, jossa lataus ei olisi mahdollista, voisi käyttää normaalia osaa liittymästä.

Tällaisista liittymistä puhuttaessa voitaisiin kysyä, mitä hyötyä ylipäänsä olisi yhdistää esimaksettu ja normaali liittymä? Mielestäni edellisessä

kappaleessa mainitsemani syyn lisäksi, ehkä tärkein syy olisi puhelujen hinnat. Monissa maissa esimaksetut puhelut ovat halvempia kuin normaalit. Yhdistämällä kaksi erityyppistä liittymää voitaisiin saada esimaksettujen puheluiden halvat hinnat, mutta samalla monet esimaksettujen puheluiden ongelmat poistuisivat.

6. Lopuksi

Tässä tutkimuksessa on kuvattu GSM-puhelun ja esimaksetun GSM-puhelun kulkua GSM-verkossa. Puhelun kulun ymmärtämiseen kuuluu olennaisesti GSM-verkon rakenteen ymmärtäminen, joka on tässä tutkimuksessa kuvattu varsin karkealla tasolla. Edellä mainittujen lisäksi on tutkittu esimaksetun tekniikan vaikutusta käyttäjiin. Vaikutuksista on pyritty tuomaan esille sekä hyviä, että huonoja puolia.

GSM-verkon rakenne on varsin oleellinen asia selitettäessä GSM-puheluiden kulkua. Vaikka rakenne kattaa isohkon osan tutkimuksesta, katson rakenteen todella suureksi osaksi kokonaiskuvan hahmottamista. Rakenteen kuvaus on mielestäni varsin toimiva ja hyvä kokonaisuus. Rakenteen kuvausta voisi kuitenkin parantaa esimerkiksi rajapintojen kuvauksella. Rajapintoja on kuitenkin GSM-verkossa paljon, joten tämän kokoisessa tutkimuksessa niiden käsitteleminen olisi todella hankalaa.

Myös GSM-puhelun kulku on kuvattu varsin karkealla tasolla. Jos tutkimusta laajentaisi, voisi tämän osan tehdä tarkemmin, ja tarkastella eri osien välistä kommunikointia lähemmin. Laajennettaessa eri osien väliseen kommunikointiin, laajenisi tutkimus väkisin myös edellisessä kappaleessa mainittuihin rajapintoihin. Eri osien kuvaaminen ilman rajapintojen kuvaamista olisi hankalaa, ja jopa epäloogista.

Esimaksetun puhelun kulun kuvauksessa olisi mielestäni enemmän tarkentamisen varaa. Esimaksettujen puheluiden ollessa vain yksi osa älyverkoista, pitäisi esimaksettujen puheluiden pohjustaa älyverkoista kertomalla. Älyverkot aihepiirinä voisi jopa antaa mahdollisuuden pro gradu -tutkielmalle sen ollessa varsin laaja alue.

Tutkimuksessa käytiin läpi myös esimaksettujen puheluiden hyviä ja huonoja puolia. Tämä tuntuu olevan hieman vähemmän tutkittu alue, ja saattaisi olla hyvä tutkimuskohde ajatellen esimerkiksi pro gradu -tutkielmaa. Kaikenkaikkiaan tässä tutkimuksessa tuli esille esimaksettujen puheluiden tärkeimmät hyvät sekä huonot puolet, mutta tarkennettavaa varmasti olisi ollut.

Viiteluettelo

- [Granlund, 2001] Kaj Granlund, *Langaton tiedonsiirto*. Docendo, 2001.
- [Granlund, 1999] Kaj Granlund, *Tietoliikenne*. TecnoLit, 1999.
- [Honkanen A, 2003] Simo Honkanen, keskustelu, 31.10.2003
- [Honkanen B, 2003] Simo Honkanen, sähköpostikeskustelu 11.12.2003
- [Kielenniva, 2003] Jani Kielenniva, keskustelu, 31.10.2003
- [Lundan, 2003] Juha Lundan, keskustelu, 28.10.2003
- [Nokia A, 2000] Nokia, *MSG/IN 3 Steps to IN:First Step*. Nokia, 2000.
- [Nokia B, 2000] Nokia, *MSG/IN 3 Steps to IN:Second Step*. Nokia, 2000.
- [Nokia C, 2000] Nokia, *MSG/IN Prepaid Concept*. Nokia, 2000
- [Nokia, 1999] Nokia, *IN Training for Newcomers*. Nokia Telecommunications Oy, 1999.
- [Nokia, 1998] Nokia, *SYSTRA GSM System Training*. Nokia Telecommunications Oy, 1998.
- [Scourias, 1999] John Scourias, Overview of the Global Systems for Mobile Communications, 1999, <http://www.privateline.com/PCS/GSM0.html>, Tarkistettu 23.11.2003
- [Yi-Bing , 1999] Yi-bing Lin, Herman C.R. Rao and Ming Feng Chang, Mobile Prepaid Phone Services, *INFOCOM(3)*, 1999.

Kansainvälisen WWW-sivuston suunnitteluhaasteet ja käytettävyydestaus

Laura Mustonen

Tiivistelmä.

World Wide Web on nimensä mukaisesti maailmanlaajuinen palvelu. Erilaisista kulttuureista tulevat käyttäjät asettavat kansainvälisille sivustoille uudenlaisia käytettävyyksvaatimuksia. Tämä tutkielma pyrkii valottamaan kansainvälisen käytettävyyden käsitettä ja esittelemään kansainvälisten WWW-sivustojen käytettävyydestauksen menetelmiä ja haasteita, sekä pohtimaan ongelmia, joita kohdataan kulttuurien välisessä kanssakäymisessä käytettävyydestauksen yhteydessä. Kirjallisuuskartoituksen avulla saatiin selville, että kansainvälisen käytettävyydestauksen suorittamiseksi on tarjolla erilaisia, sekä pienin resurssein suoritettavia edullisiakin vaihtoehtoja että laajamittaisia menetelmiä. Lisäksi havaittiin, että kansainvälisiä käytettävyydestejä tekevien on hyvä olla selvillä testaukseen liittyvistä erityiskysymyksistä.

Avainsanat ja -sanonnat: kansainväliset käyttöliittymät, kansainvälinen käytettävyydestaus, internationalisointi, lokalisointi, World Wide Web.

CR-luokat: H.5.2

1. Johdanto

Kansainvälisyys ja globalisaatio ovat ajankohtaisia asioita tällä hetkellä, ja niiden merkitys korostuu erityisesti nykyisten viestintäteknologioiden parissa. World Wide Web (jäljempänä WWW) on nimensä mukaan maailmanlaajuinen palvelu. Yritykset tavoittelevat verkon kautta yhä laajempia markkinoita, mutta kansainvälisille markkinoille tarkoitetun palvelun menestyminen vaatii erityisen hyvää suunnittelua ja erilaisten käyttäjäryhmien tuntemista ja huomioonottamista.

Hyvä käytettävyys edesauttaa palvelun menestymistä. Käyttäjän alkuperän, kulttuurin ja kielen huomiointi on osa taitavaa käyttäjälähtöistä suunnittelua, ja onnistuessaan johtaa tunteeseen hyvästä käytettävyydestä. Käyttäjän on tärkeä tuntea, että hänen hankkimansa tuote tai käyttämänsä palvelu on ikään kuin häntä varten suunniteltu.

Sekä kansainvälistyminen että WWW ovat ilmiöinä suhteellisen uusia. Vakiintuneita käytäntöjä ei vielä ole. Siksi käytettävyyden saralla tehtävää on paljon, ja tutkimusta tarvitaan. WWW:n käyttäjäkunta on suuri ja

heterogeeninen, mutta WWW-suunnittelijat eivät aina osaa ottaa sitä työssään tarpeeksi kattavasti huomioon. Verkossa voi helposti törmätä kansainvälisiin sivustoihin, joiden käytettävyys on huono juuri kansainvälisyyden asettamien vaatimusten laiminlyönnin seurauksena.

Käytettävyystestauksen avulla voidaan löytää parannuskeinoja. Kun halutaan saada tietoa nimenomaan kulttuurin vaikutuksesta käytettävyyteen, täytyy testauksen mukautua sen asettamiin vaatimuksiin. On tärkeää löytää oikeanlaiset työskentelytavat, jotta tutkimusasetelma tukisi niitä kysymyksiä, joihin vastauksia haetaan.

Käytettävyystestaus unohdetaan usein sillä perusteella, että se aiheuttaa lisäkustannuksia jo muutenkin kalliisiin projekteihin. Testausta voidaan kuitenkin tehdä pienin kustannuksin, eikä kansainvälinen testauskaan välttämättä aiheuta suuria lisäkuluja, sillä sitä voidaan tehdä myös pienin resurssein ja jopa ilman matkustamista.

Tämä tutkielma johdattaa lukijansa kansainväliseen käytettävyyteen ja käytettävyystestauksen menetelmiin nimenomaan WWW-suunnittelun näkökulmasta. Luvussa 2 pohditaan kansainvälisyyden asettamia käytettävyysvaatimuksia ja WWW-sivuston kansainvälistämisprosessia. Luku 3 käsittelee kulttuurien välisen viestinnän erityispiirteitä lähinnä Geert Hofsteden tutkimusten valossa. Luku 4 esittelee kansainvälisen käytettävyystestauksen menetelmiä ja pohtii niiden soveltuvuutta erilaisiin projekteihin. Koska kansainvälinen käytettävyystestaus vaatii testaaajalta jonkin verran lisäpanosta verrattuna tavalliseen testaukseen, luvussa 5 tarjotaan ratkaisuja näihin erityiskysymyksiin. Luku 6 kokoaa yhteen saadut tulokset ja arvioi tutkielman merkitystä sekä tulevaisuuden tutkimustarpeita.

2. WWW-sivuston kansainvälistäminen

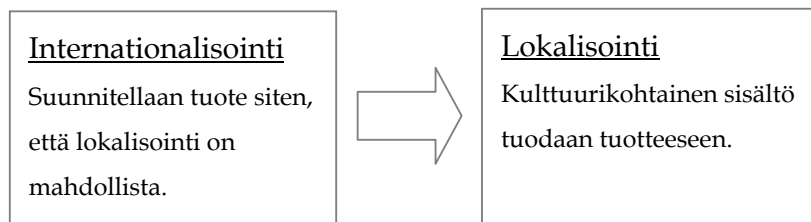
2.1. Kulttuuri ja lokaali

Kansainvälistämisessä on tärkeää ymmärtää kulttuurin moniulotteinen olemus ja lokaalin käsite. Hofstede [1996] määrittelee kulttuurin mielen ohjelmistoksi ("software of the mind"). Hänen mukaansa kulttuuri mieleemme ohjelmituna ohjaa päivittäistä toimintaamme. Kulttuuri on opittu ja kollektiivinen ilmiö, joka erottaa ihmisryhmät toisistaan. Kulttuuri vaikuttaa siihen, miten toimimme, ja millaisia tulkintoja ja merkityksiä annamme asioille. Kulttuurin osatekijöitä ovat uskomukset, arvot, tavat, perinteet ja kieli. Tässä tutkielmassa kulttuuri viittaa ihmisryhmään, jota yhdistää vähintään asuinpaikka ja kieli, ja johon voidaan viitata maan nimellä ('Ruotsi') tai kansalaisuudella ('ruotsalaiset').

Lokaali puolestaan viittaa suhteeseen kielen ja maa-alueen välillä [Käpyaho, 2001]. Joitakin kieliä puhutaan eri puolilla maailmaa, esimerkiksi ranska on virallinen kieli muun muassa Ranskassa, Sveitsissä, Belgiassa, Kanadassa sekä joissain Afrikan valtioissa. Usein kielen ilmauksissa on lokaalikohtaisia eroja, lähinnä tavoissa ilmaista päivämääriä, aikaa ja numeroita. Myös valuutta ja sitä edustavat symbolit saattavat poiketa toisistaan. Poikkeavuuksien vuoksi on tärkeää erottaa lokaalit toisistaan.

2.2. Internationalisointi ja lokalisointi

Tuotteen tai palvelun kansainvälistäminen käsittää kaksi askelta: internationalisoinnin ja lokalisoinnin [Russo and Boor, 1993]. Internationalisointi tarkoittaa tuotteen suunnittelua siten, että lokalisointi on mahdollista. Käytännössä internationalisoinnissa luodaan edellytykset tuoda tuotteeseen lokalisoituja sisältöjä. Lokalisointi on tuotteen sopeuttamista kohdekulttuurin vaatimuksiin. Lokalisoinnissa tuotteeseen tuodaan kohdekulttuurin vaatimusten mukainen sisältö. Alla oleva kaavio (Kuva 1.) kuvaa kansainvälistämisprosessin vaiheita. On huomattava, että internationalisointi edeltää lokalisointia.



Kuva 1. Kansainvälistämisprosessi.

2.3. Kulttuuri ja käytettävyys

Eräs suuntaus ihmisen ja tietokoneen vuorovaikutuksen alalla on pyrkimys universaaliin käytettävyyteen. Tämän suuntauksen mukaan sovellusten ja laitteiden tulisi vastata kaikenlaisilta käyttäjäryhmiltä, myös eri kulttuureista, tuleviin käytettävyysvaatimuksiin [UniversalUsability, 2000]. World Wide Web Consortiumin pitkän tähtäimen tavoitteisiin kuuluu myös WWW:n universaali, erilaiset kulttuurit ja kielet huomioiva, saatavuus [W3C, 2003]. Toisinaan ajatellaan, että käyttöliittymän kääntäminen kohdekielelle riittää tyydyttämään kansainvälisen käytettävyyden vaatimuksen. On kuitenkin tulossa yhä selvemmäksi, että kieli on vain osa tuotteen kansainvälistämistä [Evers, 2002]. WWW-palveluissa on mahdollista käyttää ääntä, kuvaa ja liikkuvaa kuvaa, ja ne kaikki on huomioitava kansainvälistämisessä.

Patricia Russo ja Stephen Boor [Russo and Boor, 1993] ovat valmistelleet tarkistuslistan asioista, jotka tulee ottaa huomioon kansainvälisessä suunnittelussa. Seuraavaksi esitellään huomioonotettavat asiat esimerkkien avulla.

Teksti. Käyttöliittymän tekstin kääntäminen kohdekielelle on ensimmäinen askel. Slangin tai muun erikoistyylin ilmaukset eliminoidaan, koska ne eivät yleensä ole ymmärrettäviä kaikissa kulttuureissa. Haastavaa on myös löytää kelvolliset vastineet sanoille, joita ei ole kohdekielessä. Suunnitteluvaiheessa on otettava huomioon erilaisten merkkijärjestelmien tuki, kuten diakriittisten merkkien (esim. ë, ã) syöttö ja esittäminen. Myös tuotenimet on mietittävä huolellisesti. Usein tuotteen nimi ei suoraan käänny järkevästi kohdekielelle.

Numerot, päivämäärä- ja aikaformaattit. Vaikka arabialaisia numeroita käytetään suuressa osassa maailmaa, numeroformaattit voivat vaihdella maiden välillä. Euroopassa yleinen käytäntö on erottaa luvun desimaaliosa pilkulla (2,5), Yhdysvalloissa desimaalit erotetaan pisteellä (2.5). Myös päivämäärän esitystavat vaihtelevat. Euroopassa yleisin formaatti on päivä-kuukausi-vuosi, kun taas Yhdysvalloissa kuukausi-päivä-vuosi on yleinen. Erilaiset kalenterit ja ajanlaskutavat on myös huomioitava suunnittelussa. Kellonajan ilmaisemiseen del Galdo [1990] suosittelee sekä 12 että 24 tunnin notaation sallimista.

Kuvat. Kuvat ovat kulttuurin kielen visuaalisia edustuksia. Kuvilla voi olla eri merkitykset eri kulttuureissa, ja jossain kulttuurissa jokin kuva ei välttämättä edusta mitään. Kuvan tunnistettavuus ja ymmärrettävyys ovatkin seikkoja, joita tulee pohtia kansainvälisessä suunnittelussa. Vaikka kuva olisi ymmärrettävä, se ei välttämättä ole hyväksyttävä. Erityisesti uskonnollisia symboleja sisältävien kuvien käytössä kannattaa olla varovainen. Tarvittaessa kuvia on muokattava vastaamaan kohdekulttuurin normeja tai korvattava kokonaan toisella kuvalla.

Symbolit. Symbolien käyttö saattaa altistaa väärinymmärryksille kulttuurien välillä. Kuten edellisessä kappaleessa mainittiin, erityisesti uskonnolliset symbolit ovat tulenarkoja. Käytettävien symbolien merkitys kohdekulttuurissa tulee selvittää huolellisesti, koska väärin käytetty symboli voi pilata tuotteen mahdollisuudet menestyä.

Värit. Värien tulkinta vaihtelee merkittävästi kulttuurien välillä. Suomalaiselle punainen ilmaisee vaaraa, mutta kiinalaiselle se merkitsee onnellisuutta. Jos värejä käytetään informaation välittämiseen käyttäjille, värien tulee olla vaihdettavissa sopiviin lokalisointivaiheessa.

Luku- ja kirjoitussuunta. Tekstin ja grafiikan asettelun näytölle tulisi tukea kulttuurin loogista tiedonkulkusuuntaa. Kun meillä luetaan vasemmalta

oikealle, esimerkiksi kiinassa ja hepreassa suunta on oikealta vasemmalla. Hyvässä suunnittelussa mahdollistetaan tiedon järjestämisen muutokset.

Toiminnallisuus. Kuten kuvat, symbolit ja värit, myös tuotteen ominaisuudet eivät välttämättä ole globaalisti päteviä. Toiminnallisia ominaisuuksia voi arvioida esimerkiksi seuraavassa luvussa käsiteltävän Hofsteden kulttuurien luokittelun avulla. Esimerkiksi Suomen kaltaisessa pienen valtaetäisyyden kulttuurissa käyttäjä saattaisi arvostaa yksilöllisiä ja vapaita ominaisuuksia, kun taas jossain suuren valtaetäisyyden kulttuurissa asia voisi olla päinvastoin.

3. Kulttuurien välisen viestinnän erityispiirteitä

Kuten luvussa 2 todettiin, kulttuuri on toimintaamme osaltaan määrittelevä tekijä. Jokainen vieraiden kulttuureiden kanssa tekemisiin päässyt ymmärtää, että ihmiset näkevät saman asian eri tavoin juuri kulttuurista riippuen. Meillä on myös havaintojemme pohjalta muodostettuja käsityksiä muista kulttuureista. Käsitykset saattavat olla varsin paikkansapitäviä, mutta joskus niiden takana voi olla kärjistettyjä stereotyyppioita, joita ei käytännön elämässä tapaa. Kansainvälisiä käytettävyydesteistä tekevät joutuvat väistämättä pohtimaan myös kulttuurien välisen viestinnän kysymyksiä. Vieraan kulttuurin erityispiirteisiin tutustuminen ja ennakkoluulottomuus auttavat onnistumaan haasteellisessa tehtävässä.

Geert Hofstede [1996] on laajoissa tutkimuksissaan päätenyt luokittelemaan kulttuureja dimensioiden avulla. Niitä tarkastelemalla voi laajentaa näkemystään maailmasta, ja hahmottaa vieraita kulttuureja suhteessa omaan kulttuuriin. *Valtaetäisyys* tarkoittaa sitä, miten kulttuurissa suhtaudutaan vallanpitäjiin ja auktoriteetteihin. Pienen valtaetäisyyden kulttuurissa, kuten Suomessa, työntekijällä ja esimiehellä voi olla kaverillinen suhde, mutta suuren valtaetäisyyden kulttuurissa hierarkkiset rakenteet ja niiden välinen kanssakäyminen ovat jäykkiä ja tarkoin määriteltyjä. *Epävarmuuden sietokyky* viittaa siihen, missä määrin kulttuurissa vieraat ja uudet asiat koetaan uhkaaviksi ja vältettäväksi. Kulttuurin sijoittuminen *feminiinisyyden-maskuliinisuus* - dimensiolla kertoo, kannatetaanko mieluummin pehmeitä arvoja, kuten sukupuolten välistä tasa-arvoa, vai kovempia, miehisiä arvoja, kuten mahdollisuutta huippuansioihin. *Kollektiivisuus-individuaalisuus* - dimensio indikoi, ovatko vallalla yksilölliset vai yhteisön etua korostavat arvot. *Pitkän tähtäimen orientaation* kulttuureissa sitkeys ja peräänantamattomuus ovat merkitseviä piirteitä, kun taas lyhyen tähtäimen orientaation kulttuureita leimaa turvallisuuden ja vakauden tunne sekä perinteiden arvostus. Kulttuurien kuvauksia tarkastellessa tulee muistaa, että ne koskevat kokonaisuutta, ja huomattavia yksilöllisiä eroja löytyy kulttuurien sisältä.

Vieraan kulttuurin asiantuntijaksi kehittyminen vaatii pitkäaikaista oleskelua kyseisessä maassa, mutta kulttuurin piirteitä voi pyrkiä ymmärtämään opiskelun ja lyhytaikaisenkin kanssakäymisen kautta. Kansainvälisiä käytettävyydestejä tekevä voi välttää kulttuurierojen synnyttämiä ongelmia testien yhteydessä, jos hän on selvillä kulttuurien välisen viestinnän erityispiirteistä.

4. Kansainvälisen käytettävyydestestauksen menetelmät

Käytettävyydestestauksessa pyritään rekrytoimaan koehenkilöiksi todellisia prospektiivisiä käyttäjiä [Dumas, 1993]. Kansainvälisessä käytettävyydestestauksessa on mielekästä ottaa koehenkilöiksi kohdekulttuurien edustajia, koska vain silloin hankittua tietoa voidaan pitää pätevänä. Tässä luvussa esitellään menetelmiä testauksen suorittamiseksi ja arvioidaan niiden soveltuvuutta erilaisiin projekteihin. Viimeiseksi pohditaan, ovatko kaikki menetelmät hyviä minkä tahansa kulttuurin kohdalla. On nimittäin näyttöä, että tiettyjä menetelmiä ei voida menestyksekkäästi soveltaa joissakin kulttuureissa [Evers, 2002].

4.1. Kansainvälinen arviointi (International inspection)

Kansainvälinen arviointi ei varsinaisesti ole käytettävyydestestauksen menetelmä, mutta se on yksi erinomainen tapa saada tietoa sivuston sopivuudesta kansainväliselle yleisölle. Sen voi järjestää rekrytoimalla eri kulttuurien edustajia, mieluiten käytettävyyden asiantuntijoita, arvioimaan sivustoa oman kulttuurinsa näkökulmasta ja raportoimaan mahdollisista ongelmista. Käytettävyydasiantuntijoilta voi pyytää arvion jo määrittely- tai luonnosvaiheessa, jolloin ongelmat saadaan korjattua hyvissä ajoin.

Jos käytettävyydasiantuntijoiden käyttäminen arvioinnissa ei ole mahdollista esimerkiksi pienen budjetin vuoksi, arvioinnissa voi käyttää apuna vaikkapa oman yrityksen ulkomaan toimistojen työntekijöitä. On huomattava, että ei-ammattilaisilla saattaa olla vaikeuksia hahmottaa käyttöliittymä pelkän kirjoitetun spesifikaation perusteella, joten tässä tapauksessa arvioinnin kohteena kannattaa käyttää valmista sivustoa tai prototyyppiä.

Vaikka kansainvälinen arviointi voi osoittautua varsin hyödylliseksi menetelmäksi, se ei kuitenkaan korvaa varsinaista käytettävyydestestausta. Parhaimmillaankin siitä saadut tulokset ovat vain yksilöiden omien havaintojen pohjalta tekemiä arvauksia. [Nielsen, 2000]

4.2. Testaaja matkustaa

Käytettävyydestestaus käyttäjän omassa ympäristössä on monin tavoin hyödyllistä. Käyttötilanne on silloin mahdollisimman aito ja todellisuutta

vastaava. Vieraassa ympäristössä testaaja voi tehdä havaintoja, jotka auttavat ymmärtämään paremmin vierasta kulttuuria, ja joita voi soveltaa sivuston suunnittelussa. Jos testaaja matkustaa ulkomaille tekemään käytettävyydestejä, hän voi yrittää järjestää testit käyttäjän tiloihin, jolloin testitilanne ja -ympäristö vastaavat mahdollisimman hyvin aitoa käyttötilannetta.

Koska matkustaminen on kallista, kannattaa harkita mahdollisuutta suorittaa epävirallisia käytettävyydestejä jonkin muun syyn takia tehdyn matkan aikana. Testejä voi epävirallisissa tapauksissa suorittaa missä tahansa tietokoneen äärellä, kunhan testattava sivusto saadaan ladattua näytölle. [Nielsen, 2000]

4.3. Etätetaus (Remote testing)

Käytettävyydestejä voi suorittaa myös siten, että käyttäjä ja testaaja ovat eri paikoissa. Kumpikin voi siis olla kotimaassaan. Riittää, että molemmat pääsevät testattavalle sivustolle Internetin kautta, ja että heidän välillään on toimiva kommunikointiyhteys. Tällöin tarvittava puheyhteys saadaan aikaan tavallisen puhelun, videoneuvottelulaitteiston tai Internet-puhelun avulla.

Etätetauksen huonona puolena on se, että testaaja ei voi visuaalisesti havainnoida käyttäjän toimintaa sivustolla, vaan toiminnasta saatava tieto välittyy pääasiassa käyttäjän kertoman perusteella. Etuna tässä menetelmässä on puolestaan alhaiset kustannukset, koska matkustamista ei tarvita. [Nielsen, 2000]

4.4. Itsenäinen testaus (Self-administered tests)

Testauksen voi suorittaa myös ilman testin johtajaa. Tehtävät voi lähettää koehenkilöille ja pyytää heitä raportoimaan tuloksista. Tässä etätetauksen muodossa tehtävät kannattaa suunnitella siten, että niistä raportoiminen on yksinkertaista, ettei väärinymmärryksiä pääse syntymään. Esimerkiksi voi pyytää hakemaan sivustolta jonkin tiedon, ja raportoimaan tehtävän onnistumisesta ja suoritukseen kuluneesta ajasta. Vapaamuotoisia kommentteja voi myös kysyä. [Nielsen, 2000]

4.5. Testaaminen käytettävyydelaboratoriossa

Nielsen [2000] näkee, että kansainvälisten käytettävyydestien suorittaminen käytettävyydelaboratoriossa ei ole suinkaan välttämätöntä, vaan yksinkertaisinta on luottaa omiin vaikutelmiin ja muistiinpanoihin videonauhojen katselun sijaan. Merkittävimmät käytettävyysongelmat tulevat ilmi jo ensimmäisessä havainnointivaiheessa.

Toisaalta käytettävyydestien nauhoittaminen videolle mahdollistaa palaamisen testitilanteeseen, jos esimerkiksi kieliongelmiin vuoksi jokin seikka

on jäänyt testaajalta ymmärtämättä. Myös testaajan työtoverit voivat hyötyä videonauhasta.

4.6. Menetelmän valinta

Sopiva testausmenetelmä valitaan projektin laajuuden ja käytettävissä olevien resurssien perusteella. Jos resursseja allokoidaan riittävästi kansainväliseen testaukseen, matkustaminen ja testaaminen kohdekuulttuurissa tulevat kyseeseen. Jos taas matkustaminen ei onnistu, järjestetään hyvin suunniteltu etätestiasetelma. Kansainvälisen arvioinnin järjestäminen on suhteellisen vaivatonta ja edullista, jos ei rekrytoida käytettävyyden ammattilaisia, joten sitä kannattaa hyödyntää useilla iteraatiokierroksilla.

4.7. Menetelmien soveltuvuus eri kulttuureihin

Evers [2002] raportoi tapaustutkimuksesta, jossa verrattiin WWW-palvelun ymmärtämistä eri kulttuureista tulleiden opiskelijoiden kesken samoilla menetelmillä arvioituna. Tutkimuksessa ilmeni, että menetelmien soveltuvuudessa eri kulttuureihin on eroja. Englantilaisilla ääneen ajatteluun tottuminen vaati vahvistusta enemmän kuin hollantilaisilla ja pohjoisamerikkalaisilla. Japanilaisille yksilöllisen toiminnan tarkkailu ja ääneen ajattelu aiheutti epä mukavan olon, koska he eivät siinä tilanteessa voineet pyrkiä yhteiseen näkemykseen muiden kanssa. Lisäksi Itä-Aasian kulttuureissa suunnittelijaan kohdistuvaa avointa kritiikkiä ei pidetä hyväksyttävänä, koska se saattaa aiheuttaa suunnittelijalle kasvojen menettämisen [Fernandes, 1995]. Aiheesta tarvitaan lisätutkimuksia, mutta tuloksia odotellessa kansainvälisiä käytettävyydestejä suorittavan kannattaa ainakin yrittää miettiä valitsemansa menetelmän mahdollisia ongelmia ja niiden ratkaisemista. Muuten käytettävyydesteillä haettava uusi tieto jää hyvin vähäiseksi, eikä tuloksia voida pitää pätevinä [Yeo, 1998].

5. Testauksen erityispiirteitä

5.1. Kielikysymykset

Kieliongelmiä kohdataan sekä etätestauksessa että kohdekuulttuuriin matkustettaessa. Testikielen valintaan vaikuttaa osapuolien kielitaito. Kieleksi valitaan jokin kieli, jonka kaikki osapuolet hallitsevat riittävän hyvin. Kieltä valitessa on varmistuttava, että kaikkien kielitaito todella riittää WWW-sivuston käyttökokemuksesta kommunikointiin [Nielsen, 2000]. Jos yhteistä kieltä ei löydy, on parasta ja välttämätöntä turvautua tulkin apuun.

Merholz [2003] suosittelee vieraskieliseen testaukseen seuraavanlaista asetelmaa: kohdekielen natiivi avustaja testihuoneeseen, reaaliaikaisia

käännöksiä antava tulkki tarkkailuhuoneeseen, testaaja itse tekemässä muistiinpanoja tarkkailuhuoneessa. Kupper [2000] puolestaan näkee tulkin käytön ongelmallisena ja perustelee näkemystään informaation menetyksellä. Sen sijaan hän suosittelee kokeneen, kohdekieltä sujuvasti puhuvan käytettävyyssasiantuntijan toimimista tulkkina.

5.2. Testitehtävien suunnittelu

Kansainvälisessä käytettävyystestauksessa tehtävien suorittamiseen kuluu todennäköisesti enemmän aikaa kuin tavallisissa testeissä. Nielsen [2000] suosittelee varaamaan 50 % enemmän aikaa jokaista testitehtävää kohti tavalliseen testaukseen verrattuna. Myös kansainvälisessä arvioinnissa ja itsenäisessä testauksessa tulee varautua aikaeroista johtuviin viiveisiin [Merholz, 2003].

Kielikysymykset vaikuttavat myös testitehtävien suunnitteluun. Tarkasti määritelty tehtävä pienentää tulkintaerojen todennäköisyyttä. Kannattaa varmistaa, ettei tehtävien käännöksessä häviä oleellista tietoa.

5.3. Tulosten arviointi

Analyysivaiheessa on otettava huomioon kansainvälisyyden ja kulttuurien välisen kanssakäymisen mahdolliset vaikutukset tuloksiin. Usein testauksen yhteydessä saadaan informaatiota sivuston käytettävyyso Ongelmien lisäksi käyttäjien suhtautumisesta palveluntarjoajaan yleisesti. Näistä havainnoista saattaa muodostua testauksen arvokkain tulos. Käytettävyystudkimus ei ehkä ainoastaan paranna sivuston käyttöliittymää, vaan auttaa palveluntarjoajaa ymmärtämään oman paikkansa maailmassa [Merholz, 2003].

6. Yhteenveto

Kansainvälisen käytettävyyden saavuttaminen WWW-suunnittelussa ja kansainvälinen käytettävyystestaus ovat haasteellisia tehtäviä. Pelkkä käytettävyyden asiantuntemus ei riitä - lisäksi on hankittava tietoa vieraista kulttuureista ja kansainväliselle yleisölle suunnittelusta. Kohdekulttuuriin tutustuminen auttaa löytämään soveltuvia suunnitteluratkaisuja sekä toimimaan testitilanteissa joustavasti. Testauksen toteuttaminen vaatii myös huolellista esityötä ja enemmän aikaa kuin tavallinen käytettävyystestaus. Kansainvälisyys tuo testeihin ylimääräisiä muuttujia, joita testaajan on pyrittävä kontrolloimaan. Tulosten arvioinnissakin kulttuurierojen vaikutus tulee punnita.

Kirjallisuuskartoituksessa ilmeni, että testausta varten on olemassa monia menetelmiä, joista voi valita omaan projektiin sopivimman. Käytettävyyssasiantuntijien palkkaaminen ja laajamittaisten testien teettäminen

käytettävyysslaboratorioissa ulkomailla ovat yleensä vain kaikkein suurimpien yritysten ulottuvilla. Pienillä resursseilla voi silti hankkia arvokasta tietoa sivustonsa kansainvälisestä käytettävyydestä. Oleellista on löytää vähintään yksi kohdekulttuurin edustaja arvioimaan sivustoa omasta näkökulmastaan. Internetin käyttö viestintäkanavana helpottaa tehtävää, jos matkustaminen ei tule kyseeseen.

Kansainvälisen WWW:n haasteet ulottuvat myös tässä tutkielmassa käsiteltyjen asioiden ulkopuolelle. Tutkielmasta rajattiin tietoisesti pois tärkeitäkin suunnitteluun liittyviä seikkoja, kuten käyttöliittymän kielen valintaan vaikuttavat tekijät ja kansainvälistämisen teknisten ratkaisujen laajempi käsittely. Käytettävyydestäukseen liittyen tässä tutkielmassa ei otettu kantaa koehenkilöiden valintaan eikä siihen, kuinka monta koehenkilöä yleensä tulisi rekrytoida. Yleinen linja kuitenkin on, että jos halutaan saada selville jonkin kulttuurin trendi, tarvitaan vähintään 6 – 8 koehenkilöä. Suuntaantavan arvion voi saada jo yhden koehenkilön avulla.

Kansainvälisen käytettävyydestäuksen haasteita oli mielenkiintoista selvittää, koska pyrkimys globaaliin tietoverkkoon tekee aiheesta ajankohtaisen, ja sitä voidaan lähestyä monista näkökulmista. Internet-teknologiat, ihmisen ja tekniikan vuorovaikutus, ja erilaisten ihmisten välinen vuorovaikutus sekä kasvokkain että verkossa luovat puitteet monipuolisille tutkimusongelmille. Aiheen ympäriltä löytyisi vielä paljon tutkittavaa. Erityisen mielenkiintoista olisi tutkia empiirisesti käytettävyydestäuksen menetelmien soveltuvuutta eri kulttuureihin, jotta käytettävyydestäusta voitaisiin suorittaa tehokkaammin.

Viiteluettelo

- [Dumas, 1993] Joseph S. Dumas, *Practical Guide to Usability Testing*. Ablex Publishing Intellect, 1993.
- [Evers, 2002] Vanessa Evers, Cross-cultural applicability of user evaluation methods: A case study amongst Japanese, North-American, English and Dutch users. In: *CHI'02 Extended abstracts on Human factors in Computer systems*, 2002.
- [Fernandes, 1995] Tony Fernandes, *Global Interface Design: A Guide to Designing International User Interfaces*. Academic Press Professional, 1995.
- [del Galdo, 1990] Elisa del Galdo, Internationalization and translation: Some guidelines for the design of human-computer interfaces. In: Jakob Nielsen (ed.), *Designing User Interfaces for International Use*. Elsevier, 1990.
- [Hofstede, 1996] Geert Hofstede, *Cultures and Organizations: Software of the Mind*. The McGraw-Hill Companies, 1996.

- [Kupper, 2000] Ansgar Kupper, Tips for international usability testing. Systemconcept, 2000. Available at: <http://www.system-concepts.com/articles/global.html> (last accessed 12.12.2003).
- [Käpyaho, 2001] Jere Käpyaho, Internationalisation in Operating Systems for Handheld Devices. Master's Thesis, Department of Computer Science, University of Tampere 2001. Available at: http://www.cs.uta.fi/research/theses/masters/Kapyaho_Jere.pdf (last accessed 10.12.2003).
- [Nielsen, 2000] Jakob Nielsen, *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing, 2000.
- [Merholz, 2003] Peter Merholz, Conducting international usability. Adaptive Path, 2003. Available at: <http://www.adaptivepath.com/publications/essays/archives/000098.php> (last accessed 12.12.2003).
- [Russo and Boor, 1993] Patricia Russo, Stephen Boor, How fluent is your interface? Designing for international users. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1993.
- [UniversalUsability, 2000] Universal usability issues. Universalusability.org, 2000. Available at: <http://www.universalusability.org/index.html> (last accessed 11.12.2003).
- [W3C, 2003] About the World Wide Web Consortium. World Wide Web Consortium, 2003. Available at: <http://www.w3.org/Consortium/#goals> (last accessed 11.12.2003).
- [Yeo, 1998] Alvin Yeo, Cultural effects in usability assessment. In: *CHI 98 conference summary on Human factors in computing systems*, 1998.

Kulttuurierojen huomioiminen käyttöliittymäsuunnittelussa

Minna Nurminen

Tiivistelmä.

Tutkimukseni käsittelee kulttuurierojen vaikutusta käytettävyyteen. Kulttuurit voivat erota toisistaan pintatason seikkojen, kuten kielen, ja kulttuurisidonnaisten seikkojen, kuten värien ja symbolien tulkinnan suhteen. Niillä voi olla myös erilaisia yhteiskunnallisia arvoja, jotka on otettava huomioon käyttöliittymiä suunniteltaessa. Kulttuurillisten ominaisuuksien tulkinnan pohjana käytetään yleisesti Hofsteden teoriaa. Tutkimuksessani olen yhdistellyt tietoja useista artikkeleista, joissa käsitellään niin lokalisointia ulkoasun ja kulttuurillisten ominaisuuksien kannalta. Tarkoitukseni on antaa yleiskuva kolmesta eri lokalisoinnin tasosta: pintatason, kulttuurillisen tason ja yhteiskunnallisen tason lokalisoinnista.

Avainsanat ja -sanonnat: Käytettävyyssuunnittelu, kulttuurit, lokalisointi
CR-luokat: D 2.2, H 1.2, H 5.2

1. Johdanto

Tutkimuksessani käsittelen kulttuurierojen vaikutusta käytettävyyteen. Tutkin eri kulttuureille ominaisia tekijöitä, jotka vaikuttavat siihen, miten käyttöliittymien parissa toimitaan. Tutkimukseni kannalta tärkeitä ovat visuaaliset seikat, kuten värien ja symbolien merkitykset. On tärkeää tietää minkälaista kuvamateriaalia sivuilla voi esittää. Hyvän maun rajat ja moraalikysymykset on otettava huomioon. Käsittelen myös yhteiskunnallisia seikkoja, jotka vaikuttavat yleisesti toimintaan käyttöliittymien ja tietokoneiden parissa. Käytän tulkinnoissani apuna mm. Hofsteden teoriaa, jossa kulttuureja luokitellaan viiden niiden arvoihin ja asenteisiin liittyvän ominaisuuden perusteella.

Kulttuurierojen tutkiminen ei pääty vain sivustojen ulkonäön, sisältömaterialin ja rakenteen pohtimiseen, vaan jatkuu vielä erilaisen toiminnallisuuden pariin. Käsittelen myös sitä, onko tarpeellista ottaa kulttuurierot huomioon suunniteltaessa käyttöliittymän toimintoja.

Kulttuurierojen huomioonottaminen säästää väärinkäsityksiltä ja jopa loukkauksilta. Aihepiiri on tärkeä esimerkiksi monikansallisten yhtiöiden kannalta, jotka laajentavat toimintaansa kotimaan rajojen ulkopuolelle. Erilaisten käyttöliittymien suuntaaminen eri kulttuureille kertoo yrityksen

halusta ottaa eri kulttuuritaustaiset asiakkaansa huomioon. Perusteellinen suunnittelu välittää asiakkaille tunteen siitä, että yritys arvostaa heitä ja on valmis näkemään vaivaa saadakseen käyttöliittymästä kullekin kulttuuri-ryhmälle sopivan.

Tutkielmassani esittelen ensin käyttöliittymän ulkoasuun liittyviä seikkoja, jotka olen jakanut kahteen ryhmään: pintatason ja kulttuurillisen tason elementteihin. Sitten käsittelen yhteiskunnallisia ominaisuuksia Hofsteden teorian pohjalta.

2. Käyttöliittymän ulkoasuun liittyvät seikat

Seuraavaksi käsittelen käyttöliittymäelementtejä, joihin tulee kiinnittää huomiota suunniteltaessa käyttöliittymää toiseen kulttuuriin. Alaluvuissa kerron yksityiskohtaisesti elementtien ominaisuuksista ja erityispiirteistä.

Käyttöliittymän valmistelu kansainvälisille markkinoille on kaksivaiheinen. Russo ja Boor [1993] käyttivät vaiheista termejä internationalization ja localization. Heidän mallissaan ensimmäisessä vaiheessa käyttöliittymä muutetaan kansainväliseksi ja toisessa lokalisoidaan. Kansainväliseksi muuttaminen käsittää kaikkien kulttuurillisten elementtien identifioinnin ja muuttamisen kohdekulttuuriin sopiviksi. Tällaisia elementtejä ovat teksti, numerot ja päivämäärät. Huolellisesti tehdyssä lokalisoinnissa mennään vieläkin pidemmälle ja kiinnitetään huomiota edellä mainittujen elementtien lisäksi myös seikkoihin, jotka liittyvät kulttuurin arvoihin, etikkaan ja moraaliin.

Myös Sun [2001] puhuu kaksivaiheisesta lokalisoinnista. Hänen mukaansa ensimmäiseen vaiheeseen, pintatason lokalisointiin, kuuluu samat elementit kuin Russolla ja Boorilla. Toiseen vaiheeseen eli kulttuurilliseen tasoon kuuluvat kuvien, värien, loogisuuden, estetiikan ja toiminnallisuuden lisäksi myös kulttuurilliset tekijät. Niitä ovat mm. yhteiskunnan ja työpaikan hierarkkinen rakenne, erot tiedon käsittelyssä sekä kulttuurin jäsenten jakama tieto.

Kulttuurilliset elementit voidaan Barberin ja Badren [1998] mukaan jakaa taulukon 1 mukaisiin kategorioihin. Kategorioiden alle on merkitty esimerkkejä niihin liittyvistä elementeistä.

Taulukko 1. Kulttuurikohtaiset elementit web-sivuilla

HTML	Ikonit/vertauskuvat	Värit	E erityiset värit	Ryhmittely
Linkkien värit, taulukot,	kellot, sanomalehdet,	punainen, sininen,	liput, grafiikka, kuvat, reunat,	symmetrinen, epäsymmetrinen,

kehykset, alleviivaukset, ääni, video, taustakuvat ja - värit	kirjat, sivut, postimerkit, kirjekuoret, nuotit,	vihreä, violetti, pinkki, musta, keltainen...	tausta	läheisyys, taseus, reunukset, sulkeutuminen, yhteydet
Lippu	Kieli	Maantiede	Sommitelu	Ääni
oman maan, muun maan, monia	oman maan, muun maan, monia	kartat, maapallo	keskitetty, vasen -> oikea, oikea -> vasen	musiikki, puhe
Kirjasin-tyyli	Linkit	Paikalliset	Muodot	Arkkitehtuuri
kursiivi, paksunnettu, koko, varjostus	väri, sijoittelu, sisäiset, ulkoiset	kasvillisuus, eläimet, maisemat, vesi, aavikko	neliöt, ympyrät, kolmiot, suorakulmiot, viivat, nuolet	valtaa edustavat rakennukset, talot, kirkot, toimistot, kaupunkikuvat

2.1. Pintatasolla tapahtuva lokalisointi

Sunin [2001] mukaan pintatason lokalisointiin kuuluu tekstin, päivämäärien, numeroiden ja ajan muuttaminen kohdekulttuuriin sopivaksi.

2.1.1. Teksti

Ensimmäinen vaihe lokalisaatiossa on tekstin kääntäminen kohdemaan kielelle. Käyttöliittymän kääntäminen on usein hankalaa, koska siinä on otettava huomioon konteksti ja käyttöliittymän toiminta. On tiedettävä sanojen tarkoitukset tietokoneympäristössä. Esimerkiksi sanan 'menu' kääntäminen ruokalistaksi ei toimi tässä kontekstissa. Ideaalitapauksessa kääntäjä toimii yhteistyössä käyttöliittymäspesialistien ja ohjelma-asiantuntijoiden kanssa [Russo and Boor, 1993].

Jargonin käyttöä olisi vältettävä. Teksti tulisi ensin kääntää 'oikealle kielelle' ennen kääntämistä kohdemaan kielelle. Jargonin karsiminen pois käyttöliittymästä on tärkeää, koska harva ymmärtää edes englanninkielistä tietokonealan erikoiskieltä [Russo and Boor, 1993].

Englanti on maailman nopeimmin kasvava kieli. Siihen keksitään jatkuvasti uusia, varsinkin tekniikkaan liittyviä, sanoja. Siksi termien kääntäminen toiselle kielelle on vaikeaa ja joskus jopa mahdotonta, koska muissa kielissä sanasto ei välttämättä ole yhtä rikas. Tällöin on hyvä säilyttää sanat sellaisenaan ja tarjota niille selitys kohdemaan kielellä [Russo and Boor, 1993]. Sukaviryan ja Moranin [1990] tekemässä tutkimuksessa huomattiin, että käyttäjät pitivät

sanojen alkuperäistä muotoa käännöstä parempana kun he olivat oppineet sanan tarkoituksen.

Suunnittelijoiden on myös otettava huomioon erilaiset kirjaimistot. Eurooppalaiset kielet sisältävät monia erikoismerkkejä, joiden näyttö ja syöttö on mahdollistettava käyttöliittymässä [Russo and Boor, 1993].

Muita kielialueita ei useinkaan oteta huomioon kun valitaan tuotteelle nimeä. Joissain tapauksissa nimi tarkoittaa jotain toisessa kielessä. Noloa on, jos nimi tarkoittaaakin jotain negatiivista [Russo and Boor, 1993].

2.1.2. Numerot, päivämäärät ja aikakäsitteet

Useassa maassa käytetään arabialaista numerojärjestelmää. Eroja kuitenkin ilmenee numeroiden esittämisessä esimerkiksi Euroopan ja Pohjois-Amerikan välillä. Eurooppalaiset erottavat pilkulla desimaalit (2,1) ja käyttävät pistettä suurissa luvuissa (1.000.000). Amerikkalainen tapa on käyttää pistettä desimaaliluvuissa (2.1) ja pilkkua suurissa luvuissa (1,000,000) [Russo and Boor, 1993].

Päivämäärien esittämiseen on olemassa monta erilaista tyyliä. Euroopassa päivämäärät ovat muotoa päivä/kuukausi/vuosi. Kun taas Amerikassa käytetään kuukausi/päivä/vuosi-muotoa. Onkin suositeltavaa mahdollistaa käyttöliittymässä kahden edellä mainitun käytön lisäksi vielä kolmas muoto: vuosi/kuukausi/päivä. Ei-länsimaisissa maissa on usein käytössä erilainen ajanlasku. Tällaisia maita ovat esimerkiksi Japani, Kiina ja Israel [Russo and Boor, 1993].

Kellonaikojen näyttö eroaa maiden välillä. Voidaan käyttää 12-tuntista tai 24-tuntista aikaa. Joissain maissa myös sekuntien näyttäminen kuuluu asiaan [Russo and Boor, 1993].

2.2. Kulttuurillisella tasolla tapahtuva lokalisointi

Seuraavaksi käsittelen kulttuurillisen tason lokalisointia, johon kuuluu kuvien, symbolien, värien, sommittelun, toiminnallisuuden, vertauskuvien sekä mentaalisten mallien lokalisointi.

2.2.1. Kuvat

Kuvien käyttäminen symboleina käyttöliittymissä on yleistä. Kaikkein tunnetuin kuvallinen vertauskuva on Microsoftin Windows-käyttöliittymä, jossa työpöytä käyttää hyväkseen toimisto-metaforaa. Se tuo mukanaan ongelmia kuvien ja niiden yhteyksien ymmärtämisen suhteen. Käyttöliittymässä on roskakori ja kansioita. Roskakori on yleensä länsimaisesti ymmärrettävä, mutta muissa kulttuureissa se voi aiheuttaa hämmennystä. Sukaviriya ja Moran [1990] huomasivat tutkimuksessaan, etteivät thaimaalaiset

ymmärtäneet Applen työpöydän roskakorin tarkoitusta. Heidän kulttuurissaan tyyppillinen roskakori on punottu koppa, jonka ympärillä pörrää kärpäsiä. Myös Briteillä on Russon ja Boorin [1993] mukaan ongelmia Applen roskakorin kanssa. Heidän mielestään se näyttää enemmänkin postilaatikolta. Yhden maan sisälläkin voi olla ongelmia ikonin ymmärtämisessä. Applen postilaatikon kuva, joka esitti tyyppillistä yhdysvaltojen maaseudun laatikkoa, jossa lippu nostetaan ylös postin saapumisen merkiksi, ei ollut selvä vertauskuva kaupunkien asukkaille.

Kuvien ymmärrettävyyden lisäksi niiden soveliaisuus on otettava huomioon. Erityistä huolellisuutta on käytettävä suunniteltaessa vartaloon, naisein, uskonnollisiin symboleihin ja käsimerkkeihin liittyvien kuvien käyttöä [Russo and Boor, 1993].

2.2.2. Symbolit

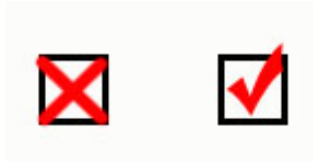
Risti on länsimaisessa kulttuurissa sekä uskonnollinen symboli kuin kieltomerkkikin. Myöskin valinnan ruksaaminen laatikkoon voi tapahtua rastilla tai väkäsellä, kuten kuvassa 1 [Russo and Boor, 1993]. Käyttäessä erilaisia käsimerkkejä on oltava varovainen, koska niillä on eri merkityksiä kulttuureissa. Dysart [2002] sanoo esimerkiksi kämmen ulospäin olevan käden merkitsevän uhkaavuutta Aasian maissa. Negatiivisia merki-tyksiä omaavat joissain kulttuureissa peukalon nostaminen sekä etusormi ja peukalo yhdessä muodostettava ok-merkki.

2.2.3. Värit

	Punainen	Sininen	Vihreä	Keltainen	Valkoinen
Egypti	kuolema	toivo, totuus, hyvyys	hedelmällisyys	onnellisuus, vauraus	ilo
Intia	elämä, luovuus	Krishna * (hindut)	vauraus, hedelmällisyys	menestys	kuolema, puhtaus
Japani	viha, vaara	roistomaisuus	tulevaisuus, nuoruus, energia	ylevyys, jalous,	kuolema
Kiina	onnellisuus	taivas, pilvet	Ming dynastia, taivas, pilvet	syntymä, rikkaus, valta	kuolema, puhtaus
Ranska	yläluokka	vapaus, rauha	rikollisuus	väliaikaisuus	neutraali
USA	vaara, "pysähdy"	maskuliinisuus	turvallisuus, "mene"	pelkuruus	puhtaus

Taulukko 2. Värien merkitykset kulttuureissa [Russo and Boor, 1993], *) [Holzschlag, 2000]

Taulukossa 2 käsitellään värien merkityksiä kulttuureittain. Erilaisten merkitysten lisäksi kulttuureissa on erilaisia mieltymyksiä värien suhteen. Kuten Simon [2001] havaitsi, tutkimukseen osallistuneista aasialaisista 72% piti hillittyjä sävyjä kirkkaita värejä parempina. He suosivat vihreitä sinisten sijaan. Eurooppalaisten ja amerikkalaisten enemmistö puolestaan suosi vaaleampia ja kirkkaampia sävyjä.



Kuva 1. Kaksi eri tapaa ruksata valinta

2.2.4. Sommittelu

Hedberg ja Brown [2002] esittelevät JAL-matkailuyrityksen esitteitä, jotka on suunnattu länsimaihin ja Japaniin. Länsimainen esite on ulkoasultaan selkeä ja yksinkertainen. Japanilaiselle esitystavalle on tyypillistä suurempi informaation sisältö, jolloin esitteen etusivu on täynnä tekstiä ja kuvia. Siinä on myös yritykseen tai mainostettavaan asiaan liittymättömiä piirroshahmoja koristeena.

He tutkivat myös kolmea Shell-yhtiön sivua, jotka ovat suunnattu australialaiselle, kiinalaiselle ja hongkongilaiselle yleisölle. Australialainen sivu oli käytössä myös useissa muissakin länsimaissa. Se oli ulkoasultaan järjestelmällinen. Valikot oli sijoitettu länsimaiseen tyyliin vasempaan reunaan. Sivun sommittelu koostui erilaisista suorakulmioista; valikoista, tekstialueista ja kuvista, sekä kirkkaista ja vahvoista väreistä. Hongkongilaisille suunnatut sivut kuvasivat kaupunkia sarjakuvamaisesti. Talojen katoilla ja teiden varsilla olevat kyltit toimivat linkkeinä alasivuille. Sivut sisälsivät myös animaatioita. Sommittelua ei oltu järjestetty tietyn kaavan mukaan, vaan linkit sijaitsivat sikin sokin sivulla. Kiinalainen etusivu oli ulkoasultaan rauhallinen ja hillitty. Se käytti hyväkseen itämaista kulttuuriperintöä kalligrafisten symbolien muodossa. Sivun sommittelu painottui keskelle, jossa kuva sijaitsi. Linkit oli sijoitettu kuvan ympärille.

2.2.5. Toiminnallisuus

Kulttuureilla voi olla erilaisia vaatimuksia toiminnallisuuden suhteen. Russo ja Boor [1993] puhuvat ranskalaisesta LYRE-ohjelmasta, jota lanseerattiin Skandinaviaan. Ohjelman tarkoituksena oli opettaa runojen tulkintaa eri näkökulmista. Opettaja voi lisätä näkökulmia, joita oppilaat pystyivät

käyttämään hyväkseen tulkinassa. Ranskassa LYRE toimi mainiosti, mutta Skandinaviassa sitä ei otettu yhtä hyvin vastaan. Skandinaavinen ja ranskalainen kulttuuri eroavat toisistaan siinä, että Skandinaviassa oppilaita kehoitetaan omien tulkintojen tekoon, mikä ei ole käytäntönä Ranskassa. Siksi Skandinavian markkinoille ohjelmaa olisi tullut muuttaa niin, että oppilaatkin olisivat voineet lisätä omia näkökulmiaan.

Simonin [2001] mukaan aasialaisille on tärkeää voida kommunikoida toisten kanssa. Tämä on hyvä ottaa huomioon myös käyttöliittymiä suunniteltaessa, varsinkin Internetkauppojen ollessa kyseessä. Simon muistuttaa myös Internetin lähtökohdan vaikutuksesta sen toiminnallisuuteen. Se on hyvin länsimainen järjestelmä, jossa annetaan suuri arvo rationaalisuudelle, teknologialle, kehitykselle, nopeudelle, ajan säästölle, individuaalisuudelle ja demokraattisuudelle. Kaikki kulttuurit eivät kuitenkaan arvosta näitä asioita, silloin on Simonin mukaan parasta käyttää yleisempää monia kulttuureja miellyttämään pyrkivää suunnittelua. Simonin tekemässä tutkimuksessa länsimaalaiset halusivat muuttaa Internetsivuja selkeämmiksi ja nopeakäyttöisimmiksi navigointipalkkien ja yksikertaisten työkalujen, kuten nuolien ja nappien avulla. Aasialaisten ja eteläamerikkalaisten ehdotukset keskittyivät sivujen ulkonäön parantamiseen.

2.2.6. Vertauskuvat ja mentaaliset mallit

Vertauskuvien käyttö voi aiheuttaa ongelmia kulttuureissa, joissa tietty asia tehdään toisin kuin muualla. Länsimaisille ihmisille niinkin arkipäiväinen ja itsestään selvä asia kuin kirjasto on maoreille tuntemattomampi käsite. Heillä tieto siirretään suullisena ihmiseltä toiselle ilman sen ylöskirjaamista. Dunckerin [2002] tekemän tutkimuksen mukaan maoreilla oli vaikeuksia länsimaisen tiedon järjestelyn kanssa, koska he ovat oppineet toisenlaisen tavan. Kirjasto korostaa myös individuaalisuutta, kun taas maorien kulttuuri on kollektivistinen. Heidän tapansa välittää tietoa on edelleenkin kasvatusten tapahtuvaa, eikä länsimainen kirjasto tue sitä. He eivät myöskään hyväksy tiedon avointa saatavuutta Internetissä. Heidän menneisyyteensä liittyvät tiedot eivät saa olla muiden nähtävillä.

Duncker ehdottaakin kulttuurillisesti relevanttien digitaalisten kirjastojen toiminnallisuuteen muutamia muutoksia, jotka ottaisivat eri kulttuuritaustaiset käyttäjät huomioon. Tämä voisi näkyä tiedon saannin rajoituksissa ja tiedon järjestämisessä paikallisen kulttuurin mukaisesti. Jos länsimainen tietohierarkia ei ole tuttu, sen tuntemista ei pitäisi edellyttää etsintätyökalujen toiminnoissa.

Eri kulttuureilla saattaa myös olla erilaisia tapoja yhdistää asioita. Marcus [2003] mainitsee erot amerikkalaisen ja kiinalaisen katsomistavan välillä. Pyydettyäessä kuvailemaan, mitä akvaariossa näkee, kiinalaiset katsojat

kuvailivat kasvien, kivien ja kalojen välisiä yhteyksiä. He keskittyvät suhteisiin, kontekstiin ja kokonaisuuksien välisiin riippuvuussuhteisiin. Amerikkalaiset puolestaan kuvailivat yksittäisten kalojen ominaisuuksia kategorioiden, luokittelujen ja yhteisten ominaisuuksien kannalta.

Walton, Vukovic ja Marsden [2002] tutkivat Internetin navigointikäytännön kulttuurisidonnaisuutta. Heidän tehtävänä oli tutkia puudiagrammien ymmärrettävyyttä ei-länsimaisessa kulttuurissa. Heidän koehenkilöinä oli joukko eteläafrikkalaisia opiskelijoita. Koehenkilöt osoittivat totutusta länsimaisesta käytännöstä poikkeavaa asioiden yhdistelyä sukupuuta piirrettäessä. Heille sopivin muoto oli piirtää isän ja äidin puolen sukupuut erikseen. Eriäinen mentaalinen malli voi vaikeuttaa länsimaiseen hierarkiaan nojaavaa navigointia Internetissä. Varsinkin käyttäjän liikkeitä syvemmälle sivustoon kuvaavan navigointipolun ja kalan silmä-näkymän käyttö voi olla hankalaa.

2.3. Lokalisoinnin edut

Gouldin [2000] mukaan kansainvälisillä yrityksillä on kaksi vaihtoehtoa: joko yksi sivusto kaikille tai lokalisoidut sivut eri kulttuuriryhmille. Yksi sivusto voi tulla kysymykseen siinä tapauksessa, jos siinä toimii yhdistävänä tekijänä esimerkiksi ammatti, kuten astronomit. Hänen mukaansa tämä toiminta ei tule onnistumaan, jos yhdistävä tekijä puuttuu.

Sun [2001] on verrannut pintapuolista ja syvemmälle menevää lokalisointia ja niiden vaikutusta käyttäjiin. Tutkimuksessaan hän käytti Lotuksen ja Adoben Yhdysvaltojen, Saksan, Kiinan sekä Brasilian sivuja. Lotukselta oli lisäksi mukana vielä tanskalaiset sivut. Lotuksen sivujen suunnittelussa oli kiinnitetty huomiota kieleen, visuaalisiin elementteihin, väreihin ja sommitteluun. Adoben sivuilla lokalisointi oli tehty vain kielen tasolla. Tarkoituksena oli havainnoida käytettävyyden yhteyttä kulttuurillisiin elementteihin.

Koehenkilöt tutustuivat Adoben kotimaansa sivuihin sekä yrityksen Yhdysvaltojen sivuun. Lotuksen tapauksessa he tutkivat edellä mainittujen lisäksi vielä yhtä sivustoa. Tärkeimmäksi elementiksi saksalainen koehenkilö mainitsi hierarkian. Kiinalainen ja brasilialainen arvostivat visuaalisuutta ja värejä. Taulukossa 3 esitetään Lotuksen sivujen kulttuurilliset elementit, jotka ilmenivät Sunin tutkimuksesta.

Taulukko 3. Lotuksen sivujen kulttuurilliset elementit

	Valikko	Värit	Visuaaliset elementit	Kuvat	Sommittelu
Yhdysvallat	tärkein ensin	kirkkaita keltaisia	pyöreitä muotoja	erirotuisia ihmisiä	paljon tyhjää tilaa
Saksa	aakkosjärjestys	valkoista, mustaa	kulmikkuutta	kuuluisa saksalainen rakennus	vähän tyhjää tilaa
Kiina				lootuskukka	
Brasilia		kirkkaita		sokeritoppa-vuori	

Elementit on valittu kullekin sivulle eri kulttuurien mieltymysten mukaan. Tutkimuksesta ilmenee, että lokalisatiosta on hyötyä sivuilla. Koehenkilöt olivat tyytyväisiä oman kulttuurinsa elementtien hyväksi käyttämiseen suunnittelussa. Adoben sivuja selaillessaan he eivät aluksi edes huomanneet olevansa oman maansa sivuilla. Saksalainen koehenkilö kommentoikin, etteivät yhdysvaltalaiset sivut ja saksankielinen teksti sovi yhteen, aivan kuin sivun osat olisi suunnitellut eri ihminen.

Koehenkilöitä tutkimuksessa oli vain kolme, joten tuloksia ei voi yleistää liikaa. Tuloksista voidaan kuitenkin tehdä johtopäätöksiä siitä, että pelkkä kielen kääntäminen ei riitä lokalisoinnissa. Suunnittelijoiden on käytettävä useita elementtejä. Kulttuurillisten elementtien käyttö helpottaa navigointia ja saa käyttäjän tuntemaan olonsa ”kotoisaksi”. Ei saa unohtaa, että sisältö on etusijalla. Sivujen olisi sisällettävä paikalliselle käyttäjälle tärkeää tietoa.

Myöskään kaikki kulttuurilliset elementit eivät ole sopivia kaikille sivuille. Suunnittelijan on otettava kohdeyleisönsä huomioon ja pohdittava tarkoituksen mukaisia elementtejä. Valinta riippuu siitä, millaisen viestin haluaa käyttäjälle välittää. Esimerkiksi viihdesivuilla ja yrityksen kotisivuilla on erilainen funktio. Lisäksi stereotyyppien käyttöä tulisi välttää. Suunnittelijan on helppo suortua niiden käyttöön, koska tieto kohdemaan kulttuurista ei ole riittävä. Myös kulttuurin sisäiset vähemmistöt jäävät usein huomiotta.

Nortonin [2002] tekemän tutkimuksen mukaan käytettävyyks ei välttämättä parane, jos käyttöliittymässä on lokalisoituja elementtejä. Hänen mukaansa niiden ymmärrettävyys ei ollut sen parempi kuin Windowsin omien elementtienkään. Nortonin tutkimukseen osallistuneista Botswanalaisista 57% piti käyttöliittymän lokalisoituna tarpeellisenä, mutta vain 25% halusi siihen paikallisen kielen. Suurimmalla osalla koehenkilöistä ei ollut ongelmia tulkita käyttöliittymää kulttuurista johtuvien erojen takia. Lisäksi kolmasosa heistä

toivoi lokalisointia erilaisten kulttuurien erojen tunnustamiseksi ja entistä suuremman väestön osan tekniikan käyttömahdollisuuden takaamiseksi.

3. Yhteiskunnalliset seikat

3.1. Hofsteden teoria

Seuraavissa luvuissa aion esitellä Hofsteden teorian ja siihen liittyvät käsitteet. Tutkimieni artikkelien perusteella Hofsteden teoria kulttuureja erottavista tekijöistä on yksi tärkeimmistä. Yhteiskunnallisia eroja tutkii myös toinen, Trompenaarsin teoria. Käsittelen sitä lyhyemmin, koska en ole katsonut sitä yhtä tärkeäksi teoriaksi. Tämä johtuu siitä, että sitä on käytetty artikkeleissa vähemmän. Seuraavat käsitteiden kuvaukset on otettu Marcusin ja Gouldin [2000] artikkelista. He ovat myös tutkineet 53 maata Hofsteden teorian perusteella. Mainitsen kappaleissa esimerkkejä muutamista maista kyseisen ominaisuuden osalta.

3.1.1. Vallan jakautuminen (power distance)

Vallan jakautuminen tarkoittaa sitä, miten vähän valtaa omaavat ihmiset suhtautuvat vallan jakautumiseen. Maissa, joissa valta jakautuu epätasaisesti, poliittinen valta on keskittynyt pienelle ryhmälle, organisaatorakenne on erittäin hierarkkinen sekä erot tuloissa ja asemassa ovat suuret. Auktoriteetteja kuuluu arvostaa ja totella.

Maissa, joissa valta jakautuu tasaisemmin hierarkiatasot ovat matalampia ja esimiesten ja alaisten suhteet läheisempiä. Vanhemmat ja lapset sekä opettajat ja oppilaat ovat tasa-arvoisemmassa asemassa. Tasa-arvoisuus on haluttu ja odotettu ominaisuus.

Marcusin ja Gouldin tutkimuksen mukaan kuusi epätasaisimman vallan jakautumisen maata ovat Malesia, Guatemala, Panama, Filippiinit, Meksiko ja Venezuela. Kaikkein tasaisimmin valta on jaettu Itävallassa, Israelissa, Tanskassa, Uudessa Seelannissa ja Irlannissa. Yhdysvallat on sijalla 38 ja Suomi 46 (53 merkitsee tasaisimmin jakautunutta valtaa).

Käyttöliittymäsuunnitteluun tämä vaikuttaa taulukon 4 mukaisesti:

Taulukko 4. Vallanjakautuminen käyttöliittymäsuunnittelun kannalta

Vallan jakautuminen	
Tasainen	Epätasainen
Pääsy käsiksi tietoon rajoittamaton	Pääsy käsiksi tietoon rajoitettu
Sosiaalisen ja moraalisen järjestyksen korostaminen symbolien muodossa ei-tärkeää	Sosiaalisen ja moraalisen järjestyksen korostaminen symbolien muodossa tärkeää
Keskittyminen auktoriteetteihin, ekspertteihin, virallisiin logoihin ja leimoihin vähäistä	Keskittyminen auktoriteetteihin, ekspertteihin, virallisiin logoihin ja leimoihin suurta
Etusijan antaminen kansalaisille, asiakkaille ja työntekijöille	Etusijan antaminen johtajille
Rajoitteiden asettaminen epätavallista	Rajoitteiden asettaminen tavallista
Sosiaalisten roolien perusteella tapahtuva tiedon organisointi epätavallista	Sosiaalisten roolien perusteella tapahtuva tiedon organisointi tavallista

Epätasaisen vallanjakautumisen maissa tulee kohdistaa huomio yritysten tai yhteisöjen logoon. Kuvamateriaali käsittää yleensä kuvia esimerkiksi yliopistorakennuksesta tai yrityksen päämajasta. Sivuilla keskitytään antamaan tietoa yrityksen historiasta, toiminnasta ja henkilöstöstä. Käyttäjän toimintojen huomioonottaminen ei ole tärkeää. Sen sijaan on hyvä keskittyä tarjoamaan tietoa yrityksen johdosta ja täten luoda luotettava kuva siitä. Siten sivuilla kävijä voi lähteä luomaan kestävästä suhdetta yritykseen.

Gould [2000] huomasi tämän tutkiessaan yhdysvaltalaisen ja malesialaisen sivujen eroja. Hän vertaili mm. kirjakauppojen sivuja ja havaitsi, että malesialainen kirjakauppa keskittyi organisaatiorakenteen kuvaamiseen. Kirjoja sivuilta ei voinut tilata, kokoelma oli pieni ja kirjoista oli hyvin vähän tietoa tarjolla. Yhdysvaltalainen kirjakaupan sivu puolestaan keskittyi enemmän käyttäjiin. Heille annettiin mahdollisuus laajojen valikoimien tutkimisen ja kirjojen ostamisen lisäksi mm. keskustella muiden käyttäjien kanssa ja lukea kirjailijahaastatteluja.

3.1.2. Kollektivismi/individualismi (collectivism vs. individualism)

Individualismi-ominaisuus kulttuurissa merkitsee löyhiä suhteita ihmisten kesken. Heidän kuuluu pitää huolta itsestään ja lähisukulaisistaan eikä kenestäkään muusta. Tällaisissa kulttuureissa arvostetaan omaa aikaa, vapautta ja haasteita. Kotioloissa arvostetaan totuutta ja rehellisyyttä sekä asioista puhumista. Materiaaliset palkinnot motivoivat työn tekoon. Individualistisessa

yhteiskunnassa arvostetaan yksilön oikeuksia, mielipiteenvapautta, äänestysoikeutta, lehdistön sananvapautta sekä itsensä toteuttamista ja valtaa päättää itse omista asioistaan.

Kollektivismi puolestaan tarkoittaa sitä, että ihmiset liittyvät jo syntyessään suuremman ryhmän jäseniksi. Tämä ryhmä suojelee heitä ja vaatii vastapalvelukseksi ehdotonta lojaaliutta. Työpaikalla arvostetaan koulutusta tehtävään, taitoja ja työn osaamisesta koituvaa palkitsemista korkeammalla asemalla. Perhepiirissä arvostetaan harmoniaa ja hiljaisuutta. Yhteiskunnassa yhteisön intressit ylittävät yksilön tarpeet. Ideologisella tasolla toteutetaan ajatusta harmonisesta, yksimielisestä ja tasa-arvoisesta yhteiskunnasta.

Käyttöliittymäsuunnittelussa tämä näkökohta tulee ottaa huomioon taulukon 5 mukaisissa seikoissa:

Taulukko 5. Individualismi ja kollektivismi käyttöliittymäsuunnittelussa

Individualismi	Kollektivismi
Henkilökohtaisiin saavutuksiin perustuva motivaatio suuri	Henkilökohtaisiin saavutuksiin perustuva motivaatio vähäinen
Menestys ilmenee materialismina ja kulutuksena	Menestys ilmenee sosiaalis-poliittisten tavoitteiden saavuttamisena
Haastava puhetyyli ja äärimmäisten mielipiteiden sieto tai niihin rohkaiseminen	Viralliset iskulauseet ja mielipiteiden eroavaisuuksien hillitseminen,
Etusijalla nuoret ja toiminta	Etusijalla vanhat, paljon kokeneet, viisaat johtajat
Yksilöt	Tuotteet
Korostetaan totuutta	Korostetaan ihmissuhteita
Muutoksen korostaminen: uutuudet ja ainutlaatuisuus	Muutoksen korostaminen: perinteet ja historia
Halukkuus henkilökohtaisten tietojen antamiseen	Suojellaan henkilökohtaisia tietoja, joiden avulla yksilön voisi erottaa ryhmästä.

Kaikkein individualistisimpia maita olivat Yhdysvallat, Australia, Iso-Britannia, Alankomaat ja Kanada. Kollektivistisimpia taas olivat Guatemala, Ecuador, Panama, Venezuela ja Kolumbia. Suomi on sijalla 17, missä 1 merkitsee kaikkein individualistisinta maata.

Tämä ominaisuus on huomioitava toiminnallisuuden suunnittelussa. Marcus ja Gould [2000] tarkastelivat costaricalaista ja yhdysvaltalaisista luonnonpuiston kotisivua. Costaricalainen sivu keskittyy kuvaamaan luontoa kävijän kustannuksella. Sivuilla on myös suuressa roolissa kansallinen

iskulause "no artificial ingredients". Kun taas yhdysvaltalainen sivu keskittyy käyttäjään ja hänen toimiinsa sivulla sekä puistossa.

Gould [2000] kiinnitti huomiota erilaisten suhteiden luomiseen malesialaisen ja yhdysvaltalaisen yliopiston sivuja tutkiessaan. Malesialainen yliopisto keskittyy koko elämän kestäväan suhteeseen, joka ei pääty opiskelijan valmistuttua. Sen vuoksi sivut luovatkin kuvaa organisaatiosta, jolla on korkeat tavoitteet, joista opiskelija voi olla ylpeä. Yhdysvaltalainen sivusto puolestaan keskittyy organisaation ja opiskelijan tasa-arvoiseen suhteeseen, josta kumpikin osapuoli hyötyy.

3.1.3. Feminiinisyyden/maskuliinisuus (femininity vs. masculinity)

Tämä aspekti kuvaa sukupuolirooleja, jotka ovat jakautuneet maskuliinisiin ja feminiinisiin. Maskuliinisuuteen liittyy kilpailu ja kovuus, kun taas feminiinisuuden herkkyys, koti, lapset ja ihmisläheisyys. Maskuliinisissa kulttuureissa perinteisiä eroja pidetään yllä kun erojen pienentyminen ja roolien päällekkäisyys on ominaista feminiiniselle kulttuurille. Ensiksi mainitussa kulttuurissa työssä tavoitellaan tuloja, asemaa, etenemistä ja haasteita. Jälkimmäinen keskittyy hyviin suhteisiin esimiesten, kollegoiden ja alaisten kanssa, hyvään työympäristöön sekä turvattuun työpaikkaan.

Vahvasti maskuliinisia maita ovat Japani, Itävalta, Venezuela, Italia ja Sveitsi. Feminiinisiä puolestaan ovat Ruotsi, Norja, Alankomaat, Tanska, Jugoslavia, Costa Rica ja Suomi. Yhdysvallat on 15:n maskuliinisimman maan joukossa.

Marcus ja Gould suosittelevat käytettäväksi taulukossa 4 esitetyjä elementtejä:

Taulukko 6. Feminiinisyyden/maskuliinisuus-ominaisuus käyttöliittymissä (* oma lisäys)

Maskuliininen	Feminiininen
Perinteiset perhe-, sukupuoli- ja ikäroolit	Sukupuoliroolien hämärtyminen
Työtehtävät, roolit ja asiantuntius, nopeat tulokset rajatusta määrästä tehtäviä	Yhteistyö, tiedonvaihto ja tukeminen
Navigaatio suunnattu tutkimiseen ja kontrolliin	-
Huomio kiinnitetään pelien ja kilpailujen avulla	Huomio kiinnitetään runoudella, visuaalisella esteettisyydellä ja vetoamalla yhdistäviin arvoihin
Grafiikkoja, ääntä ja animaatioita käytetään tarkoituksenmukaisesti	Grafiikkoja, ääntä ja animaatioita käytetään esteettisiin tarkoituksiin*

3.1.4. Epävarmuuden välttäminen (uncertainty avoidance)

Epävarmuuden välttäminen liittyy siihen, kuinka paljon ihmiset yrittävät välttää epävarmuustekijöitä. Kulttuureissa on erilaisia vaatimuksia virallisuuden, täsmällisyyden ja laillis-sosiaalis-uskonnollisten seikkojen suhteen. Kun kulttuuri on suuntautunut välttämään epävarmuustekijöitä on yleistä, että alkoholismien esiintyminen, itsemurhien, onnettomuuksien ja vankien määrä ovat suuret. Yrityselämässä on enemmän sääntöjä, vaaditaan pidempää sitoutumista uralle ja keskitytään taktiikkaan strategiaa enemmän. Kulttuurit ovat ilmaisevia; ihmiset puhuvat käsillään sekä kovalla äänellä ja näyttävät tunteensa. He vaikuttavat aktiivisilta, tunteikkailta ja jopa aggressiivisilta. He arvostavat hierarkkisuutta instituutioissa ja ihmissuhteissa, jotta he pystyvät paremmin tulkitsemaan tilanteita. Erilaisuus koetaan uhaksi.

Kulttuureissa, joissa epävarmuustekijöitä siedetään paremmin on yleistä juoda enemmän kahvia, syödä vähemmän kaloreja sisältävää ruokaa, myös sydänsairaudet ovat yleisempiä. Liiketoiminta on epävirallisempaa ja tulevaisuuteen suuntautunutta. Ihmiset käyttäytyvät hillitysti ilmaisematta tunteitaan selvästi ja vaikuttavat rennoilta. Erilaisuus koetaan mielenkiintoiseksi tai jopa huvittavaksi.

Korkean epävarmuuden välttämisen arvon maita ovat Kreikka, Portugali, Guatemala, Uruguay ja Belgia. Asteikon toisesta päästä löytyvät Singapore, Jamaika, Tanska, Ruotsi ja Hongkong. Suomi on asteikon puolenvälin tienoilla ja Yhdysvallat kymmenen matalimman arvon joukossa.

Taulukko 7. Epävarmuuden välttämismuutokset suunnittelun kannalta

Epävarmuutta vältetään	Epävarmuutta siedetään
Yksinkertaisuus, selkeät vertauskuvat, rajattu määrä valintoja ja tietoa	Kompleksisuus, paljon valinnanvaraa ja sisältöä
Toimintojen tulosten ennakointi	”Vaeltelun” ja riskinoton hyväksyminen ja niihin rohkaisu
Navigaatiolla estetään eksyminen	Navigaation kontrolli vähäistä, esim. linkit avaavat uuden ikkunan
Vähennetään virhetoimintojen määrää mentaalisten mallien avulla	Konseptin ymmärtäminen mentaalisten mallien avulla
Epämääräisyyden välttäminen mm. värien, äänien ja typografian avulla	Värejä, typografiaa ja ääniä käytetään tiedon maksimointiin

3.1.5. Pitkä-/lyhytjännitteisyys (long- vs. short-term orientation)

Pitkäjännitteisyys on tärkeässä asemassa Aasian maissa, joissa Kungfutselaisuus on vaikuttanut tuhansia vuosia. Seuraavat uskomukset ovat yleisiä:

- Vakaa yhteiskunta vaatii epätasa-arvoiset suhteet.
- Perhe on sosiaalisen organisaation prototyyppi, vanhemmilla on enemmän valtaa kuin nuorilla ja miehillä enemmän kuin naisilla.
- Hyveellinen käytös on sitä, ettei tee toisille niin, kuin ei haluaisi itseäänkään kohdeltavan.
- Hyveellinen käytös työelämässä käsittää taitojen ja koulutuksen hankkimisen, kovan työnteon sekä säästeliäisyyden, kärsivällisyyden ja periksiantamattomuuden.

Käyttöliittymäsuunnittelussa kannattaa ottaa huomioon seuraavat seikat:

Taulukko 8. Pitkä- ja lyhytjännitteisyys käyttöliittymäsuunnittelun kannalta

Pitkäjännitteisyys	Lyhytjännitteisyys
Sisältö keskittyy käytäntöön ja käytännön hyötyyn	Sisältö keskittyy totuuteen ja uskomusten varmuuteen
Suhteet tiedon ja uskottavuuden lähteinä	Säännöt tiedon ja uskottavuuden lähteinä
Kärsivällisyys tavoitteiden saavuttamisessa	Halu saada tuloksia heti ja saavuttaa tavoitteet nopeasti

Länsimaissa on yleistä korostaa individualismia, tasa-arvoa sekä täyttymyksen löytämistä luovuudesta ja itsensä toteuttamisesta. Aasialaisissa maissa keskitytään hyveelliseen käytökseen kun taas länsimaissa keskitytään uskoon ja totuuden etsimiseen.

Tämä aspekti on lisätty Hofseden teoriaan jälkikäteen. Siksi tutkimustuloksia ei ole saatavilla kaikista 53:stä maasta. Hofstede tutki uudelleen kaikkiaan 23 maata, joista pitkäjännitteisimpiä ovat Kiina, Hongkong, Taiwan, Japani ja Etelä-Korea. Lyhytjännitteisimpiä puolestaan ovat Pakistan, Nigeria, Filippiinit, Kanada ja Zimbabwe.

Käyttöliittymien suunnittelun kannalta tällä on vaikutusta Marcusin ja Gouldin mukaan navigointiin. Lyhytjännitteisissä maissa navigaation täytyy olla selkeää ja tavoitteet nopeasti saavutettavissa. Pitkäjännitteisissä maissa tavoitteisiin pääseminen voi vaatia enemmän kärsivällisyyttä.

3.2. Trompenaarsin teoria

Kulttuurillisten ulottuvuuksien määrittelyyn on käytetty yleisesti Hofsteden teoriaa, mutta Gould [2000] esittelee Trompenaarsin vastaavaa teoriaa. Siinä käytetään seitsemää ulottuvuutta Hofsteden viiden sijaan. Ne ovat:

- Yleisyys/yksityiskohtaisuus (universalism/particularism), onko kulttuuri suuntautunut sääntöihin vai suhteisiin.
- Individualismi/kommunitarismi (individualism/communitarism) vastaa Hofsteden individualismi/kollektivismi ulottuvuutta.
- Erikoistuneet/hajautuneet suhteet (specific/diffuse relationships), vaatiiko liike-elämä tarkat säännöt vai hyvät suhteet yhteistyön aloittamiseen.
- Neutraali/tunteikas kommunikaatiotyylit (neutral/affective communication styles) näytetäänkö tunteita vai ei.
- Saavutukset/huomion osoitukset (achievement/ascription), saavutaanko asema 'tekemällä' vai 'olemalla' tietyssä roolissa.
- Suhtautuminen aikaan: tapahtumat seuraavat toisiaan/menneisyys, nykyhetki ja tulevaisuus ovat sekoittuneet.
- Suhtautuminen luontoon: luonnon hallitsemiseen pyrkiminen, eläminen harmoniassa sen kanssa vai antaminen luonnon kontrolloida.

4. Yhteenveto

Kulttuurien erojen vaikutusta käytettävyyteen on tutkittu paljon. Tutkimuksissa on keskitytty varsinkin Internetsivujen käytettävyyteen. Etenkin verkossa toimivien kauppojen ja yritysten kannalta on tärkeää tunnistaa kulttuurierot, jotta entistä suurempi osa kulttuureista saadaan tavoitettua.

Elementit, kuten tekstit ja päivämäärät, ovat pintatason lokalisointiin liittyviä. Niiden avulla saa nopeasti muutettua käyttöliittymän kohdekulttuuria vastaavaksi. Se ei kuitenkaan riitä. On myös otettava huomioon kulttuurillisen tason elementit, kuten värit, symbolit ja sommittelu, joiden suhteen mieltymykset vaihtelevat kulttuureittain. Joidenkin symbolien tapauksessa on oltava todella tarkkana, ettei tule käyttäneeksi jopa loukkaavaa kuvakieltä. Yleisesti länsimainen sommittelu on vasempaan reunaan rajattua ja aasialainen keskitettyä. Elementtien käytön suhteen suurten kulttuuriryhmien sisälläkin on mieltymyseroja. Saksalaiset suosivat aakkosten mukaan järjestettyä navigointipalkkia, kun taas muualla linkkien tärkeysjärjestys on suositumpi vaihtoehto. Aasiassa kiinalaiset pitävät hillityistä väreistä ja sommittelusta. Hongkongilaiset ja japanilaiset puolestaan suosivat animoituja elementtejä ja kirkkaampia värejä.

Hosteden teoria tutkii kulttuurillisia elementtejä viiden ominaisuuden avulla. Nämä ominaisuudet kuvaavat kulttuureja syvällisesti ja niistä voi vetää tiettyjä johtopäätöksiä käyttöliittymäsuunnittelun avuksi. Hofstede jaottelee kulttuurit seuraavien ominaisuuksien mukaan: individualistisuus/kollektivistisuus, vallan jakautuminen, epävarmuuden välttäminen, maskuliinisuus/feminiinisyys ja pitkä-/lyhytjännitteisyys.

Käyttöliittymien lokalisoinnin tärkeydestä ei oltu yksimielisiä. Jotkin tutkimukset [Sun, 2001] totesivat sen parantavan käytettävyyttä. Toiset [Norton, 2002] taas huomasivat, ettei lokalisoinnilla ollut parantavaa vaikutusta käytettävyyteen. Mielestäni on kuitenkin olennaista tietää kohdemaan kulttuurista jotain, ennen kuin alkaa suunnittelemaan käyttöliittymää. Esimerkiksi maorien parissa tehty tutkimus [Duncker, 2002] kirjastometaforan toimivuudesta osoittaa, että kulttuurillinen tietämys on tarpeen. Mielestäni ainakin kielen kääntäminen on ehdotonta. Ulko-asuseikkojen, kuten värien, lokalisointi voi tulla kyseeseen, jos käyttötarkoitus sitä vaatii. Tällainen voisi olla esimerkiksi naisille suunnattu sivusto. Värimieltymyksien suhteen voi mielestäni vetää joitain johtopäätöksiä kulttuureittain, mutta varmasti ne vaihtelevat yksilöittäin. Niiden lokalisointi ei saa mennä tärkeämpien seikkojen edelle. Tällaisia ovat Hofsteden mainitsemat viisi ulottuvuutta. Ne kuvaavat hyvin eri kulttuureja ja ovat hyödyllisiä apuvälineitä lokalisoinnissa.

Lokalisoinnin arvoa ei ole vielä ymmärretty kaikissa yrityksissä. Nielsenin mukaan yleinen asenne on "kyllähän ne englantia ymmärtävät" [Nielsen, 1990]. Tämä on tosin vanha kommentti, asenteet ovat voineet muuttua. Yleisesti ollaan kuitenkin sitä mieltä, että kaikkia miellyttävä lokalisointi on lähes mahdotonta. Yksittäisen maan sisälläkin on suuria eroja kulttuureissa, alakulttuureista puhumattakaan. Lokalisointi kuitenkin todistaa, että yritys on kiinnostunut asiakkaistaan ja haluaa ottaa heidän tarpeensa huomioon.

Viiteluettelo

- [Barber and Badre, 1998] Wendy Barber and Albert Badre, *Culturability: The Merging of Culture and Usability*. In: *Proceedings of the 4th conference on human factors and the web*. Basking Ridge, NJ.
- [Duncker, 2002] Elke Duncker, *Cross-cultural usability of the library metaphor*. In: *Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries*, 223-230.
- [Dysart, 2002] Joe Dysart, *Creating a truly international web site*. Online **26**, 1 (Jan/Feb 2002), 50-53.
- [Gould et al., 2000] Emilie W. Gould, Norhayati Zakaria and Shafiz Affendi Mohd. Yusof, *Applying culture to web site design: a comparison of*

- Malaysian and US web sites. In: *Proceedings of IEEE professional communication society international professional communication conference and Proceedings of the 18th annual ACM international conference on Computer documentation: technology & teamwork*, 161-171.
- [Hedberg and Brown., 2002] John G. Hedberg ja Ian Brown, Understanding cross-cultural meaning through visual media. *Education Media International* **39**, 1 (Mar 2002), 23-30.
- [Holzschlag, 2000] Molly Holzschlag, Integrated design. *WebTechniques* **5**, 54 (Sept. 2000) 40.
- [Marcus, 2003] Aaron Marcus, User-interface design and China: a great leap forward. *Interactions* **10**, 1 (Jan. 2003), 21-25.
- [Marcus and Gould, 2000] Aaron Marcus ja Emilie West Gould, Cultural dimensions and global web user-interface design. *Interactions* **7**, 4 (July 2000), 32-46.
- [Nielsen, 1990] Jakob Nielsen, Designing for international use. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, 291-294.
- [Norton, 2002] Diane Norton, Implementation of an electronic viewing application for multi-cultural users. In: *CHI'02 extended abstracts on Human factors in computer systems*, 534-535.
- [Russo and Boor, 1993] Patricia Russo and Stephen Boor, How fluent is your interface? Designing for international users. In: *Proceedings of the conference on Human factors in computing systems*, 342-347.
- [Simon, 2001] Steven John Simon, The impact of culture and gender on web sites: an empirical study. *ACM SIGMIS Database* **32** (1), 2001, 18-37.
- [Sukaviriya and Moran, 1990] Piyawadee Sukaviriya and Lucy Moran, *User interfaces for Asia*. In Jakob Nielsen (ed.) *Designing User Interfaces for International Use*. Elsevier, New York, 1990.
- [Sun, 2001] Huatong Sun, Building a culturally-competent corporate web site: an exploratory study of cultural markers in multilingual web design. In: *Proceedings of the 19th annual international conference on Computer documentation*, 95-102.
- [Walton *et al.*, 2002] Marion Walton, Vera Vukovic and Gary Marsden, 'Visual literacy' as challenge to the internationalization of interfaces: a study of South African student web users. In: *CHI '02 extended abstracts on Human factors in computer systems*, 530-531.

Suorakäyttöisyys kolmiulotteisessa veisto-ohjelmassa

Jyrki Parviainen

Tiivistelmä.

Tässä tutkimuksessa tutkitaan kolmiulotteista veisto-ohjelmaa ja sen suorakäyttöisiä työkaluja. Työkaluihin on myös ehdotettu muutamia 3D-maailmaa havainnollistavia apuvälineitä. Tutkimusta varten on rakennettu veisto-ohjelman prototyyppi, josta voi käytännössä nähdä, kuinka ratkaistut asiat toimivat.

Avainsanat ja -sanonnat: Veistäminen, suorakäyttöisyys, 3D.

CR-luokat: I.3.4, I.3.7, H.5.2

1. Johdanto

Veisto-ohjelmat eroavat perinteisistä 3D-mallinnusohjelmista käyttötavan ja tarkoituksensa perusteella. Veisto-ohjelmaa käyttäessä käyttäjän ei tarvitse harjoitella paljoakaan ohjelman käyttöä, eikä hänen tarvitse ymmärtää matematiikkaa tai 3D-grafiikkaa. Käyttäjä voi suoraan alkaa veistää luonnollisella tavalla ja katsoa, mitä tuotoksesta syntyy. Käyttäjä toimii kuin taiteilija, joka luo kappaleita veistämällä ja kaivertamalla. Lopputulos ei ole matemaattisesti tarkka vaan lähestulkoon samanlainen kuin veistäisi savimöykkyä.

Veisto-ohjelmia on tutkittu jo 1970-luvulta lähtien, jolloin ne olivat varsin rajoittuneita tietokoneiden laskentatehon takia. Nykyään on mahdollista tehdä monipuolisia työkaluja veisto-ohjelmiin sekä kehittää niihin monia uusia ideoita tietokoneiden tehokkuuden kasvettua.

Veisto-ohjelmat ovat suorakäyttöisiä [Shneiderman, 1982], jolloin niiden käyttö on nopeaa ja luonnollista. Suorakäyttöisessä ohjelmassa komennot suoritetaan fyysisinä toimintoina tai napinpainalluksina, ilman että käyttäjän tarvitsee hakea komentoja valikoista tai paleteista. Toiminnot ovat nopeita, toisiaan täydentäviä ja helposti peruutettavissa. Lisäksi suoritettujen toimintojen tulokset näytetään käyttäjälle reaaliaikaisesti.

3D-veisto-ohjelman työkalut ovat olennainen osa veisto-ohjelmaa, jolloin on tärkeää, että ne toimivat järkevästi ja niillä voidaan tehdä haluttuja asioita riittävän nopeasti ja helposti. Hyvät työkalut ovat myös tärkeitä sen takia, että ne määrittelevät sen, mitä ohjelmalla voidaan tehdä, ja kuinka joku toiminto saadaan tehtyä. Veisto-ohjelma voi olla täysin käyttökelvoton, jos sen työkalut

ovat vääriä tietynlaisiin tilanteisiin tai niillä ei voi tehdä haluttuja asioita, vaikka ohjelma muuten olisikin moitteeton.

Tässä tutkimuksessa luodaan suorakäyttöisiä työkaluja, jotka tukevat veisto-ohjelman toimintaa. Työkaluja ovat materiaalin lisäys- ja poistotyökalut pallo- ja laatikkomuodoilla. Työkaluihin on myös rakennettu 3D-maailman hahmottamista parantavia apuvälineitä. Tutkimuksessa keskitytään myös työkalujen ja kappaleen vuorovaikutteisuuden luonnollisuuteen ja käyttöön. Uusien työkalujen rakentamisen myötä nähdään, voidaanko veisto-ohjelman tiettyjä toimintoja tehdä halutulla tavalla.

Uusien työkalujen kehittäminen on myös merkittävä pohja jatkotutkimukselle, joka voi keskittyä uusien työkalujen luomiseen tai koko veisto-ohjelmaan kehittämiseen. Tutkimuksessa selviää, voidaanko 3D-veisto-ohjelmaan luoda uudenlaisia suorakäyttöisiä työkaluja, joilla kappaleiden muodostaminen onnistuu järkevästi, helposti ja todellista maailmaa kuvaavalla tavalla.

Seuraavassa luvussa käsitellään veisto-ohjelmiin liittyvää edeltävää työtä ja käyttötarkoituksia. Luvussa kolme käydään läpi veisto-ohjelmien ja niiden työkalujen yleisiä toimintaperiaatteita. Luvussa neljä käydään läpi tätä tutkimusta varten rakennetun veisto-ohjelman rakenne. Luvussa viisi esitellään tulokset ja sen jälkeen on yhteenveto tutkimuksesta.

2. Taustaa

Tässä luvussa käsitellään aiheeseen liittyvää edeltävää työtä ja veisto-ohjelmien käyttötarkoituksia. Edeltäviä töitä esitellään lyhyesti, jolloin niistä mainitaan vain merkittävimpiä ominaisuuksia. Luvun lopuksi on pohdittu, millaisiin käytännön tilanteisiin veisto-ohjelmat sopisivat.

2.1. Veisto-ohjelmien taustaa

Parent [1977] esitteli kolmiulotteisen veistämisen idean. Hänen veisto-ohjelmallaan pystyi muodostamaan uusia kappaleita leikkaamalla ja yhdistämällä niihin toisia kappaleita. Galyean ja Hughes [1991] kehittivät veisto-ohjelman, joka perustuu 3D-vokseliruudukkoon ja suorakäyttöisyyteen. Heidän ohjelmassaan kappaleen ajatellaan olevan ikään kuin savimöykky, jota voidaan veistää erilaisilla työkaluilla. He kehittivät 3D-vokselimallin perinteisten 2D-piirto-ohjelmien pohjalta. 2D-ohjelmissa käyttäjä muokkaa kaksiulotteisen ruudukon pikseleitä, jolloin 3D-veisto-ohjelman ratkaisu on vain tämän laajenus kolmiulotteiseksi.

Galyeanin ja Hughesin ohjelmassa työkalu on myös toteutettu 3D-vokselien avulla, jolloin työkalun ja muokattavan kappaleen leikkaus ja liitos on nopea laskea. 3D-vokselityökalu mahdollistaa myös hyvin erikoiset kappaleiden

muodot, mutta rajoittavat työkalujen kokojen muuttamista. Wang ja Kaufman [1995] kehittivät samanlaisen ohjelman, jossa oli lisäksi sahaustyökalu. Raviv ja Elber [2000] kehittivät veisto-ohjelman, joka mahdollistaa muuttuvan tarkkuuden vokseliruudukossa. Tällöin tiettyjä ruudukon osia jaetaan pienempiin aliruudukkoihin, jolloin kappaleeseen saadaan haluttuun kohtaan parempi tarkkuus. Tällainen ratkaisu on erittäin hyvä, koska kappaleissa saattaa olla joitakin osia, jotka vaativat suurempaa tarkkuutta kuin toiset osat. Esimerkiksi ihmisen kasvoja veistettäessä on selvää, että silmiä mallinnettaessa veistoalueen pitää olla huomattavasti tarkempi kuin toisaalta isoja pintoja sisältäviä poskea tai otsaa tehtäessä.

Perng *et al.* [2001] kehittivät samanlaisen järjestelmän kuin Wang ja Kaufman, jossa ohjaus oli hoidettu virtuaalisesti 3D-ohjaimella, jolloin työkalun liikuttaminen oli luonnollista ja vapaata. Perngin ratkaisussa oli myös esitetty paranneltu Marching cubes -algoritmi, jolla muodostetaan kappaleelle yhtenäinen pinta. Alkuperäisen Lorenzenin ja Clinen [1987] esittelemän Marching cubes -algoritmin ongelma oli reunojen epätasaisuus.

2.2. Veisto-ohjelmien käyttötarkoituksia

Veisto-ohjelmia voidaan hyödyntää monessa eri yhteydessä. Ohjelmilla voidaan luoda monenlaisia kappaleita. Esimerkiksi elokuvateollisuudessa, tietokonepeleissä kuin mainonnassakin voidaan käyttää hyväksi veisto-ohjelmilla tehtyjä 3D-kappaleita, kuten autoja, huonekaluja, hahmoja ja eläimiä. Ohjelmilla ei voi luoda matemaattisesti tarkkoja kappaleita, mutta niillä voidaan luoda kappaleiden hahmotelmia. Ohjelmat sopivat hyvin esimerkiksi taiteilijoiden ja graafikoiden avuksi.

Ohjelmia voidaan lisäksi käyttää erilaisten kappaleiden rakenteiden tutkimiseen, koska ohjelmilla voidaan kaapia kappaleesta kerroksia yksi kerrallaan pois. Tällaista kerroksien tutkimista voi esimerkiksi lääketiede hyödyntää, jolloin tietystä kolmiulotteisesta reaali maailman kappaleesta, kuten sisäelimestä tai aivoista, muodostettua 3D-kappaletta tutkitaan veistämällä siitä esiin aina vain sisempiä kerroksia.

2.3. Yhteenveto

Veisto-ohjelmia on tutkittu jo 1970-luvulta lähtien, jolloin niiden toteuttamisen tavat ovat muuttuneet ajan myötä. Ohjelmat ovat kehittyneet ja nykyään ne mahdollistavatkin reaaliaikaisen käyttöliittymän, jolloin veistäminen on luonnollista ja helppoa. Veisto-ohjelmia voidaan hyödyntää monenlaisissa eri yhteyksissä, kuten elokuvateollisuudessa ja tietokonepeleissä.

3. Ohjelman yleinen toiminta

Tässä luvussa käsitellään veisto-ohjelmien yleistä toimintaa ja niiden työkalujen ohjaamista. Ensimmäisessä aliluvussa kerrotaan veisto-ohjelmien yleisiä periaatteita ja niiden käyttötarkoituksia. Työkalun ohjaaminen -luku käsittää teorioita suoraikäyttöisyydestä, kaksikäyttöisyydestä ja työkalun ohjaamisesta 3D-maailmassa.

3.1. Veisto-ohjelma

Veisto-ohjelmassa käyttäjä muokkaa kappaletta, jonka voidaan ajatella olevan reaali maailmaa vastaava savimöykkyä tai puupala. Tätä kappaletta käyttäjä pystyy muovaamaan erilaisilla työkaluilla saadakseen halutun lopputuloksen. Ohjelman käytön on tarkoitus olla mahdollisimman samankaltaista oikean veistämisen kanssa. Kun ohjelma kuvaa realistista tilannetta, ohjelman käyttäminen on helppoa. Käyttäjillä on mielikuvia tosielämästä, jolloin ohjelman käyttö ei vaadi suurempaa ohjelman opettelua. Käyttäjä voi veistää omien mielikuviansa mukaansa, jolloin hänellä on jo mielessään arkielämästä syntynyt käsitys veistämisestä.

Veisto-ohjelman tekniikka mahdollistaa monikerroksisten ja -mutkaisten kappaleiden rakentamisen, mutta sillä ei ole mahdollista tehdä matemaattisesti tarkkoja kappaleita. Kappaleet, joita veisto-ohjelmalla pystytään tekemään, sopivat hyvin reaalielämän esineiden hahmottamiseen ja sellaisten kappaleiden luomiseen, joissa tarkkuus ei ole niin merkitsevä.

Veisto-ohjelmia voi tehdä monella eri tavalla, mutta tässä tutkimuksessa kiinnitetään huomiota vokselipohjaiseen tapaan. Vokselipohjaisessa tavassa kappaleen muokkaaminen tapahtuu muokkaamalla vokseliruudukkoa, jossa vokseleilla on kaksi tilaa: "päällä" ja "ei päällä". Vokselin tila määräytyy sen mukaan, kuuluuko kyseinen vokseli muokattavan kappaleen sisälle vai ulkopuolelle. Alussa kaikki vokselit ovat joko päällä tai yksikään vokseli ei ole päällä. Veistäessä tutkitaan, mitä vokseleita työkalu koskettaa. Tämän perusteella vokselit, joita työkalu leikkaa, otetaan aina pois päältä. Samoin tapahtuu pursotuksessa, jolloin asia menee vain päinvastoin. Pursottaessa vokseleiden tila muutetaan jälleen niin, että ne ovat päällä. Vokselien tilojen pohjalta kappaleelle rakennetaan yhtenäinen pinta.

3.2. Ohjaaminen

Veisto-ohjelmissa työkalun ohjaaminen ja vuorovaikutus voidaan tehdä monella eri tavalla. Eräs tapa on ohjata työkalua ja kappaletta hiirellä. Ongelma tässä tavassa on se, että näyttö ja ohjain ovat kaksiulotteisia, jolloin ohjaaminen ei ole niin helppoa ja tarkoituksenmukaista. 2D-maailma ja -ohjain eivät voi luontevasti esittää kolmiulotteista kappaletta. 3D-maailman kannalta paras tapa ohja-

ta työkalua on virtuaaliohjain, kuten datakäsine tai 3D-ohjain. Tällöin työkalun käyttäminen on lähempänä arkitodellisuutta.

Kahden käden hyödyntäminen on myös mahdollista, jolloin toisella kädellä ohjataan työkalua ja toisella kappaletta. Tässä työssä on käytössä kaksi ohjainta, jolloin hallitsevalla kädellä ohjataan työkalua ja toisella kädellä pyöritetään kappaletta. Tällainen ohjaustapa tukee kaksikäsisyyden hyödyntämistä. Buxton ja Myers [1986] ovat todenneet, että kahden käden hyödyntäminen parantaa tuloksia varsinkin suorakäyttöisessä järjestelmissä ja järjestelmissä, joissa molempien käsien yhtäaikainen käyttö on jatkuvaa. Kaksi kättä voi parantaa tehokkuutta myös, vaikka käyttö olisi vuoroittaista. Tässä tutkimuksessa käsien käyttö on yhtäaikaista. Tosin hetkittäin käyttäjän ei tarvitse käyttää kuin ainoastaan kättä, jolla työkalua ohjataan mikäli kappale on käännetty ensin sopivaan asentoon.

Ohjaaminen on suorakäyttöistä ja reaaliaikaista, jolloin työkalun vaikutukset nähdään välittömästi, eikä komentoja tarvitse antaa hankalasti valikoiden kautta. Shneidermanin [1982, 1983] mukaan suorakäyttöisessä ohjelmassa komennot suoritetaan fyysisinä toimintoina tai napinpainalluksina ilman, että käyttäjän tarvitsee hakea komentoja valikoista tai paleteista. Lisäksi toimintojen tulee olla nopeita ja tulosten pitää näkyä käyttäjille välittömästi toiminnon tehtyä.

Työkalujen toimintaa voidaan kuvata luonnollisen maailman työkalujen metaforilla, jolloin niiden toimintaa voidaan ymmärtää ja kuvata paremmin. Carroll *et al.* [1988] mukaan ohjelmien ohjausta voidaan helpottaa, jos niitä suunnitellaan metaforien avulla. Käyttäjät pystyvät käyttämään ohjelmia reaali maailman perustietämystensä ja kokemustensa ansiosta. Tutkimuksessa käytetty työkalujen ohjaus ajatellaan olevan esimerkiksi veitsen pää, joka osuessaan kappaleeseen poistaa siitä osan.

3.3. Yhteenveto

Veisto-ohjelmia voidaan toteuttaa eri tavoilla, joista vokselipohjainen tapa on yleisin. Veisto-ohjelmien käyttö on suorakäyttöistä ja reaaliaikaista, jolloin niiden käyttäminen on helppoa. Lisäksi veisto-ohjelmissä voidaan hyödyntää kaksikäsisyyttä, jolloin ohjelman käyttämisestä tulee sujuvampaa.

4. Ohjelman rakenne

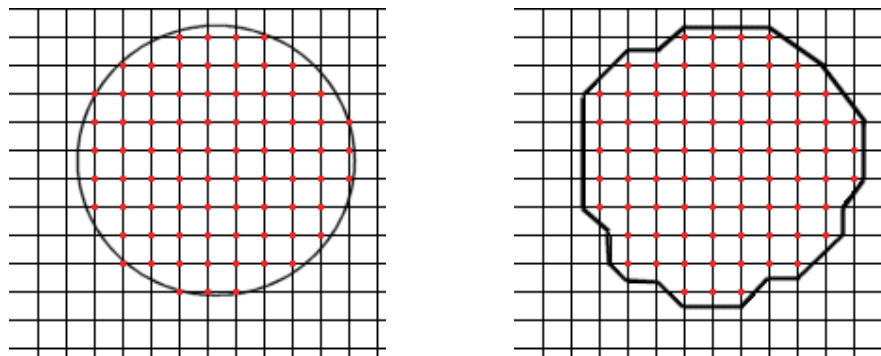
Tässä luvussa käsitellään ohjelman rakennetta ja toimintoja, joita se sisältää. Ohjelman rakenteesta ja toiminnoista kerrotaan niissä käytetyt algoritmit ja rakenteet pääsääntöisesti. Kappaleen yhtenäisen pinnan muodostamisessa apuna käytetty Marching cubes -algoritmi esitellään yleisellä tavalla.

4.1. Kappaleen esittäminen

Ennen kuin kappale voidaan piirtää ruudulle, tarvitsee sen vaatima kolmioverkko muodostaa työalueen 3D-ruudukosta. Tämä ruudukko koostuu $32 \times 32 \times 32$ kuutioista. Näitä kuutioita kutsutaan vokseleiksi. Vastaavasti 2D-ruudukon alkioita kutsutaan pikseleiksi. Työalue on rajattu alue, jolloin kappaletta muokattaessa rajan tultua vastaan kappale leikataan rajaa vasten. Bönning ja Müller [2002] ovat esitelleet ratkaisun, jossa alue olisi rajaamaton, mutta ainakin tällä hetkellä alueen rajoittaminen tietyn kokoiseksi ei haittaa veistämistä, kunhan alue on vain tarpeeksi suuri.

Kuutioverkon liitoskohdissa olevia pisteitä muokataan työkaluilla, jolloin määritellään, ovatko ne kappaleen sisällä vai kappaleen ulkopuolella. Näitä pisteitä voimme myös kutsua vokseleiksi. Tämän kuutioverkon pisteiden arvojen perusteella muodostetaan Marching cubes -algoritmillä kappaleelle yhtenäinen kolmiopinta. Verkko muodostetaan ulommaisten pisteiden ympärille, jotka vielä kuuluvat kappaleeseen.

Kuvassa 1 nähdään kuinka piirto tapahtuu vokseliruudukkoon kaksikulotteisessa tapauksessa. Ensimmäisessä kuvassa on merkattu pallomuodon sisäpuolelle jäävät vokselit. Toisessa kuvassa on hahmoteltu minkälainen pinta kyseisestä pallosta tulisi. Marching cubes -algoritmia parantamalla kuviosta saadaan enemmän tarkoitetun pallon muotoinen.

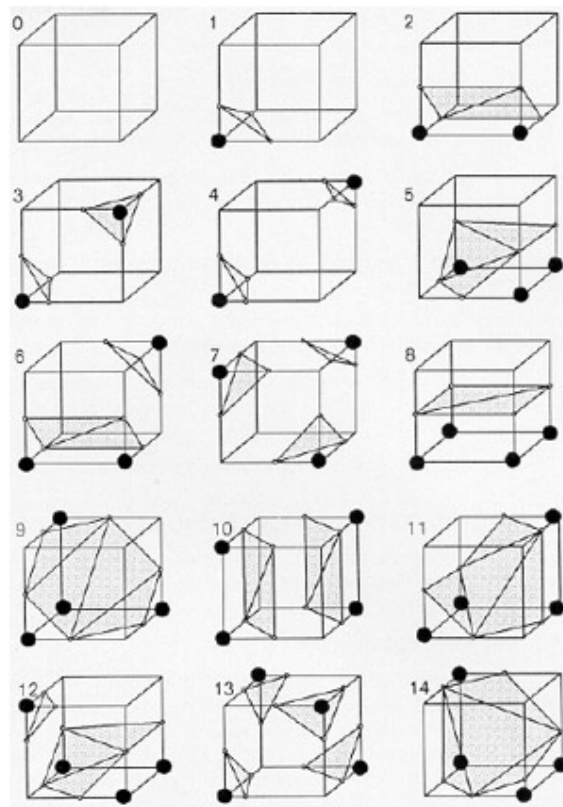


Kuva 1. Piirron tapahtuminen vokseliruudukkoon kaksikulotteisessa tapauksessa.

Ruudukon tarkkuus vaikuttaa suoraan siihen, miltä kappale näyttää ja kuinka yksityiskohtainen kappale voi olla. Mitä tiheämpi kuutioverkko on, sen paremmalta kappale näyttää. Työaluetta olisi mahdollista kasvattaa, mutta tällöin rakenteen vaatima muistimäärä ja laskentateho nousevat runsaasti. Muistinkulusta on mahdollista vähentää erilaisilla toimenpiteillä. Jos vokselien paikat lasketaan reaaliaikaisesti, niiden paikkojen tallennusta ei tarvita ja muistinkulutus vähenee jonkin verran. Rakenteen läpikäymistä voidaan nopeuttaa huomattavasti laittamalla 3D-ruudukko octree-tietorakenteeseen, jol-

loin ruudukosta päivitetään vain muokatut kohdat. Nykyinen ratkaisu käy koko 3D-ruudukon läpi, vaikka kappaletta muokattaisiin vain yhdestä kulmasta.

Kappaleen pinta muodostetaan Marching cubes -algoritmillä. Voimme määrittellä vokselikuution kahdeksalla pikselillä. Nämä pikselit sijaitsevat kuution kulmissa. Joidenkin pikseleiden arvon ollessa pienempi kuin ennaltamääritelty pinnanarvo ja toisaalta joidenkin muiden samalla suurempi, voimme päätellä, että kyseinen kuutio leikkaa kappaleen pintaa. Määrittelemällä, mitkä kuution sivuista leikkaavat pintaa, voimme muodostaa kolmioita, joista pinta muodostuu. Koska jokaisessa kuutiossa on kahdeksan pikseliä ja tiloja on kaksi mahdollista, on erilaisia yhdistelmiä 2^8 eli 256 kappaletta. Kuution tilojen symmetrisyyden ansiosta erilaisten tilojen määrä saadaan laskettua 15 kappaleeseen, jotka ovat nähtävissä kuvassa 2. Näiden tilojen perusteella tiedetään, minkälaisia kolmioita rakennetaan kuution sisälle. Algoritmissa kuljetaan jokaisen kuution läpi muodostaen tarvittavat kolmiot, jolloin kolmioverkko muodostuu. Kolmioverkon täytyy olla yhtenäinen, muutoin pintaan jää määrittelemättömiä kohtia. [Lorensen and Cline, 1987].



Kuva 2. Kuutioiden 15 eri yhdistelmää. [Lorensen and Cline, 1987].

Kun kappaleen pinta on muodostettu, kolmioille lasketaan tekstuuri-koordinaatit ja valaistusta varten pinnan normaalit. Kolmioiden normaalit las-

ketaan vokseleiden normaalien avulla laajentamalla ne kolmion sivuille. Vokseleiden normaalit ovat muodostettu vokseleiden sävyjen keskihajonnasta. Kolmioverkon muodostamisen jälkeen kolmioverkko piirretään ruudulle.

4.2. Työkalut

4.2.1. Työkalujen rakenne

Ohjelman työkalujen muodot ovat pallo ja kuutio. Muodot perustuvat kaavoihin, joten niiden ja vokselien leikkauskohta on helppo laskea. Joissakin veistohjelmissä, kuten Galyeanin ja Hughesin [1991] ratkaisussa, työkalut on tehty vokseleina samanlailla kuten kappalekin. Tällöin työkalujen muotojen on mahdollista olla minkälainen vokselikuviota tahansa, joka mahtuu määritellyn 3D-vokseliruudun eli työkalun muodon sisään. Työkalun ja vokselien leikkausta ei tarvitse laskea, koska tällöin tiedetään koko ajan, mitä vokseleita työkalu leikkaa. Työkalun liikkuminen tosin tapahtuu vokseliruudun mukaan, joten se on hieman rajoitettua. Tällaisen ratkaisun ongelma on työkalujen koon rajoittaminen ja muuttaminen.

Tutkimukseni ohjelmassa työkalut on muodostettu kaavoilla, jolloin niiden kokoa voidaan muuttaa keskipisteen suhteen helposti, jolloin esimerkiksi kuutiomuodosta saadaan helposti aikaiseksi erilaiset taso- ja puikkomuodot. Kaavoihin perustuvat työkalut toki ovat hieman muodoltaan rajoittuneempia kuin vokselityökalut, mutta koon muuttaminen on mielestäni tärkeää ja se tuo paljon uusia mahdollisuuksia ohjelman käyttöön. Valinta sopiikin mielestäni käyttötarkoitukseen hyvin.

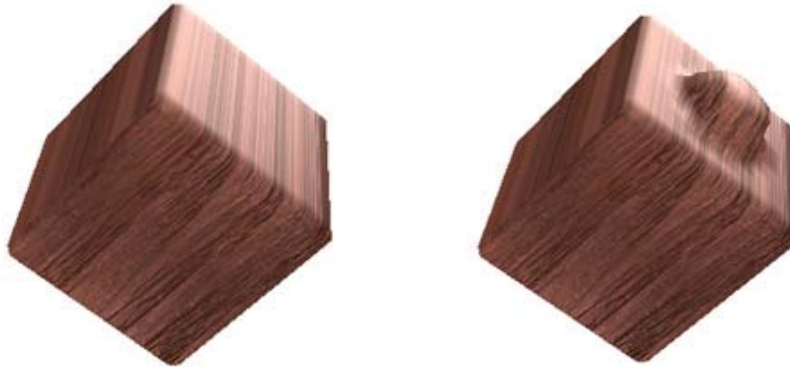
Ohjelmassa ei voi tällä hetkellä pyörittää työkaluja niiden omien akselidensa ympäri, jolloin vinojen pintojen muodostaminen olisi mahdollista yhdellä hiiren painalluksella. Tämä mahdollisuus tulee ohjelmaan jatkokehitysvaiheessa.

4.2.2. Pursotus

Kappaleen pursotustyökalu pursottaa kappaletta lisäämällä siihen valittuja muotoja. Tällä hetkellä valittavat muodot ovat pallo ja kuusisivuinen monikulmio. Työkalujen kokoja voidaan muuttaa ohjelman käytön aikana. Monikulmiosta saadaan koon muuttamisen avulla muodostettua esimerkiksi taso ja puikko.

Työkalun muodot ovat kaavoihin perustuvia muotoja, jolloin työkalun ja vokseleiden leikkauskohdat on nopea laskea. Pursotustyökalu muuttaa "pois päällä" -olevien vokseleiden arvoksi "päällä", jolloin ne piirretään lopulliseen kolmioverkkoon.

Kuvassa 3 on nähtävissä pursotustyökalujen käyttöä käytännössä. Kuution muotoiseen kappaleeseen on pursotettu pallon muotoinen uloke keskelle yhtä sivua.



Kuva 3. Kuutio, jossa on pursotettu pallo keskellä kuution yhtä sivua.

4.2.3. Kaiverrus

Kappaleen veistotyökalu kaivertaa kappaletta poistamalla siitä valittuja muotoja. Tällä hetkellä valittavat muodot ovat pallo ja kuusisivuinen monikulmio. Työkalujen kokoja voidaan muuttaa.

Työkalun muodot ovat kaavoihin perustuvia muotoja, jolloin työkalun ja vokselien leikkauskohdat on nopea laskea. Kaiverrustyökalu muuttaa "päällä" olevien vokselien arvoksi "pois päältä", jolloin niitä ei piirretä lopulliseen kolmioverkkoon.

Kuvassa 4 on nähtävissä kaiverrustyökalun käyttöä käytännössä. Kuutiosta on poistettu pallon muotoinen alue yhdestä kulmasta.



Kuva 4. Kuutio, josta on kaiverrettu pallolla.

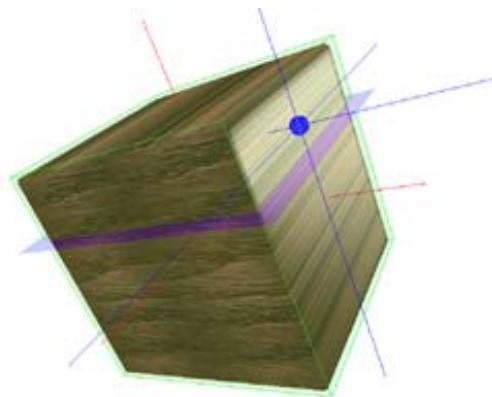
4.2.4. 3D-maailman hahmottamisen aputoiminnot

Ohjelmassa on apuvälineitä, joiden avulla 3D-maailman hahmottaminen paranee. Näitä ovat origokeskeinen koordinaatisto, työkaluun liitettävä koordi-

naatisto, rajauslaatikko ja rajauslaatikon taso. Käyttäjä voi halutessaan kytkeä toimintoja päälle tai pois. Hahmotusvälineiden kehitys on vielä kesken ja niihin on mietitty parempia toimintamuotoja, jotka tullaan toteuttamaan tulevaisuudessa.

Origokeskeinen koordinaatisto näyttää käyttäjälle, missä muokattavan alueen keskipiste sijaitsee, jolloin sen ympärille on helpompi rakentaa symmetrisiä kappaleita. Työkaluun liitettävä koordinaatisto liikkuu työkalun mukana. Tämä auttaa työkalun paikallistamista 3D-avaruudessa, jolloin käyttäjän on helpompi nähdä, missä työkalu sijaitsee suhteessa kappaleeseen. Koordinaatiston akselit leikkaavat työaluetta ja näin myös kappaletta, jolloin hahmottaminen helpottuu. Rajauslaatikko näyttää työalueen koon, jota voi myös käyttää kohdistukseen. Tällöin esimerkiksi voidaan työalueen kulmien suhteen sijoittaa kappaleen osia helposti. Rajauslaatikon taso on rajauslaatikon sisällä liikkuva taso, jonka voi jättää tietylle paikalle. Tätä tasoa voi käyttää apuna kappaleiden osien samalle tasolle tasaamiseen. Tasoa voi käyttää kolmessa eri suunnassa, jolloin käyttäjä voi valita, mitä akselia pitkin taso liikkuu.

Kuvassa 5 on nähtävissä 3D-maailman hahmottamisen apuvälineet käytössä. Kuvassa näkyvät origokeskeinen koordinaatisto, työkalun koordinaatisto, rajauslaatikko ja rajauslaatikon taso.



Kuva 5. 3D-maailman hahmottamisen aputoiminnot.

4.3. Ohjaus

Veisto-ohjelman ohjaaminen on toteutettu kahden käden ohjauksena, jolloin toisella kädellä kappaletta käännetään ja toisella liikutetaan työkalua. Samanlaista tapausta ovat tutkineet LeBlanc *et al.* [1991], jotka tulivat siihen tulokseen, että tämänlainen ohjaustapa on luonnollisempi kuin perinteiden CAD-ohjelmien ohjaus. Näissä ohjelmissa kappale esitetään monesta eri kulmasta samanaikaisesti eri ikkunoissa ja ohjaaminen tapahtuu yhdellä hiirellä ja korkeintaan tämän lisäksi näppäimistöllä. Lisäksi heidän tutkimusten

mu-kaan tämä ratkaisu nopeuttaa yleisiä toimintoja ja lisää käyttäjän ja ohjelman vuorovaikutusta.

Molemmat ohjauslaitteista sopivat kumpaankin käteen, joten käyttäjä voi valita, millä kädellä ohjaa mitäkin laitetta. Käytännössä hallitsevalla kädellä liikutetaan työkalua ja toisella kädellä pyöritetään kappaletta, koska työkalun liikuttaminen on vaikeampaa ja vaatii enemmän tarkkuutta. Kappaleen pyörittämiseen käytetään LogiCad Magellan Spacemouse laitetta [3Dconnexion, 2003], joka on 3D-ohjain ja mahdollistaa kuuden eri suunnan ohjaamisen. Kappaletta voisi myös liikuttaa piirtoalueella tällä ohjaimella, mutta liikuttaminen ei toistaiseksi ole kovinkaan tärkeää, vaan riittää, että kappaletta pyöritellään piirtoalueen origon ympärillä. Työkalua liikutetaan hiirellä, jossa on rulla. Rullalla liikutetaan työkalua syvyysuunnassa ja muita tasoja perinteisellä tavalla. XY-tasolla liikuttaessa työkalu liikkuu, kun hiiren oikea painike painetaan pohjaan. Hiiren kursori ja työkalu ovat tällä hetkellä kaksi eri asiaa. Oikean napin painamisella ohjelma tietää, että käyttäjän tarkoituksena on liikuttaa työkalua eikä hiiren kursoria, jolla valitaan ohjelman toiminnot. Tällainen ratkaisu on melko arveluttava eikä kaikkein luonnollisin, mutta kuitenkin yhdistelmä toimii eikä aiheuta käyttäjien kesken ylitsepääsemättömiä ongelmia [Sainio, 2004]. Jatkokehitysvaiheessa pyritään kehittämään toisenlainen ratkaisu kahden napin painalluksen tilalle.

3D-veisto-ohjelman isoimpia ongelmia on kolmiulotteisen maailman ymmärtäminen ja työkalun oikeanlainen liikuttaminen, koska näyttö ja työkalun ohjainlaitteet ovat vain kaksiulotteisia. Työkalun etäisyyksiä voi olla vaikea hahmottaa, jolloin muokkaaminen ei ehkä onnistu halutulla tavalla. Tätä varten on rakennettu muutamia hahmotusapuvälineitä, joita käyttäjä voi käyttää jos haluaa. Nielson ja Olsen [1986] ovat tutkineet 3D-kappaleiden ohjausta 2D-ohjaimilla ja kehitelleet kursoriin hahmottamista helpottavia välineitä, kuten kursorin mukana liikkuvan koordinaatiston. Samaa periaatetta käytetään hyväksi tutkimukseni hahmotusapuvälineissä.

Parempi ratkaisu kolmiulotteisten veisto-ohjelmien ohjaukseen olisi tuntopalautteeseen perustuva virtuaalinen ohjaus. Chen ja Sun [2002] ovat tutkineet veistämistä kuudensuunnanohjaimella, jossa on tuntopalaute. Tällöin käyttäjä voi ohjata työkalua luonnollisesti kolmiulotteisessa maailmassa ja tuntopalaute auttaa käyttäjiä tuntemaan kappaleen reunat, jolloin työskentelytapa on jo melko lähellä reaali maailman veistämistä.

4.4. Tiedon keräys

Ohjelmaan on rakennettu käyttäjän tietojen keräämis- ja tallennusjärjestelmä. Ohjelma tallentaa tiedostoon kaikki ohjelmassa tapahtuneiden tapahtumien tiedot. Näihin kuuluu työkalujen valinta, työmuotojen valinta, apuvälineiden

valinta ja kellonajat kaikista toiminnoista. Näitä tietoja käytetään hyödyksi ohjelman testikäytössä, jolloin testeistä saadaan enemmän täsmällistä tietoa esiin. Näiden tietojen perusteella ohjelmaa voi kehittää jatkossa paremmaksi.

Ohjelmassa voi myös tallentaa ja avata uudelleen muodostetut kappaleet. Kappaleita voidaan näin ollen muokata myöhemmin. Tiedostoon tallennetaan ainoastaan 3D-ruudukon vokseleiden tiedot.

4.5. Yhteenveto

Tutkimusta varten rakennettu veisto-ohjelma toimii vokselipohjaisesti, jossa kappaletta muokataan kaavoihin perustuvilla työkaluilla. Työkalujen muodot ovat pallo ja kuutio, joiden kokoja voidaan muuttaa. Kappaleen pinta muodostetaan vokseliruudukon pohjalta Marching cubes -algoritilla. Ohjelmassa voidaan myös tallentaa ja avata uudelleen muokatut kappaleet.

5. Tulokset

Tutkimuksessani tehty ohjelma toimii halutulla tavalla. Ohjelmalla pystyy luomaan kappaleita suoraikäyttöisesti veistämällä ja pursottamalla. Työkalut toimivat niin kuin oli tarkoitus ja kappaleen asennon ja työkalun paikan 3D-maailmassa ymmärtää helposti. 3D-maailman hahmottamisen aputoiminnot selvästikin auttavat ohjelman käyttöä tekemällä käytöstä nopeampaa. Ohjelmaa on testattu useilla käyttäjillä, joiden huomioiden perusteella joistakin ohjelman toiminnoista saatiin kehitettyä toimivampia kuin ne aiemmin olivat. Joitakin huomioita käytetään avuksi ohjelman jatkekehitysvaiheessa. Kuvassa 6 on nähtävissä joitakin ohjelmalla tehtyjä kappaleita.

Ohjelmaan jääneitä ongelmia ovat kappaleen huono tarkkuus ja työkalun aktivoimisessa käytetty hiiren oikea painike. Kappaleen tarkkuus johtuu 3D-ruudukon koosta ja Marching cubes -algoritmin toiminnasta. Tulevaisuudessa ohjelmaa optimoidaan, jolloin 3D-ruudukon kokoa voidaan suurentaa ja kappaleesta saadaan tarkempi. Lisäksi Marching cubes -algoritmia parannetaan, jolloin se muodostaa täydellisempiä ja pehmeämpiä pintoja. Tällöin työkalun jälki ja muokattavan kappaleen ulkonäkö on parempi. Hiiren oikean napin pohjassa pitämisen ongelmaan on tarkoitus keksiä toimivampi ratkaisu myöhemmissä kehitysvaiheissa.



Kuva 6. Ohjelmalla veistettyjä kappaleita: tuoli, malja ja karhunpää.

6. Lopuksi

Tämän tutkimuksen tarkoituksena on ollut selvittää kolmiulotteisen veisto-ohjelman ja sen työkalujen rakentamiseen liittyviä seikkoja ja tarkastella veisto-ohjelmien rakennetta ja toimintaa. Veisto-ohjelmat tarjoavat luonnollisen tavan rakentaa kappaleita verrattuna perinteisiin CAD-ohjelmiin ja ovat siksi hyvä alusta pohtia suorakäyttöisyyden hyödyntämistä 3D-ohjelmissa.

Tietokoneiden laskentatehon kasvaessa ja tekniikan kehittyessä veisto-ohjelmia voidaan kehittää paremmaksi niiden ominaisuuksien ja ohjaustavan mukaan. Jatkokehitys mahdollisuuksia ja uusia tutkimusaiheita on paljon ja niihin tullaan palaamaan lähitulevaisuudessa.

Lähteet

- [3Dconnexion, 2003] 3Dconnexion Logitech company, Spacemouse, <<http://www.3dconnexion.com/spacemouseplus.htm>>, 2003, Tarkistettu 13.12.2003.
- [Buxton and Myers, 1986] William Buxton and Brad A. Myers, A study in two-handed input. *Human Factors in Computer Systems, CHI '86 Conference Proceedings*, ACM Press, 1986, 321-326.
- [Bönning and Müller, 2002] Ralf Bönning and Heinrich Müller, Interactive sculpting and visualization of unbounded voxel volumes. *Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications*, ACM Press, 2002, 212-219.
- [Carroll *et al.*, 1988] John M. Carroll, Robert L. Mack, and Wendy A. Kellogg, Interface metaphors and user interface design. In Helander, M. (eds.), *Handbook of Human-Computer Interaction*. Elsevier Science Publishers, 1988, 283-307.

- [Chen and Sun, 2002] Hui Chen and Hanqiu Sun, Real-time haptic sculpting in virtual volume space. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, ACM Press, 2002, 81-88.
- [Galyean and Hughes, 1991] Tinsley A. Galyean and John F. Hughes, Sculpting: an interactive volumetric modeling technique. *Computer Graphics*, **25** (4), 1991, 267-274.
- [LeBlanc *et al.*, 1991] André LeBlanc, Prem Kalra, Nadia Magnenat Thalmann, and Daniel Thalmann, Sculpting with the "ball & mouse" metaphor. *Proceedings of the Graphics Interface '91*, 1991, 152-159.
- [Lorensen and Cline, 1987] William E. Lorensen and Harvey E. Cline, Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics*, **21** (4), 1987, 163-169.
- [Nielsen and Olsen, 1986] Gregory M. Nielsen and Dan R. Olsen Jr., Direct manipulation techniques for 3D objects using 2D locator devices. *Proceedings of the 1986 Workshop on Interactive 3D Graphics*, ACM Press, 1987, 175-182.
- [Parent, 1977] Richard E. Parent, A system for sculpting 3-D data. *Computer Graphics*, **11** (2), 1977, 138-147.
- [Perng *et al.*, 2001] Kuo-Luen Perng, Wei-The Wang, and Mary Flanagan, A real-time 3D virtual sculpting tool based on modified marching cubes. *International Conference on Artificial Reality and Tele-Existence (ICAT)*, 2001, 64-72.
- [Raviv and Elber, 2000] Alon Raviv and Gershon Elber, Three dimensional freedom sculpting via zero sets of scalar trivariate functions. *Computer-Aided Design*, **32** (8-9), 2000, 513-526.
- [Sainio, 2004] Nina Sainio, 3D-veisto-ohjelman työkalujen käytettävyys, Raportti B-2004-x, Tietojenkäsittelytieteiden laitos, Tampereen yliopisto, 2004.
- [Shneiderman, 1982] Ben Shneiderman, The future of interactive systems and the emergence of direct manipulation. *Behaviour and Information Technology* **1**, 1982, 237-256.
- [Shneiderman, 1983] Ben Shneiderman, Direct manipulation: A step beyond programming languages. *IEEE Computer*, **16** (8), 1983, 57-69.
- [Wang and Kaufman, 1995] S.W. Wang and A.E. Kaufman, Volume sculpting. *Symposium on Interactive 3D Graphics Proceedings*, ACM Press, 1995, 151-156.

Loppukäyttäjät osana tietojärjestelmäprojektia

Juha Pieviläinen

Tiivistelmä.

Tässä tutkimuksessa paneudutaan loppukäyttäjien rooleihin tietojärjestelmäprojekteissa. Aluksi esitellään aiheen tutkimuksen historiaa ja syitä miksi asia on tärkeä. Tämän jälkeen kuvataan erilaisia tapoja vaiheistaa tietojärjestelmäprojekti, jonka jälkeen kuvataan erilaisia metodeja käyttäjien huomioimiseksi. Lopuksi keskustellaan käyttäjien huomioimisen ongelmista ja yleisesti aihetta koskevista asioista.

Avainsanat ja -sanonnat: tietojärjestelmien suunnittelu, käyttäjälähtöinen suunnittelu, käyttöliittymä suunnittelu.

CR-luokat: H.5.2, K.6.1

1. Johdanto

Tietojärjestelmäprojektit ovat yrityksille suuria investointeja. Projektit sitovat henkilöstöä ja niihin uppoaa paljon rahaa ja henkilötyötunteja. Kaikki panostukset voivat olla kuitenkin turhia projektin epäonnistuessa, suuri osa projekteista onkin joko täysin epäonnistuneita tai osittain epäonnistuneita. Syitä projektin epäonnistumiseen voi olla monia, joissakin tapauksissa projektin lopputuloksena ei edes synny toimivaa järjestelmää, osassa järjestelmä toimii huonosti tai jää syystä tai toisesta vajaa käytölle. Syitä järjestelmän vajaa käyttöön on monia. Järjestelmän käyttäjät voivat kokea järjestelmän tehottomaksi verrattuna aikaisempaan järjestelmään, järjestelmä voi olla myös vaikea käyttää tai järjestelmä ei yksinkertaisesti täytä niitä vaatimuksia joiden mukaan se luotiin.

Tietojärjestelmiä tuottavien ja niitä hankkivien yritysten kannalta on hyvin tärkeää, että tietojärjestelmäprojektien onnistuminen olisi paljon nykyistä tilannetta todennäköisempää. Tietojärjestelmiä hankkivat yritykset haluavat pienentää projektiin liittyviä suuria taloudellisia riskejä ja toisaalta tietojärjestelmiä tuottavien yritysten on helpompi saada asiakkaita kun asiakkaan riskit pienenevät projektien onnistumisen todennäköisyyden parantuessa. Onnistumisen todennäköisyyttä saadaan nostettua etsimällä ja tunnistamalla eri ongelma kohtia tietojärjestelmäprojektin koko elinkaaren kattavassa prosessissa.

Tärkeimpänä seikkana tietojärjestelmäprojektissa on luoda järjestelmä, joka palvelee käyttäjiensä tarpeita ja soveltuu käytettäväksi heidän työssään. Jotta järjestelmä saataisiin vastaamaan käyttäjien tarpeita ja sopimaan osaksi heidän työtapojaan ja tehtäviään, on nämä tarpeet ja työtavat ja työtehtävät kartoitettava ja mallinnettava soveliaalla tavalla. Ulkopuolinen järjestelmän suunnittelija voi olla kokenut mallintaja, mutta hänellä ei ole kokemusta tietojärjestelmän kohdealueen työtehtävistä ja varsinkin niistä eri tehtävistä joita tietojärjestelmää hankkivassa yrityksessä tai yhteisössä tehdään.

Oman lisänsä käyttäjien tarpeiden, työtapojen ja työtehtävien kartoittamiseen tuo eri ihmisten työtapojen eroavaisuus. Tuskin löytyy kahta ihmistä jotka tekisivät saman työtehtävän täsmälleen samalla tavoin. Eroavaisuudet saattavat johtua käyttäjien yksilöllisten mieltymysten ja tottumusten lisäksi organisaation sisäisestä kulttuurista ja tietysti vallitsevasta yhteiskunnallisesta kulttuurista. Jotta käyttäjien tarpeiden, työtapojen ja työtehtävien mallintaminen olisi mahdollista tarpeeksi tarkalla ja yksityiskohtaisella tavalla vaaditaan käyttäjien itsensä osallistumista määrittely ja mallintamistehtävään. Käyttäjät eivät välttämättä toimi itse mallintajina vaan pelkästään tietolähteinä, tämä koska käyttäjillä ei ole yleensä ole tarpeellista tietotaitoa itse mallintamistehtävän suorittamiseen.

Tämän tutkimuksen punaisena lankana on käyttäjien roolien kuvaaminen ja eri kuvaamisen menetelmät tietojärjestelmäprojektien eri vaiheissa.

2. Taustaa

2.1. Historiallista taustaa

Tietojärjestelmien suunnittelumenetelmät ovat monimutkaistuneet runsaasti siirryttäessä tietojärjestelmien historian alkuhämäristä nykypäivään [Ruohonen ja Salmela, 1999, s. 12-14]. 1950- ja 1960-luvuilla tietojärjestelmien suunnittelu oli teknologialähtöistä. Teknologialähtöisestä suunnittelusta kehityskulku nykyiseen loppukäyttäjät huomioonottaviin suunnittelumenetelmiin alkoi johtotason käyttäjien huomioonottamisella. Davis kartoittaa artikkelissaan kattavasti suunnittelulähtökohtien kehitystä [Davis, 1985] alkuajoista 1980-luvulle.

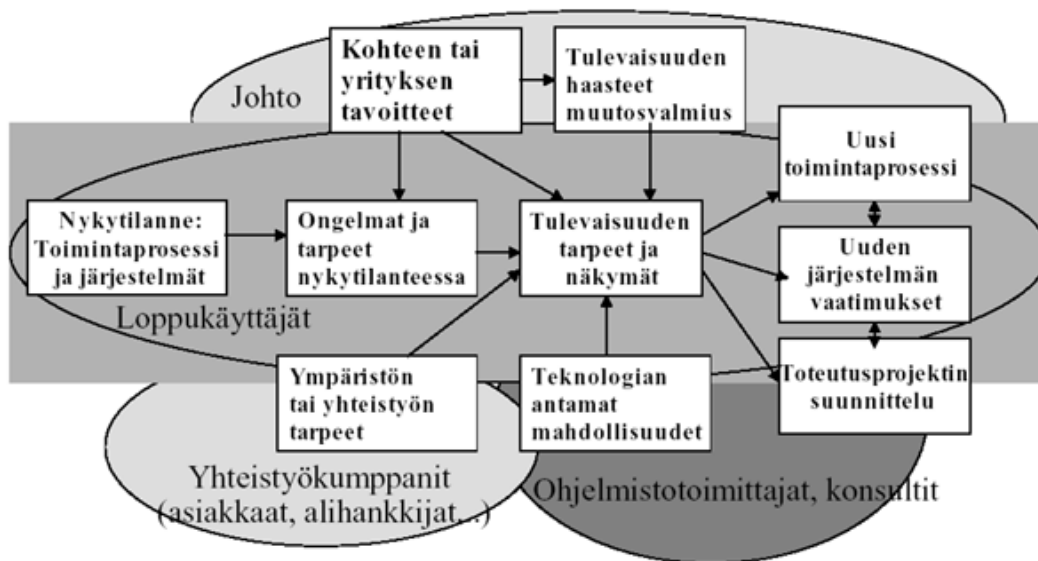
Kuten Davis [1985] artikkelissaan ilmaisee, että vaikka aikaisimmat maininnat käyttäjien huomioimisesta nousivat esille jo 1960-luvun keskivaiheilla, ei käyttäjien huomioimisesta puhuttu laajemmalti ennen kuin 1970-lukua. 1970-luvun näkemysten selkeänä heikkoutena oli se seikka, että niissä käyttäjiä pidettiin toimintatavoiltaan samankaltaisina.

Davisin [1985] tekemän kirjallisuuden historiakartoituksen mukaan 1980-luvulle siirryttäessä alettiin huomata aikaisemmin vallalla olleiden käyttäjien samankaltaisuusnäkömyksen virheellisyys. Käyttäjiä alettiin jakaa eri ryhmiin taitojensa ja työtehtäviensä mukaan. Kirjassaan Ruuhonen ja Salmela [1999, s. 12-14] toteavat nykyisin pyrittävän huomioimaan yksittäisen käyttäjän tarpeet ja ominaisuudet.

Tietojärjestelmien suunnittelumenetelmät ovat kehittyneet ajan kuluessa hyvin yksinkertaisesta teknologia- ja lähtökannasta yksittäisen käyttäjän tarpeita ja ominaisuuksia huomioivaan näkökantaan. Matkan varrelle on mahtunut monia erehdyksiä ja onnistumisia, mutta niin pitkään kuin huomattavan suuri osa tietojärjestelmäprojekteista ei saavuta tavoitteitaan [Aaltonen *et al.*, 2002], täysin tai ollenkaan, on suunnittelumenetelmissä kehittämisen varaa.

2.2. Käyttäjien tietomäärä

Käyttäjillä on hyvin paljon tietoja joihin tietojärjestelmänprojektin eri vaiheet perustuvat. Jotta projektin eri vaiheet voitaisiin toteuttaa onnistuneesti on järjestelmän toteuttajien saatava tietyt tiedot käyttöönsä ja suuri osa näistä tiedoista on toteutettavan järjestelmän loppukäyttäjien tiedossa.



Kuva 1. Loppukäyttäjien tietojärjestelmän toteutukseen liittyvät tiedot [Kettunen ja Simons, 2001, s.136]

Kuvassa 1. kuvataan tietojärjestelmäprojektin eri osapuolten omaavia tietotaito-resursseja tietojärjestelmään koskevissa eri aspekteissa. Kuten kuvasta 2. on helppo huomata on suurin osa tietotaidosta järjestelmän loppukäyttäjillä. Koska käyttäjät ovat varsinaisen työntekijöitä on heillä paras tuntemus

nykytilanteesta ja sitä koskevista ongelmista. Heillä on myös vaikutusta tulevaisuuden näkyviin ja tarpeisiin yhdessä johdon ja yhteistyökumppaneiden kanssa. Tämä taasen johtaa siihen, että uuden järjestelmän vaatimukset perustuvat hyvin pitkälti käyttäjien tietoihin.

Koska käyttäjillä on niin paljon tietoa, joka on välttämätöntä tietojärjestelmäprojektin eri vaiheissa on heidän osallistumisensa tietojärjestelmäprojektin eri vaiheisiin hyvin tärkeää projektin onnistumisen kannalta.

2.3. Käyttäjien turhautuminen ongelman ilmentymänä

Tietojärjestelmän hankkijat olettavat saavansa selkeää hyötyä hankkimastaan tietojärjestelmästä. Kuitenkin järjestelmän käyttöön liittyvien ongelmien takia saadut hyödyt saattavat mitätöityä. Mitätöityminen voi johtua siitä, että järjestelmä ei toimi, sitä ei voida tai osata käyttää oikealla tavalla. Mitä useammin käyttäjillä on ongelmia järjestelmän käytössä sitä enemmän se ilmenee käyttäjien turhautumisena.

Käyttäjien turhautuminen voi johtua monesta eri seikasta, niin teknisistä kuin ohjelmallisista seikoista [Bessiere *et al.*, 2002]. Käyttäjien turhautumista voidaan välttää ottamalla käyttäjien tiedot ja taidot paremmin huomioon suunniteltaessa esimerkiksi järjestelmän virheilmoituksia. Virheilmoituksia tärkeämpääkin on käyttäjien huomioiminen käyttöliittymän suunnittelussa. Jos käyttäjiä ei huomioida tarpeeksi ja heidän sallitaan turhautua järjestelmän käyttöön, voi lähes puolet tietokoneen ääressä vietetystä ajasta kulua hukkaan. Turhautumista voidaan ehkäistä myös järjestelmän käyttöönottovaiheen jälkeen järjestämällä riittävää tukea, koulutusta ja ohjeistusta järjestelmän käyttöön [Bessiere *et al.*, 2002].

3. Käyttäjät osana tietojärjestelmäprojektien eri vaiheita

3.1. Tietojärjestelmäprojektin vaiheistaminen

Tietojärjestelmäprojektien vaiheistamista varten on kehitetty lukuisia erilaisia malleja. Perusta on se totuus, että jokaisessa projektissa täytyy tehdä samat perusasiat, jotta saataisiin aikaan toimiva kokonaisuus. Vaiheistuksissa eroina ovat lähinnä eroavaisuudet siinä kuinka nämä työvaiheet jaotellaan. Yleisiä vaiheita ovat; organisaation ja strategian analyysi, tietojärjestelmän määrittelyvaihe, tietojärjestelmän looginen suunnittelu, järjestelmän

ohjelmointi tai muu tekninen toteutus ja käyttöönotto [Ruuhonen ja Salmela, 1999, s.167].

Kirjassaan Ruuhonen ja Salmela [1999, s.76-79] kolme erilaista tapaa jaotella tietojärjestelmäprojekti vaiheisiin. Klassinen vaihejakomalli eli vesiputousmalli jakaa projektin kuuteen eri vaiheeseen. Nämä vaiheet ovat seuraavat; 1. esikartoitus, 2. tietojärjestelmien määrittely, 3. tietojärjestelmien suunnittelu, 4. tietojärjestelmien toteutus, 5. tietojärjestelmien käyttöönotto ja 6. tietojärjestelmien ylläpito.

Toinen Ruuhosen ja Salmelan [1999] kuvaama tapa on prototyypilähestymistapa. Tässä tavassa käydään kaikki toteutusta edeltävät vaiheet käydään hyvin nopeasti läpi ja tehdään niiden perusteella prototyyppi. Käyttäjiltä pyydetään palautetta prototyypistä, jonka jälkeen prototyyppiä kehitetään saadun palautteen mukaan. Tämä jälkeen prototyyppi palaa käyttäjille kommentoivaksi, jonka jälkeen sitä kehitetään taas. Tätä prototyypin vaihevaiheelta kehittämistä jatketaan kunnes prototyyppi vastaa riittävän hyvin sille asetettuja tarpeita. Kun tämä vaihe on saavutettu käydään prototyypin tekninen toteutus läpi kertaalleen, tehdään dokumentointi ja testataan prototyyppi virheiden varalta. Vaikean ennakoitavuutensa takia prototyypilähestymistapa ei sovellu kuin hyvin harvoihin tapauksiin [Ruuhonen ja Salmela, 1999, s.78].

Kolmantena tapana Ruuhonen ja Salmela [1999] esittävän vaihejakomallin ja prototyypimallin yhdistelmään. Tässä mallissa järjestelmälle luodaan vaihejakomallin vaiheita noudattaen vaatimusmäärittely. Tämän määrittelynpohjalta sitten luodaan prototyyppi jonka perusteella testataan määrittelyn oikeellisuus. Siinä tapauksessa, että käyttäjät eivät hyväksy prototyyppiä aloitetaan alusta, muuten siirrytään noudattamaan vaihejakomallia.

3.2. Käyttäjien huomiointi eri lähestymistavoissa

Vaihejakomalli jättää paljon vapauksia sille kuinka laajalta käyttäjät otetaan huomioon tietojärjestelmänprojektin aikana. Jokaisessa vaihejakomallin vaiheessa on mahdollista edetä sillä tavoin, että käyttäjät ovat tiiviisti osa projektia. Toisaalta taas on mahdollista toteuttaa koko järjestelmän vaihejakomallin mukaisesti ettei käyttäjiä huomioida kuin vasta vaatimusten hyväksymisessä, tällöinkin käyttäjiä saattaa edustaa vain joku johtoportaan kuuluva henkilö. Vaihejakomallin väljyys mahdollistaa tilanteen jolloin järjestelmän varsinainen käyttäjä tulee mukaan kuvaan vasta uuden järjestelmän käyttökoulutus vaiheessa. Tällöin on helppo ajautua tilanteeseen jolloin toteutettu järjestelmä ei vastaa käyttäjien todellisia tarpeita tai ei sovellu käytettäväksi käyttäjien työtehtävissä ilman kankeita työtä hidastavia ja

vaikuttavia proseduureja, joiden noudattaminen saattaa vaatia totutun ja hyväksi havaitun työskentelytavan muokkaamista huonommaksi. Yhtenä esimerkkinä mainittakoon tilanne, jossa yleisimmin käytetyt toiminnot ovat haudattu useamman painalluksen taakse monimutkaisiin valikkorakenteisiin.

Prototyypilähestymistavassa käyttäjät ovat keskeisessä osassa järjestelmän toteutusta. Koska järjestelmälle ei laadita kattavaa vaatimusmäärittelyä ennen ensimmäisiä prototyyppejä ja niiden arvioimista, jää käyttäjille suuri vastuu ja valta vaikuttaa siihen kuinka toteutettava järjestelmä muotoutuu ja kuinka se tulee toimimaan. Prototyypilähestymistavan mukaisessa projektissa käyttäjien osallistuminen on lähes maksimaalista, mutta useimmiten ei kuitenkaan tehokkain tapa huomioida käyttäjät osana tietojärjestelmäprojektia. Tehokkuutta vähentää useiden eri prototyyppien arviointiin kuluva aika ja koko prosessin keston venyminen hyvin pitkäksi.

Yhdistetyssä mallissa on vahvana puolena vaatimusmäärittelyn oikeellisuuden tarkistaminen käyttäjillä prototyypin avulla. Koska koko projekti perustuu vaatimusmäärittelyyn, pääsevät käyttäjät prototyypin arvioimalla vaikuttamaan koko projektin perustaan ja siihen kuinka projektissa toteutettava tietojärjestelmä muotoutuu. Tällöin on hyvät lähtökohdat saavuttaa tietojärjestelmäprojektissa hyvä tulos, joka vastaa juuri niihin tarpeisiin joihin projektilla haettiin vastausta.

4. Käyttäjäläheisiä metodeja

Käyttäjien huomioimiseksi tietojärjestelmäprojekteissa on kehitetty useita eri metodeja. Näitä eri metodeja hyödynnetään projektin eri vaiheissa. Tässä luvussa kuvataan lyhyesti kolme erilaista metodia, joissa käyttäjien huomioiminen on pääosassa. Lisäksi kuvataan lyhyesti Rosebud Corporationin SAP-järjestelmän käyttöönottoprojektia ja Turun terveydenhuolto-organisaation tietojärjestelmän uusimisprojektia esimerkkitapauksina. Luvun loppuun arvioidaan eri metodien vahvuuksia ja heikkouksia. Esimerkki metodit valittiin sen perusteella, että ne kuvaavat kuinka käyttäjiä voidaan ottaa huomioon tietojärjestelmäprojektin eri vaiheissa. Rosebud-metodi kuvaa käyttäjien osallistumisen tärkeyttä työtehtävien kartoittamisessa, kun taas kriittisen tehtävän-metodi kuvaa käyttäjien huomioimisen tärkeyttä järjestelmän toiminnan kannalta elintärkeiden toimintojen löytämisessä. "Turun HCIS"-metodi taas kuvaa koko tietojärjestelmäprojektin kokonaisuudessaan hyvin korkealla tasolla.

4.1. Rosebud-metodi

Artikkelissaan Kawalek ja Wood-Harper kuvaavat SAP-järjestelmän käyttöönottoprosessia Rosebud Corporationissa [Kawalek and Wood-Harper, 2002], metodi nimettiin siis sen yhtiön mukaan jonka tietojärjestelmän projektissa sitä sovellettiin.

Rosebud-metodi jakautuu kahteen eri vaiheeseen, suunnittelu- ja toteutusvaiheeseen. Käyttäjien huomiointi tulee esille toteutusvaiheen eräissä osa vaiheessa, *liiketoimintaan liittyvien asioiden huomioiminen* -vaiheessa [Kawalek and Wood-Harper, 2002]. Vaikkakin käyttäjien osallistuminen tulee esille vain metodin toisen vaiheen yhdessä alavaiheessa, toteavat Kawalek ja Wood-Harper käyttäjien osallistumisen olevan tärkeä metodin osa-alue. Käyttäjien osallistumista tarvitaan eri työpisteissä tehtäviin töihin liittyvien tietojen ja mahdollisten ongelmien kartoittamisessa ja keräämisessä. Edellä mainittu on sinänsä itsestäänselvyys, koska ilman käyttäjiä eli työntekijöitä olisi lähes mahdotonta määrittellä eri työtehtäviin liittyvät seikat ongelmineen.

Käyttäjien osallistuminen huomioidaan toisessakin toteutusvaiheen osavaiheessa, *järjestelmän ymmärtäminen* -vaiheessa. Tässä vaiheessa heillä ei ole mahdollisuutta antaa tietoa toteuttajille, vaan käyttäjien rooli on lähinnä tutustua SAP-järjestelmään ja opettelemaan sen käyttöä. Tämän jälkeen heillä on tietopohjaa ilmaista mahdollisia ideoita ja nostaa esille kysymyksiä järjestelmästä. Tämä ideoiden ja kysymysten esille tuominen kuuluu aikaisemmin mainittuun *liiketoimintaan liittyvien asioiden huomioiminen* -vaiheeseen.

Rosebud Corporationin tapauksessa käyttäjien osallistumisen todettiin olleen välttämätöntä projektin onnistumisen kannalta. Kawalek ja Wood-Harper toteavatkin käyttäjien ilmaiseen tyytyväisyytensä siihen, että heidät huomioitiin järjestelmän käyttöönotossa ja että heidän mielipiteitään arvostettiin ja hyödynnettiin. Käyttäjien tyytyväisyys takasi heidän tukevan SAP-järjestelmän käyttöönottoa, joka oli projektin onnistumisen kannalta elintärkeää. Jos käyttäjät eivät ota järjestelmään omakseen alkamalla käyttää sitä työssään, niin tietojärjestelmäprojekti on epäonnistunut.

4.2. Kriittisen tehtävän-metodi

Harris ja Brightman kuvaavat artikkelissaan [Harris and Brightman, 1985] kriittisen tehtävän - metodia. Kutsun metodia jatkossa CTM:ksi, lyhenne tulee metodin englanninkielisestä muodosta *the Critical Task Method* [Harris and Brightman, 1985]. CTM:ssä pyritään määrittämään ne tehtävät, joiden suorittaminen on käyttäjille välttämätöntä, jotta he pystyvät suoriutumaan tehtävistään [Harris and Brightman, 1985].

CTM jaetaan viiteen vaiheeseen, jotka ovat seuraavat [Harris and Brightman, 1985]:

Vaihe 1. Haastattele ryhmää käyttäjistä

Haastatellaan osaa käyttäjistä, jotta saataisiin selville kuinka he suorittavat tehtävänsä.

Vaihe 2. Kehitä profiili tehtäväkuvauksista

Haastatteluiden perusteella jaetaan tehtävät kahteen ryhmään: ylläpito ja kognitiivisiin tehtäviin. Ylläpitotehtäviksi luetaan yleisluontoiset tehtävät, kuten esimerkiksi arkistointi ja kirjoittaminen. Kognitiivisiksi tehtäviksi luetaan sellaiset tehtävät joilla on selkeä tulos, kuten esimerkiksi kirjallinen raportti.

Vaihe 3. Kehitä profiili tukitavoista

Kootaan haastatteluiden perusteella lista eri tukitavoista, joilla tehtävien suorittamista tuetaan

Vaihe 4. Vahvista tehtäväkuvausten profiili

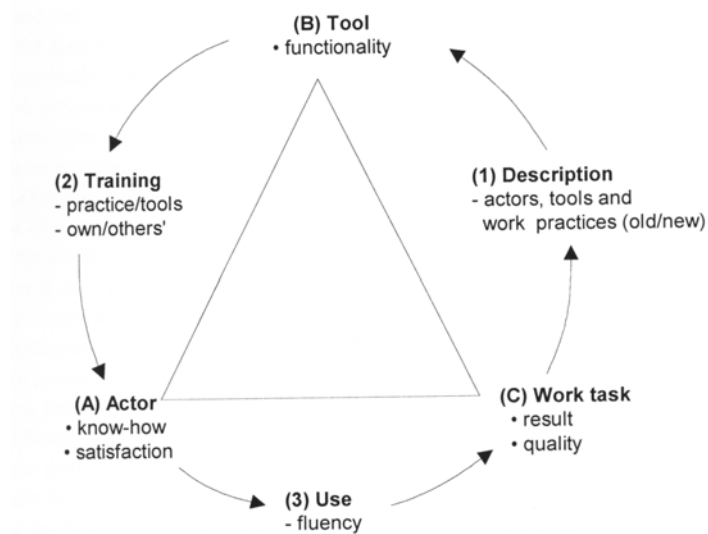
Tehtäväkuvauksien profiili ja tukitapojen profiili esitetään osalle käyttäjistä ja pyydetään heitä vahvistamaan niiden tarkkuus, selkeys ja riittävyys ja tarvittaessa tekemään muutoksia ja muokkauksia.

Vaihe 5. Tarkistuta vertailu ryhmällä

Osa lopuista käyttäjistä pyydetään jakamaan tunnistetut tehtävät pullonkauloihin, ylläpitotehtäviin ja kognitiivisiin tehtäviin. Lisäksi heitä pyydetään ilmaisemaan ne tukitavat joita he käyttävät kunkin pullonkaulan läpäisemiseksi. Tämän jälkeen tukitavat luokitellaan eri määreiden mukaisesti.

4.3. "Turun HCIS"-metodi

Aaltonen, Nurminen, Reijonen ja Vuorenheimo [Aaltonen *et al.*, 2002] kuvaavat artikkelissaan Turun kaupungin terveydenhuolto-organisaation tietojärjestelmän uusimisessa käytettyä metodia. He eivät anna käytetylle metodille varsinaista nimeä, jonka takia otin vapauden kutsua käytettyä metodia tässä artikkelissa "Turun HCIS"-metodiksi. Lyhenne HCIS tulee englanninkielisestä termistä *Health Care Information System* [Aaltonen *et al.*, 2002]. Valitsin nimen sen perusteella, että artikkelissaan Aaltonen, Nurminen, Reijonen ja Vuorenheimo [Aaltonen *et al.*, 2002] kuvaavat metodin soveltamista käytäntöön Turun terveydenhuolto-organisaation tietojärjestelmän uusintaprojektissa.



Kuva 2. Semanttinen kuvaus toteutusmetodista [Aaltonen *et al.*, 2002].

Kuvassa 2. kuvataan toteutusmetodin semanttinen kuvaus. A on käyttäjä (*Actor*), B on työkalu (*Tool*) ja C on tehtävä (*Work task*) kuvaavat asiayhteyden johon tietojärjestelmä toteutetaan. Kuten kuvasta 2. ilmenee, metodi on jaettu kolmeen eri vaiheeseen; 1 Kuvaus (*Description*), 2 Koulutus (*Training*) ja 3 käyttö (*Use*). Metodin mukaisesti ensimmäisessä vaiheessa kuvataan käyttäjät, heidän työtehtävänsä ja niihin tarvittut työkalut. Aaltonen, Nurminen, Reijonen ja Vuorenheimo [Aaltonen *et al.*, 2002] toteuavat suurempien asiakokonaisuuksien löytyvän helposti organisaatiokaavioista. Pienempiä yksityiskohtia on vaikeampi saada selville. Yksityiskohtien selvittäminen vaatii käyttäjien haastattelemista. Koulutus vaiheessa edetään siten, että ohjelmiston toteutuksesta vastanneen tahon edustajat kouluttavat osan käyttäjistä kouluttamaan loput käyttäjistä käyttämään järjestelmää. Viimeisessä vaiheessa eli käyttövaiheessa on tärkeää järjestää tukea järjestelmän käyttöön ja arvioida kuinka järjestelmän käytön luontevuutta ja tehokkuutta. Arvioiden perusteella voidaan tehdä tarvittavat muutokset järjestelmään. Arvioinnin voidaan tehdä yksinkertaisimmillaan haastattelemalla käyttäjiä. Artikkelissaan Aaltonen, Nurminen, Reijonen ja Vuorenheimo [Aaltonen *et al.*, 2002] kuvaavat kuinka käyttäjiä tulisi huomioida tietojärjestelmän projektin erivaiheissa.

5. Ongelmia käyttäjien huomioimisessa

Käyttäjien huomioimiseksi ja ottamiseksi mukaan tietojärjestelmäprojektiin on monia eri tapoja, jokaisessa tavassa on kuitenkin ongelmansa. Osa ongelmista liittyy käyttäjiin itseensä ja osa järjestelmän toteuttajiin. Ei voida myöskään

todeta itse menetelmien olevan ongelmattomia, eli syitä ongelmiin voi olla monia.

Nykyisin globaalissa markkinataloudessa tietojärjestelmäntuottaja voi olla kotoisin aivan eri kulttuurista kuin järjestelmän käyttäjät. Tämä kulttuuri ero voi muodostua ongelmaksi jos järjestelmäntoimittajilla ei ole riittävään tuntemusta käyttäjien kulttuurista [Dagwell and Weber, 1983]. Eri kulttuureissa esimerkiksi väreillä on eri merkitykset ja tekstin kulkee eri suuntaan kuin perinteisessä länsimaalaisessa vasemmalta oikealle ja ylhäältä alas suunnassa.

Käyttäjätkin on hyvin monenlaisia, käyttäjät eroavat kokemustensa ja osallistumisinnostuksensa puolesta hyvinkin paljon. Voi hyvinkin olla että juuri ne käyttäjät joilla olisi eniten hyödyllistä tietoa eivät ole innostuneita osallistumaan tietojärjestelmäprojektiin, kun taas nuoret työntekijät ovat perinteisesti olleet innokkaimpia osallistumaan erilaisiin kehitystehtäviin [Kramer *et al.*, 2000]. Käyttäjillä on myös erilaiset ja eri tavalla kattavat näkemykset asioista. Eroja saattaa olla myös käytetyissä termeissä [Kramer *et al.*, 2000].

Toteuttajat voivat taasen olla hyvinkin tietoisia siitä kuinka käyttäjät tulisi ottaa osaksi tietojärjestelmäprojektia, mutta he eivät osaa korostaa asian tärkeyttä tarpeeksi. Vaikka toteuttajat korostaisivatkin käyttäjien roolia tarpeeksi, voi heidän yrityksensä tuoda käyttäjät osaksi projektia epäonnistua. Syynä epäonnistumiseen on yleisesti se seikka, että toteuttajilla ei ole useinkaan koulutusta ja kokemusta muusta kuin toteutusta koskevista asioista. Toteuttajilla ei ole kokemusta ja tietoutta liiketoiminnasta, psykologiasta ja ihmisten välisistä suhteista, jotta he voisivat menestyksekkäästi hallita käyttäjien osallistumista ja käsitellä käyttäjiä sillä tavoin jotta heidän tiedoistaan olisi mahdollisimman suuri hyöty projektin toteutumisen kannalta. [Desanctis and Courtney, 1983]

On myöskin hyvin todennäköistä, että toteuttajilla ja käyttäjillä on vaikeuksia löytää "yhteistä kieltä". Toteuttajat ovat tottuneet yksityiskohtaisiin ja teknisiin kuvauksiin, kun taas käyttäjät ovat tottuneet yleisluontoisempiin ja helpommin hyödynnettäviin kuvauksiin [Suchman, 1983]. Tämä "yhteisen kielen" löytämisen ongelma on ensimmäinen ongelma johon täytyy löytää ratkaisu, jotta toteuttajien ja käyttäjien yhteistyö olisi alun alkaenkaan mahdollista.

6. Keskustelua

Tässä tutkielmassa olen käsitellyt monia puhuttavaa asiaa käyttäjien roolia tietojärjestelmäprojekteissa. Viime aikoina on käyttäjien roolia alettu korostaa entistä enemmän, tämä ei kuitenkaan näy kovinkaan usein käytännössä. Syytä

käyttäjien osallistumisen vähyyteen on helppo lähteä hakemaan käytössä olevista metodeista ja toteuttajista. Jos käytetyt metodit eivät tue käyttäjän osallistumista niin on käyttäjien rooli lähes yhdentekevä.

Ongelmallisia ovat myös toteuttajien eli ohjelmoijien ja järjestelmäsuunnittelijoiden asenteet käyttäjiä kohtaan. Nykyinen koulutus opettaa kyllä tehokkaasti molempia, niin toteutukseen kuin käyttäjien huomioimiseen liittyviä seikkoja. Ainoastaan näitä asioita opetetaan eri ihmisille, hyvin usein ohjelmoijilla ei ole ollenkaan kokemusta käyttäjien huomioimisesta. Ohjelmoijilla on myös usein hyvin vähättelevä asenne käyttäjiä kohtaan, he ovat teknisesti orientoituneita ja eivät näe mitä hyötyä käyttäjien osallistumisesta olisi. Tämä ohjelmoijien tekninen orientoiminen osoittaa, että vaikka tutkimus ja teoriat ovat muuttuneet paljonkin aikojen kuluessa voivat asenteet olla vielä samat kuin tietojärjestelmien historian aamuhämärissä.

Teorioissakin on paljon kehittämistä, on olemassa useita eri malleja joilla tietojärjestelmäprojekti voidaan jakaa osiin ja eri metodeja joita näissä eri osissa voidaan hyödyntää. Mutta ainakaan toistaiseksi ei ole pystytty kehittämään sellaista vaiheistusta jossa eri vaiheissa käyttäjät täytyisi huomioida riittävässä määrin. Myöskin olemassa olevissa metodeissa on kehittämisen varaa. Metodeja tulee kehittää sellaisiksi, jotta toteuttajat olisivat kykeneviä niitä hyödyntämään ja käyttäjät kykeneviä toimimaan metodeissa kuvatuissa rooleissa.

Käyttäjillä on tietoa, järjestelmän tuottajat tarvitsevat tätä tietoa voidakseen toteuttaa tarkoitustaan vastaavan järjestelmän. Ongelma on siinä kuinka tämä tieto saadaan käyttäjien aivoista tuottajien käytettäväksi sellaisessa muodossa ja laajuudessa, että se on hyödynnettävissä tehokkaasti. Käyttäjiä tulee kouluttaa kuvailemaa tietoa tarpeeksi kattavasti ja pyrkimään eliminoimaan hiljaisen tiedon pysyminen poissa tuottajien saatavilta. Toisaalta tuottajia pitää kouluttaa muissakin asioissa kuin pelkästään järjestelmien toteuttamiseen liittyvissä seikoissa. Tuottajilla tulisi olla valmiuksia ymmärtää käyttäjien toimintaa ja työtehtäviä.

Kaikkeen tähän on olemassa teoriassa valmiudet, on olemassa riittävästi tietoa luoda toimiva jako eri vaiheisiin, tehokkaita metodeja ja kouluttaa käyttäjiä ja tuottajia. Tämän tiedon kokoaminen sellaiseen muotoon jossa materialisoituu toimiva jako vaiheisiin, tehokkaat metodit ja kunnollinen koulutus on alue jossa riittää tutkittavaa hyvin paljon. Tutkimuksia tehdessä on kuitenkin pidettävä mielessä, että pelkkä tutkiminen ei riitä. Tehdyn tutkimuksen tulokset on myös osattava hyödyntää ja hyödynnettävä todellisuuteen, muuten tutkimus on ollut hyödytöntä.

7. Lähteet

- [Aaltonen *et al.*, 2002] Satu Aaltonen, Markku I. Nurminen, Pekka Reijonen and Jaana Vuorenheimo, User-driven implementation of information systems. *Information Systems Research Seminar in Scandinavia, Proceedings*, 2002.
- [Bessiere *et al.*, 2002] Katie Bessiere, Irina Ceaparu, Jonathan Lazar, John Robinson, and Ben Shneiderman, Social and Psychological Influences on Computer User Frustration. In: Erik P. Bucy and John E. Newhagen (eds.), *Media Access - Social and Psychological Dimensions of New Technology Use*, Lawrence Erlbaum Assoc, 2003.
- [Dagwell and Weber, 1983] Ron Dagwell and Ron Weber, System designers' user models: a comparative study and methodological critique. *Communications of the ACM*, **26** (11), 1983, 987-997.
- [Davis, 1985] Joseph G. Davis, A typology of management information systems users and its implications for user information satisfaction research. *Computer personnel reseach, ACM SIGCPR '85 Conference proceedings*, ACM Press, 1985, 152-164.
- [Desanctis and Courtney, 1983] Gerardine Desanctis and James F. Courtney, Toward friendly user MIS implementation. *Communications of the ACM*, **26** (10), 1983, 732-738.
- [Harris and Brightman, 1985] Sidney E. Harris and Harvey J. Brightman, Design implications of a task-driven approach to unstructured cognitive tasks in office work. *ACM Transactions on Office Information Systems*, **3** (3), 1985, 292-306.
- [Kawalek and Wood-Harper, 2002] Peter Kawalek and Trevor Wood-Harper, The finding of thorns: user participation in enterprise system implementation. *The data base for Advances in Information Systems*, **33** (1), 2002, 13-22.
- [Kettunen ja Simons, 2001] Jari Kettunen & Magnus Simons, *Toiminnanohjausjärjestelmän Käyttöönotto Pk-yrityksessä. Teknologia- ja tietotekniikan ajattelusta kohti tiedon ja osaamisen hallintaa*. VTT Julkaisuja - Publications 854. Valtion Teknillinen tutkimuskeskus, 2001.
- [Kramer *et al.*, 2000] Joseph Kramer, Sunil Noronha, and John Vergo, A user-centered design approach to personalization. *Communications of the ACM*, **43** (8), 2000, 45-48.
- [Ruohonen ja Salmela, 1999] Mikko J. Ruohonen ja Hannu Salmela, *Yrityksen tietohallinto*, Edita, 1999.
- [Suchman, 1983] Lucy A. Suchman, Office procedure as practical action: models of work and system design. *ACM Transactions on Office Information Systems*, **1** (4), 1983, 320-328.

Matkapuhelinten käytettävyyso ongelmia pelaajan näkökulmasta

Valtteri Pihlajamäki

Tiivistelmä

Tutkielma käsittelee matkapuhelinten käytettävyyso- ja käyttöliittymäongelmia pelaajan näkökulmasta. Tällaisia ongelmia aiheuttavat esimerkiksi matkapuhelinten näyttöjen ja näppäimistöjen toiminta ja suorituskyvyn riittämättömyys. Tutkielma tarjoaa ratkaisun useaan ongelma kohtaan ja luo katsauksen matkapuhelinpelien tulevaisuudennäkymiin.

Avainsanat ja -sanonnat: matkapuhelin, käytettävyyso, käyttöliittymä, matkapuhelinpelit, mobiilipelit, mobiilipelaaminen.

CR-luokat: H.5.2, C.5.3, K.8

1. Johdanto

Tutkielmani käsittelee matkapuhelinten käytettävyyso ongelmia pelaajan näkökulmasta. Aihe on tällä hetkellä erittäin ajankohtainen, koska useat matkapuhelinvalmistajat - esimerkiksi Nokia - panostavat erittäin kovasti mobiililaitteidensa peliominaisuuksiin. Nokialta ilmestyi 7.10.2003 puhelinmalli N-Gage, jota Nokia itse kutsuu mieluummin mobiilipelikonsoliksi kuin matkapuhelimeksi [Nokia, 2003]. Myös muut valmistajat ovat parantaneet huomattavasti matkapuhelintensa peliominaisuuksia. Väri näyttöt, monikanavaiset äänet sekä erilaiset ohjaimet, kuten tappihiiret ovat omalta osaltaan parantaneet puhelinten peliominaisuuksia. Matkapuhelinpelien maailmanlaajuisten markkinoiden uskotaan kasvavan vuoteen 2005 mennessä yli kolmeen miljardiin dollariin [Junkkari, 2003].

Pelien invaasio puhelimiin alkoi Nokian 5110:sta ja muista sen aikaisista malleista (1990-luvun puolivälin jälkeen). Matopeli, yksi maailman vanhimmista tietokonepeleistä, oli suuri hitti ominaisuus Nokian matkapuhelimissa. Matopelistä järjestettiin jopa suomenmestaruuskilpailuja: vuonna 1999 suomenmestaruudesta kilpaili noin 500 osallistujaa [Press.nokia.fi, 2000]. Myös monet muut matkapuhelinvalmistajat alkoivat tuottaa pieniä pelejä matkapuhelimiinsa. Matkapuhelinten peliominaisuudet eivät kuitenkaan käytännössä kehittyneet muutama vuoteen. Pelit kärsivät tuolloin puhelinten todella huonosta laskentatehokkuudesta, näyttöjen epätarkkuudesta ja väriltömyydestä, monikanavaisten äänten puutteesta sekä erilaisista ongelmista puhelinten näppäinten käytössä pelaamiseen. Pelit olivat pitkään integroitua puhelimiin: niitä ei voinut poistaa eikä uusia saanut ladattua. Lisäksi niiden

kehityskieli oli sama, jolla puhelimen kaikki muutkin ominaisuudet ohjelmoitiin.

Viime vuosina matkapuhelimet ovat kokeneet suuria muutoksia. Uudet matkapuhelimet tukevat Java-kieltä. Tämä mahdollistaa sen, että kuka tahansa Java-kieltä taitava ohjelmoija voi ohjelmoida pelejä periaatteessa kaikille Java-kieltä tukeville puhelimille [Amaro, 2003]. Kameroiden integroituminen matkapuhelimiin toi niihin vakio-ominaisuudeksi ison värinäytön. Soittoäänien muuttuminen monikanavaisiksi mahdollistaa kunnollisen äänen toistamisen myös mobiilipelien yhteydessä. Esimerkiksi nämä ja monet muut ominaisuudet mahdollistavat periaatteessa nykyaikaisten pelien toimimisen matkapuhelimeissa. Useat seikat vaikeuttavat kuitenkin pelien siirtymistä matkapuhelimiin. Tämä tutkielma esittelee tekijöitä, jotka aiheuttavat käytettävyyssongelmia matkapuhelinpeleihin ja tarjoaa niihin ratkaisuja.

Tarkastelen tutkielmassani perinteisiä matkapuhelimelleja. Esimerkiksi eri tyyppiset kämmenmikrot eivät kuulu tämän tutkielman sisältöön, vaikka niissä olisikin puhelimen ominaisuuksia. Perinteisen matkapuhelimen rajat alkavat kuitenkin hämärtyä, mitä pidemmälle tuotekehittely etenee. Kuten jo aikaisemmin mainitsin, Nokia ei kutsu N-Gage:aan matkapuhelimeksi vaan mobiilipelikonsoliksi. Joka tapauksessa N-Gage rakentuu matkapuhelimen ja sen ominaisuuksien ympärille. Toisin kuin esimerkiksi Nokia Communicator, joka pyrkii mielestäni olemaan enemmänkin kannettava tietokone täysmittaisine näppäimistöineen ja toimistosovelluksineen. Käsittelen tutkielmassani pääasiassa kolmea matkapuhelinten osa-aluetta: näyttöä, näppäimistöä ja suorituskykyä. Näistä kolmesta osa-alueesta aiheutuu eniten matkapuhelimella pelaamiseen liittyviä käytettävyyssongelmia.

Tutkielmani käsittelee mobiilipelejä. Mobiilipelien käsitettä on jossain yhteyksissä venytetty aina arpakuutioihin ja 1700-luvun optisiin leluihin asti [Järvinen, 2002]. Tarkoitan tässä tutkielmassa mobiilipeleillä ja matkapuhelinpeleillä kuitenkin elektronisilla pelikonsoleilla (esimerkiksi Nintendo GameBoy) tai matkapuhelimilla pelattavia pelejä.

Tutkielmani lähteet jakautuvat kolmeen päätyyppiin: mobiilipelaamista, -käyttöliittymiä ja -teknologiaa käsittelevät lähteet. Matkapuhelinten käytettävyyssongelmia pelaajan näkökulmasta käsitteleviä lähteitä minulla ei ole ollut käytössäni. Minulla on ollut testattavana seuraavat puhelinmallit: Nokia 3510i ja Siemens M55, Nokia 6600 ja Nokia N-Gage. Oliver Maio [Maio, 2002] käsittelee artikkelissaan Nokia 6310i ja 3650 -puhelinten peliominaisuuksia.

2. Ongelmien kartoitus

Kuten johdantokappaleessa mainitsin, matkapuhelinvalmistajat ovat alkaneet kehittää puhelinmalliensa peliominaisuuksia usealla tavalla. Pelien yhdistäminen matkapuhelimiin on edistänyt puhelinten myyntiä [Collier, 2003].

2.1. Taustaa

Pelien ja matkapuhelinten integraation alkuvaiheessa eniten ongelmia tuottivat puhelinten erittäin rajallinen laskentateho sekä pienet ja epätarkat näytöt. Puhelimet oli suunniteltu puhumista ja korkeintaan tekstiviestien kirjoittamista varten. Pelit ohjelmoitiin samalla kehityskielellä kuin puhelinten muutkin ominaisuudet.

Mind Trek -seminaarissa [Mind Trek 2003] nousi esille, että 1990-luvun loppupuolen teknologiahuumassa matkapuhelinpelien sisällöntuotantoon panostettiin jopa liikaa. Wapin piti olla vallankumouksellinen ominaisuus ja wap-pelien 2000-luvun alkupuolella jokaisen matkapuhelinta käyttävän ihmisen suurta huvia. Näin ei kuitenkaan tapahtunut. Teknologia asetti jälleen esteen, josta aiheutui seuraavanlainen ongelmien kierre: Wap-yhtiöiltä puuttui niin sanottu hittituote (engl. killer application) ja sisällöntuottajia palkattiin suuret joukot suunnittelemaan tällaisia pelejä. Samaan aikaan oli pula tarvittavan teknologian hallitsevista henkilöistä, ja yritykset joutuivat kouluttamaan kaikki työntekijät itse. Aikataulut venyivät, ja kustannukset nousivat järjettömiksi. Wap-teknologia ei yleistynyt ja kaiken lisäksi operaattorit hinnoittelivat wapin käytön niin, että suuri yleisö ei halunnut edes testata sitä, ja ne jotka testasivat, huomasivat sen olevan liian raskas käyttää. Wapilla ei pystynyt tekemään mitään, mitä ei olisi pystynyt tekemään jollain muulla tavalla nopeammin, tehokkaammin ja turvallisemmin. Myöskään laitteet eivät olleet kehittyneet juuri yhtään, vaikka wap-tuki puhelimista löytyikin. Puhelimeissa oli edelleen sama pieni kaksiväri näyttö ja huono suorituskyky. Jouduimme todistamaan 2000-luvun taitteessa wap-yhtiöiden huikeita syöksykierteitä ja oman osansa tappioista nielivät myös isommat yhtiöt, varsinkin puhelinoperaattorit. 1998 tehtyjen suunnitelmien mukaan nykyään pitäisi olla käytössä jo niin sanotut 3G-puhelimet, nopeat langattomat yhteydet ja runsaasti niihin sisällytettyjä hyödyllisiä ja viihteellisiä ominaisuuksia. Teknologiakuplan puhkeaminen vaikutti raskaasti tähän kehityskäyrään.

Tällä hetkellä ollaan ehkä järkevämällä kehityksen tiellä. Tosin vasta lähitulevaisuus näyttää, miten paljon järkevämällä tiellä, vai jouduttiinko teknologia-alan taantuman myötä suvantovaiheeseen, josta koko teknologinen

kehitys - mukaanlukien matkapuhelimet - kärsii. Vasta nyt alkavat matkapuhelinten ohjelmistot, kehitystyökalut ja laskentatehokin tietyissä malleissa olla sillä tasolla, että niihin kannattaisi ammattimaisesti alkaa suunnitella pelejä [Amaro, 2003]. Kuitenkin suurin osa matkapuhelinten pelikehityksessä käytettävästä teknologiasta on niin nuorta ja epästandardia, että se aiheuttaa omat ongelmansa. Tällä hetkellä on muutamia mahdollisia ohjelmointitekniikoita, joista saattaa tulla alan standardi. Yksi yleisimmistä on Java-teknologia. Muihin tekniikoihin lukeutuu esimerkiksi BREW, jota pidetään tehokkaampana kuin Javaa [Barbagallo, 2002]. Java on kuitenkin yleistynyt nopeasti ja sen uusimmat versiot tukevat suoraan matkapuhelinten monia peliominaisuuksia [Kontio, 2003]. Myös Javan tehokkuus kasvaa jokaisen kehitysversion myötä [Amaro, 2003]. Keskityn tutkimuksessani ohjelmointitekniikoiden osalta pääsääntöisesti Javaan. Muita mobiililaitteiden ohjelmointialustoja ovat muun muassa ExEn, Mophon ja WGE [Amaro, 2003].

Kuvitellaan, että olemme siinä pisteessä, että matkapuhelimissamme on tarkat ja mahdollisimman isot värinäytöt, hyvät äänet, riittävästi laskentatehoa ja standardienmukaiset ohjelmointityökalut. Periaatteellisella tasolla näin nykyaikana onkin. Vaikka puitteet ovat periaatteessa hyvien pelien luomiselle olemassa, tulee vastaan sekä perinteisiä käytettävyysongelmia että teknologisia ongelmia. Java on kielenä laitteistoriippumaton, mutta silti käytännössä jokainen peli täytyy ohjelmoida jokaiselle puhelinmallille erikseen. Esimerkiksi puhelinmallien erilaiset näytöt (koko, tarkkuus, maksimivärimäärä), erilaiset näppäimistöt ja eritehoiset suorittimet vaikuttavat osaltaan siihen, että mikä toimii toisessa puhelimessa, ei välttämättä toimi toisessa tai toimii väärin.

2.2. Näppäimistö

Kuten jo edellä mainitsin, näppäimistö on yksi ongelmakehoita matkapuhelimella pelattaessa. Siihen liittyy niin toiminnollisia ongelmia kuin perinteisiä käytettävyysongelmiakin. Toiminnollisista ongelmista yleisin tämänhetkissä puhelinmalleissa on yhtäaikaisten näppäinpainallusten toimimattomuus. Se, ettei edes kahta näppäintä voi painaa yhtä aikaa, aiheuttaa suuria rajoituksia pelin kehittämiseksi. Esimerkiksi perinteinen tasohyppely on erittäin vaikea kehittää siten, että pelaajan ei tarvitse käyttää yhtäaikaista painalluksia [Matthews, 2002].

Kun kysymys on toistaiseksi Nokian N-Gagea ja muutamaa muuta erikoista mallia lukuun ottamatta tavallisen tyyppisistä matkapuhelimista, aiheuttaa näppäimistö pelaajalle myös perinteisempiä käytettävyysongelmia. Tavallisella tarkoitan tässä sitä, että numeronäppäimet ovat puhelimen alaosassa symmetrisessä järjestyksessä, ja näyttö puhelimen yläosassa [Maio,

2002]. Näytön ja numeronäppäimistön välissä sijaitsee niin ikään symmetrisessä järjestyksessä puhelimen toimintonäppäimet, joista tärkeimpinä puhelun yhdistävät ja sulkevat näppäimet tai näppäin [Nokia, 2003]. Tietokone- ja videopelien ohjainvalmistajat ovat jo useita vuosia kehittäneet ihmiskäsille toinen toistaan sopivampia pad-mallisia peliohjaimia. Peliohjaimia tarkastellessa selviää nopeasti, että peukalot on havaittu luontevimmaksi tavaksi käyttää ohjaimen painikkeita. Varsinkin, jos pelin näyttö on samassa kämmenelle mahtuvassa laitteessa kuin ohjain, ovat peukalot lähes ainoa tapa kontrolloida laitetta.

Kun vertaillaan matkapuhelinta ja pad-peliohjainta, voidaan nopeasti havaita, että näppäimet on sijoitettu eri tavalla, ja ne ovat suurempia ohjaimessa kuin puhelimessa. Lisäksi matkapuhelin tulisi kääntää sivuttain ja asettaa näyttö keskelle, että kummankin käden peukaloita voisi kunnolla käyttää. Nokia onkin tehnyt näin muutamassa puhelinmallissaan [Nokia, 2003]. Kunnollisen peliohjaimen yhdistäminen matkapuhelimeen vaatisi siis aika isoja muutoksia perinteisen matkapuhelimen suunnitteluun. Tähän asti useimpien valmistajien trendi on kuitenkin ollut pitäytyä perinteisessä mallissa ja pienentää puhelinta: puhelimet painavat pienimmillään jopa reilusti alle sata grammaa [Nokia, 2003]. Pienempi koko aiheuttaa myös näppäimistön pienenemisen ja pelaaminen hankaloituu – varsinkin, jos kyseessä on vähänkin monimutkaisempi peli. Ensimmäisiä perinteisestä mallista poikkeavia puhelimia oli Nokia 5510, joka käsitykseni mukaan ei onnistunut tavoitteissaan olla suosittu viihdepuhelin. Myöskin N-Gagen alkutaival on ollut takkuinen, eikä myynti ole sujunut odotetulla tavalla [Salminen, 2003].

2.3. Näyttö

Näyttö on ollut matkapuhelinpelaamisen alkuajoista lähtien rajoittava tekijä ja on sitä edelleen [Matthews, 2002]. Tosin parempaan suuntaan ollaan koko ajan menossa, ja pian saatetaan olla tilanteessa, jossa matkapuhelimen näytön rajat on saavutettu. Yksinkertaisesti isompaa, nopeampaa, tarkempaa ja värikkäämpää näyttöä ei matkapuhelimeen pian ole enää mahdollista tehdä, ainakaan nykyteknologian avulla. Esimerkiksi Nokian tavallisissa matkapuhelinmalleissa näyttöjen tarkkuus vaihtelee noin 50*100 pikselin mustavalkonäytöistä noin 180*200 täysiväri aktiivimatriisinäyttöihin [Nokia, 2003]. Lähitulevaisuudessa matkapuhelinten näyttötarkkuudet tulevat luultavasti olemaan vielä jonkin verran korkeampia.

Alkuajoista lähtien ongelmia on tuottanut kolme seikkaa: näytön rajoittunut koko, värttömyys sekä hitaasti sammuvat ja syttyvät pikselit [Matthews 2002]. Nokian puhelimissa on ollut suhteellisen isot näytöt jo 5110-

mallista lähtien (aikaisemminkin, mutta tässä mallissa oli ensimmäisenä pelejä). Joka tapauksessa näytöt olivat poikkeuksetta kaikilla valmistajilla liian pieniä mihinkään muuhun kuin matopelien kaltaisiin hyvin pieniin puzzle-peleihin. Oliver Maio [Maio, 2002] toteaa tämän myös Nokia 6310i:n näytöstä. Matopeli toimii mielestäni Nokian puhelinmalleissa erittäin sulavasti ja pelikokemus on miellyttävä. Ilman tarkempaa tutkimusta voidaan vain spekuloida, miten paljon toimiva matopeli on vaikuttanut Nokian myyntilukuihin. Varsinkin niin sanotuissa nuorisomalleissa 5110 ja sen seuraajissa (3210, 3310, 3330 ja niin edelleen) matopelin olemassaolo on saattanut edesauttaa myyntiä huomattavasti. Matopeli ja sen eri variaatiot eivät kuitenkaan voineet olla hittipelejä vuodesta toiseen, joten 1990-luvun lopulla odoteltiin uusia ominaisuuksia matkapuhelimiin [MindTrek 2003]. Pienet kaksiväriset näytöt olivat saavuttaneet rajansa pelien maailmassa.

2000-luvulla puhelimiin alkoi ilmestyä värinäyttöjä. Värinäyttöjen yleistymistä on edesauttanut multimediamiestien keksiminen (viesti, jonka yhteydessä voi lähettää kuvamediaa ja ääntä), sekä varsinkin kameroiden integroituminen matkapuhelimiin. Matkapuhelinkameroiden saapumisen myötä myös näyttöjen tarkkuus on parantunut melko paljon. Värit ja tarkemmat näytöt antavat pelien kehittäjälle mahdollisuuden toteuttaa huomattavasti monipuolisempia pelejä matkapuhelimille kuin aikaisemmin. Näyttöjen tarkkuudet ovat parhaimmillaan lähes Commodore 64 -tietokoneen tasoa ja värimäärät halvoissakin puhelinmalleissa moninkertaiset (Commodore 64:n näyttöasetukset ovat maksimissaan 320*200/16 väriä). Commodore 64:lle tehtiin 1980-luvulla ja 1990-luvun alussa tuhansia pelejä joista parhaimmat olivat todella suosittuja. C64.com -verkkopalvelussa on imuroitavissa yli 4000 erilaisille C64-emulaattoreille käännettyä Commodore-peliä. Peljä on imuroitu palvelusta yli 10 miljoonaa kappaletta [C64.com, 2003]. Valmiiksi suunniteltuja pelejä on olemassa siis valtava määrä, ne vain odottavat matkapuhelimiin kääntämistä. Suoraa plagiointia 80 - 90 -lukujen hittipeleistä onkin ollut runsaasti havaittavissa viime vuosina. Tämä selviää vertailemalla Internetin peliarkistoja matkapuhelimille ja Commodore 64:lle. Japanissa vanhat tietokoneklassikkopelit ovat olleet viime vuosina erittäin suosittuja [Collier, 2003]. Pelikokemuksen saavuttaminen alkuperäiselle tasolle on kuitenkin osoittautunut yllättävän vaikeaksi.

Pelaamiskokemus on monen tekijän summa. Yksi ongelma on näyttöjen passiivisuus. Passiivisten näyttöjen pikselit sammuvat liian hitaasti ja tästä seuraa liikkuvan grafiikan sumentuminen [Matthews, 2002]. Peli näyttää pahimmillaan pelkältä värien sekamelskalta. Ongelmia aiheuttaa myös se, että näytön hitaasta päivittymisestä johtuen peli piirtyy näytölle selkeästi

hitaammin kuin pelaaja siihen reagoi [Matthews, 2002]. Uusimmissa puhelinmalleissa ei tätä ongelmaa enää ole. Esimerkiksi Nokia 6600-mallissa on 176*208 pikselin aktiivimatriisinäyttö, joka pystyy näyttämään 65 536 väriä [Nokia, 2003]. Tällaisella näytöllä varustetussa puhelimessa ei luulisi olevan enää mitään pelien toimivuuteen liittyviä ongelmia. Mikä tahansa Commodore 64 -peli ja vielä huomattavasti sitäkin kehittyneempi peli näyttäisi kauniilta, kun näytön ominaisuudet ovat Nokian 6600 -sarjan tasoa. Näytön ominaisuuksien hyvyydestä on todisteena esimerkiksi puhelimen mukana tuleva flipperipeli, jonka grafiikka on erittäin näyttävää. Pelaamiskokemukseen Nokian 6600 -puhelimella mielestäni negatiivisella tavalla vaikuttavia asioita aiheuttaa kuitenkin luvussa 2.2 käsitellyt näppäimistöön liittyvä seikat, sekä seuraavassa luvussa käsiteltävät, puhelimen tehokkuuteen liittyvät asiat.

Myös näytön malli on pelaamiseen vaikuttava seikka. Perinteisesti videopelejä on pelattu televisioruudun mallisella ruudulla (televisio tai näyttöpäätte), joka on siis isompi leveys- kuin korkeussuunnassa. Viimeisin trendi televisioruuduissa on ollut suurentaa leveyden ja korkeuden eroa vielä entisestään. Laajakuvatelevisioiden yleistymistä on perusteltu muun muassa sillä, että ne ovat luonnollisempia ihmisen silmien katselualueelle [digitv.fi, 2003]. Matkapuhelin on yleensä huomattavasti pidempi pystysuunnassa kuin leveysuunnassa, eli se on kapulan mallinen. Tästä johtuen myös näytölle on enemmän tilaa pystysuunnassa kuin leveysuunnassa. Monissa puhelimissa näyttö onkin kasvanut enemmän pysty- kuin leveysuuntaan: tämän voi havaita tutkimalla esimerkiksi Nokian 40 uusimman puhelinmallin näyttöjen kokoja [Nokia, 2003]. Pystymalliset näytöt aiheuttavat ongelmia jo pelien valmistusvaiheessa. Vaakatasoon suunniteltua peliä ei ole helppoa kääntää pystytasoon [Alatalo, 2003]. Lisäksi pelien näkymä sivusuuntiin kärsii, mutta sivusuunnat ovat kuitenkin yleensä tärkeämpiä useimmissa pelityypeissä. Jos puhelin käännettäisiin vaakatasoon, näyttökin voisi olla saman mallinen kuin perinteiset näyttöruudut, jolloin näkymäala sivusuuntiin kasvaisi ja pelien kääntäminen laitteelle helpottuisi. Näin ei ole kuitenkaan tehty esimerkiksi Nokia N-Gage laitteessa, vaikka se olisi ollut mahdollista. Tämä on ollut yksi asia mistä N-Gage on saanut laajasti negatiivista palautetta [mm. Skenegroup, 2003].

2.4. Suorituskyky

Matkapuhelimet ovat edelleen pääasiassa puheluita yhdistäviä laitteita, ja iso osa puhelinten muusta toiminnallisuudesta liittyy tähän ominaisuuteen läheisesti. Nimiluettelosta käyttäjä hakee nopeasti tavoittelemansa henkilön numeron, puhelutiedoista näkee soittoaikoja ja niin edelleen. Tekstiviestitkin

sisältävät vain lyhyitä asioita, joiden kertomiseen käytetään puheluja yhdistävää tekniikkaa - puhumisen sijasta asia kuitenkin kirjoitetaan. Matkapuhelimeen on kuitenkin jo kauan sisällytetty enemmän ja vähemmän tärkeitä lisäominaisuuksia, jotka sopivat matkapuhelimen mobiililuonteeseen. Kello ja kalenteri ovat integroituneet matkapuhelimiin melko sulavasti. Tämän tyyppisillä sovelluksilla on yksi yhteinen seikka: ne eivät vaadi puhelimelta juuri yhtään laskentatehoa. Toisin on pelien kanssa, varsinkin kun nykyään on käytössä myös värit, monikanavaiset äänet sekä entistä tarkemmat näyttötarkkuudet [Alatalo, 2003].

Vielä tähän mennessä matkapuhelinten yhteydessä ei olla juurikaan mainittu suorituskykyyn liittyviä ominaisuuksia [Amaro and Martins, 2003]. Tietokone- ja pelikonsolimarkkinoilla suorituskykyyn liittyvät ominaisuudet ovat erittäin tärkeitä tuotteiden markkinoinnissa, esimerkiksi Microsoft X-Boxin lehtimainoksessa laitteen suorittimen megahertsimäärä mainitaan ensimmäisenä ominaisuutena. Periaatteessa totuus on se, että mitä tehokkaampi laite, sitä paremmin pelit toimivat. Mielenkiintoista on, ettei edes Nokia N-Gagen sivuilla mainita missään kohdassa, miten tehokas suoritin tai grafiikkapiiri siinä on. Erikoista sinänsä, N-Gage on kuitenkin mobiilipelikonsoli. Suorituskyky tulee kuitenkin heti vastaan, kun pelit alkavat olla vähänkin monimutkaisia. Muisti ja laskentateho ovat siis tärkeitä, huomioonotettavia seikkoja matkapuhelinten peliohjelmoinnissa [Maio, 2002].

Nokia 3510i:n mukana tuleva laskuvarjohyppäystä matkiva peli kärsii suorituskykyongelmista, vaikka sen grafiikka näyttää hyvin alkeelliselta. Laskuvarjohyppypelissä puhelin joutuu kuitenkin pyörittämään ruudulla muutamaa värillistä tekstuuria, ja se vaatii yllättävän paljon laskentatehoa. Nokia 6600:n flipperipelissä ruudunpäivitys kärsii aina, kun puhelin tekee taustalla jotain muuta. Puhelimessa on käyttöjärjestelmänä Symbian OS, jossa on moniajo-ominaisuuksia [Alatalo, 2003]. Kokeilemani N-Gagen Tomb Raider peli tuntui välillä pyörivät laitteen suorituskyvyn ääri rajoilla, mutta tästä en ole löytänyt tarkempia testituloksia. Tomb Raiderissa on joka tapauksessa runsaasti grafiikkaa ja vaativaa 3D-laskentaa. Suorituskyky vaikuttaa suoraan pelaamiskokemuksen nautittavuuteen ja sen puutteellisuus on käytettävyysongelma pelaajan näkökulmasta.

Puhelimen suorituskyvyn puute pelattaessa voi vaikuttaa myös näppäimistön toimintaan. Jos pelin ruudunpäivitys alkaa olla alle 20 kuvaa sekunnissa, vaikuttaa se myös näppäinpainallusten reagointinopeuteen ja pelin ohjaaminen tuntuu hitaalta ja kömpelöltä [Matthews, 2002].

Suorituskyvystä tai sen mittaamisesta ei ainakaan vielä ole tullut mainostettavaa ominaisuutta matkapuhelimiin [Amaro and Martins, 2003].

Toisaalta se on hyvä asia, koska varsinkin PC-maailmassa suorituskyvyllä ratsastaminen on saanut jopa harhaanjohtavia piirteitä. Esimerkiksi AMD kutsuu suorittimiaan tietyn numerosarjan mukaisesti. Tästä numerosarjasta saa helposti käsityksen, että se tarkoittaa suorittimen megahertsimäärää, vaikka tosiasiallisesti luku on huomattavasti megahertsimäärää suurempi. Suorittimien malleja on perinteisesti kutsuttu niiden megahertsimäärän mukaan. Esimerkiksi Intel Pentium 4 2,53 on 2,53 GHz:n suoritin, kun taas AMD Athlon Xp 2500+ on 1,8 GHz:n suoritin. On mielenkiintoista, ettei suorituskykymainonta ole rantaunut vielä matkapuhelimien maailmaan. Toisaalta voisi kuvitella suorituskykymainonnan nopeuttavan matkapuhelinten suorituskyvyn kasvua. Laitevalmistajien välinen kilpailu pakottaisi ne nostamaan puhelinten suorituskykyä [Amaro and Martins, 2003].

3. Ongelmien ratkaisua

Tässä luvussa esittelen ratkaisuja luvussa 2 mainittuihin ongelmiin. Ratkaisut perustuvat pääosin omiin päätelmiini. Pidän ratkaisut kuitenkin varsin matalalla tasolla, enkä tarjoa moneenkaan ongelmaan täydellistä vastausta. Näitä ratkaisuja noudattamalla olisi kuitenkin mielestäni mahdollista valmistaa hyvin pelaamiseen soveltuva matkapuhelin.

3.1. Näppäimistö

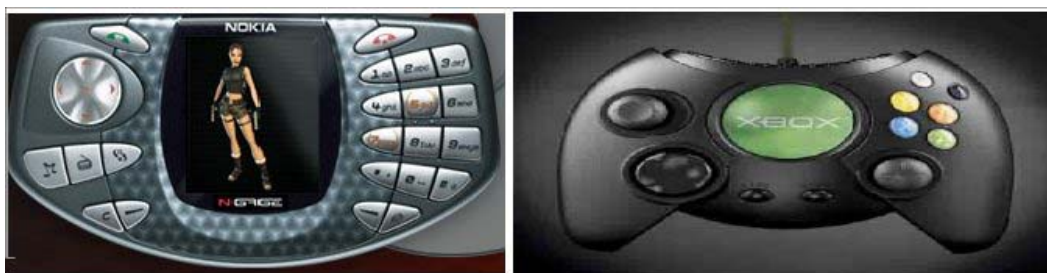
Yhtäaikaisten painallusten toimimattomuus on suuri rajoite jo suunniteltaessa peliä matkapuhelimelle - puhumattakaan siitä, miten paljon se hankaloittaa pelaamista. Esimerkiksi Nokia 6600 -mallissa tämä ongelma on jo otettu huomioon. Symbianiin perustuva käyttöjärjestelmä ja 6600:n laitealusta mahdollistavat yhtäaikaiset näppäinpainallukset [Alatalo, 2003]. Useat suoraan puhelimeen ohjelmoidut käyttöliittymät eivät kuitenkaan vielä nykyäänkään mahdollista yhtäaikaisia näppäinpainalluksia. Tällaisia ovat esimerkiksi puhelinten Nokia 3510i sekä Siemens M55 (molemmat puhelinmallit on julkaistu vuonna 2003) käyttöliittymät. Molempia malleja markkinoidaan muiden ominaisuuksien lisäksi myös hyvillä peliominaisuuksilla, ja molempiin puhelimiin on saatavilla ladattavia Java-pelejä.

Yhtäaikaiset näppäinpainallukset ovat tärkeä ominaisuus. Hyvä esimerkki on auto (tai autopeli). Ajokokemus kärsisi kovasti, mikäli jarrua ei voisi painaa samanaikaisesti kun rattia käännetään. Tämä ongelma saattaa jäädä historiaan jo lähitulevaisuudessa. Kuten yllä mainitsin, esimerkiksi Symbianiin perustuvissa puhelimissa yhtäaikaiset näppäinpainallukset toimivat.

Näppäinten yleistöiminta on myöskin ongelma. Näppäinten täytyy reagoida nopeasti jokaiseen painallukseen. Näin ei kuitenkaan ole kaikissa

testaamissani matkapuhelimissa. Tappihiiren tuleminen muutamiin puhelimiin on erinomainen asia pelattavuuden kannalta. Yhdellä sormella pystyy ohjaamaan helposti jokaiseen suuntaan, ja yksi toiminto löytyy myös nappia painamalla. Toistaiseksi en ole vielä törmännyt matkapuhelimeen, jossa olisi kaikkiin ilmansuuntiin toimiva tappihiiri. Pelaamisen kannalta kahdeksaan suuntaan toimiva ohjain olisi tietenkin parempi kuin neljään suuntaan toimiva ohjain. Esimerkiksi Nokia 6600:ssa tappihiiren viistottaiset liikesuunnat on estetty mekaanisesti. Syitä tähän on vaikea arvata, sillä alusta mahdollistaisi tappihiiren toiminnan kaikkiin ilmansuuntiin [Alatalo, 2003].

Näppäinten sijoittelu on seikka, mikä tulisi suunnitella tarkoin ottaen huomioon matkapuhelimen käyttötarkoitus. Jos puhelimen suunnittelussa lähtökohtana on tehdä perinteinen matkapuhelin, nykyinen tyypillinen näppäinten järjestys on varmasti järkevin: näppäinten paikat ovat tuttuja, numeroiden valinta käy nopeasti ja tekstiviestien kirjoittaminen on helppoa. Jos matkapuhelimesta halutaan tehdä niin sanottu pelipuhelin, on näppäinten sijoitteluun kiinnitettävä enemmän huomiota. Tilanne on helpompi, jos tavoitteena on tehdä pääasiassa pelaamiseen tarkoitettu puhelin. Mallia voidaan hakea tietokone- ja videopelien maailmasta, niiden ohjaimista. Pad-tyyppinen ohjain on suurin piirtein isohkon matkapuhelimen kokoinen, ja ohjaimissa on useita variaatioita siitä, miten näppäimet kannattaisi sijoittaa. Näitä variaatioita ohjaimia valmistavat yritykset ovat testanneet ja kehittäneet paljon [Cuciz, 2002]. Myös matkapuhelinvalmistajat hyötyvät niistä. Esimerkiksi: näyttö keskelle pad-ohjainta ja näppäimet näytön molemmin puolin. Tähän on selvästikin pyritty Nokia N-Gagessa.



Kuva 1: Nokia N-Gage ja Microsoft Xboxin pad-ohjain.

Tilanne on hankalampi, jos matkapuhelin suunnitellaan pääasiassa puhelintoimintoja varten ja se halutaan pitää perinteisen näköisenä matkapuhelimenä, mutta pelitoimintojen tulisi olla mahdollisimman hyviä. Näppäinten sijoittelu aiheuttaa kompromisratkaisuja heti alkutilanteessa. Jos matkapuhelin on kapulamallinen laite (ilman liikkuvia osia, näyttö yläreunassa ja näppäimet alareunassa), puhelin tulisi suunnitella tarpeeksi leveäksi, että

molemmille käsille on tilaa. Tappihiiri on mielestäni erinomainen valinta ohjainlaitteeksi: se säästää tilaa ja yhdellä sormella yhdestä painikkeesta voidaan ohjata kaikkiin suuntiin. Nokia 6600 on peliominaisuuksiltaan varsin onnistunut laite, vaikka sen suunnittelun lähtökohtana on ollut tehdä ammattilaisen työkalu eikä pelikone [Nokia, 2003]. Tarpeeksi leveä ote, tappihiiri ja isot näppäimet tekevät puhelimella pelaamisesta melko nautittavaa (näppäimistön osalta). Tappihiiren toimiminen vain neljään suuntaan tosin rajoittaa pelaamista. Oliver Maion [Maio, 2002] mukaan Nokia 3650:n näppäimistö soveltuu pelaamiseen hyvin. 3650:ssa näyttö on puhelimen yläosassa perinteiseen tapaan, mutta näppäimistö on normaalista poikkeavalla tavalla ympyrän mallisesti sijoitettu laitteen alaosaan.

Toinen suosittu matkapuhelimen malli on niin sanottu simpukkamalli, jossa näyttö käännetään näppäimistön päältä pystyyn. Tällaisia puhelinmalleja

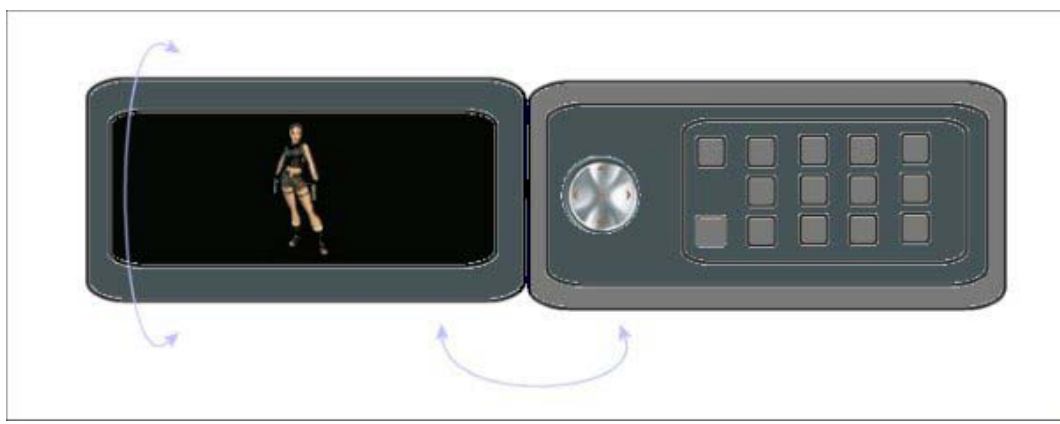


Kuva 2: Motorola T720i ja Nokia 3650

on esimerkiksi Motorola tehnyt useita. Simpukkamalli antaa suunnittelijan mielikuvitukselle enemmän tilaa. Kun näyttö käännetään auki, näppäimistön voisi suunnitella niin, että puhelinta pidetään poikittaisessa asennossa pelatessa. Tässä tapauksessa usein varsin iso näyttökin olisi laajakuvaa-asennossa ja koko näppäimistö kahden käden ulottuvilla - hiukan pad-tyylisesti - huomattavasti paremmin kuin kapulamallisessa puhelimessa. Periaatteessa sama onnistuisi myös kapulamallisella matkapuhelimella. Jos kapulamallisen puhelimen näyttö on iso (esimerkiksi Nokia 6600), se vie suuren osan puhelimen pinta-alasta ja näppäimistö jää varsin alas. Tällöin puhelimen kääntäminen poikittain ei onnistu. Auki ollessaan simpukkamallin näyttö vie alle puolet puhelimen pinta-alasta ja lisäksi se antaa hiukan joustovaraa pelaajan käsille. Pelaaja voi taivuttaa näyttöä hiukan, jolloin vasemman käden peukalo jää puhelimen näppäimistölle ja muut vasemman käden sormet jäävät näytön alle. Nykyään olisi mahdollista rakentaa näyttö niinkin, että se kääntyisi myös Y-akselinsa ympäri. Tällöin joustovaraa olisi entistä enemmän.

Simpukkamallinen puhelin, jossa on tappihiiri tai ristikko-ohjain näppäimistön yläosassa ja mahdollisimman iso näppäimistö sen alla, voisi mielestäni olla toimiva yleisratkaisu pelaamiseen tarkoitettuun matkapuhelimeen.

Kaikissa tapauksissa matkapuhelinvalmistajien tulisi ottaa huomioon näppäinten tuntuma. Näppäimien tulisi olla melko isoja, tarpeeksi erillään toisistaan, tarpeeksi kohollaan puhelimen rungosta ja painalluksen tulisi tuntea sormessa selkeästi. Tällainen näppäintuntuma on esimerkiksi Nokia 3510i:ssä. Suorituskykyä puhelimesta tulisi löytyä niin paljon, että puhelin reagoi näppäinpainalluksiin riittävän nopeasti, tai pelit tulisi optimoida niin, että pienikin laskentateho riittää [Matthews, 2002].



Kuva 3: Pelipuhelin perinteisessä simpukkamallisessa rakenteessa

3.2. Näyttö

Näytöt ovat kehittyneet viime vuosina paljon. Näyttöjen koko on kasvanut huomattavasti, ja värinäytöt ovat nykyään yleisiä. Esimerkiksi Nokian 40 uusimmasta puhelinmallista 28:ssa on värinäyttö [Nokia, 2003]. Peleihin liittyvät ongelmat ovat vähentyneet näyttöjen osalta lähes automaattisesti. Pääasiassa ainoat ongelmat nykyaikaisissa puhelimissa ovat näytön malli ja pikseleiden syttymis- ja sammumisnopeus [Matthews, 2002].

Pelien kääntäminen muilta laitteilta matkapuhelimille vaikeutuu, jos näyttö on pystymallinen. Parhaimmillaan näyttö olisi samassa suhteessa televisioruudun kanssa, eli 4:3 tai 16:9. Tällöin peli olisi näytön puolesta helppo siirtää matkapuhelimeen [Alatalo, 2003]. Jos näyttö on pystyasennossa, pelin sivuttaissuunnille ei jää tarpeeksi tilaa. Esimerkiksi Nokia N-Gagen näyttöä on kritisoitu siitä, että pelit, jotka on alun perin suunniteltu vaakatasossa olevalle näytölle, kärsivät N-Gagen pystyasennossa olevasta näytöstä (esimerkiksi Sonic N, Tomb Raider). Näkymäala sivusuuntiin jää liian pieneksi [mm. Alanen, 2003]. Uusimmissa matkapuhelimissa näytöt ovat niin sanottuja aktiivimatriisinäyttöjä, joissa pikseleiden syttymis- ja sammumisnopeus on

hyvällä tasolla. Tämä takaa sen, ettei pelaamisesta tule epäselvää, vaikka ruudulla olisi samanaikaisesti paljon liikkuvaa grafiikkaa [Alatalo, 2003].

Näyttöjen aiheuttamat käytettävyysongelmat ovat kaikkein helpointa ratkaista. Näytön pitää olla tarpeeksi iso täysvärinäyttö. Sen pitäisi olla oikean mallinen ja sen tulisi olla aktiivimatriisinäyttö. Nämä ominaisuudet ovat kameroiden ja multimediatekniikoiden myötä yleistyneet. Useissa malleissa on jo nyt vähintäänkin riittävän hyvä näyttö pelaamista varten. Esimerkiksi Nokia 6600:n iso täysvärinäyttö sopii mielestäni pelaamiseen erittäin hyvin, mutta sekin on pystyasentoon suunniteltu ja kuten edellä mainitsin, muutamat näppäimistöön ja suorituskykyyn liittyvät ongelmat huonontavat pelaamiskokemusta 6600:lla.

3.3. Suorituskyky

Vasta viime vuosina matkapuhelimiin on tullut sovelluksia, jotka vaativat parempaa suorituskykyä. Suorituskykyä vaativat pääasiassa digitaalisen kameran tarvitsemat toiminnot, moniajavat laitealustat sekä pelit [Alatalo, 2003]. Esimerkiksi Nokia 6600:n digitaalinen kamera ottaa värikuvaa puhelimen näytölle noin 23 kertaa sekunnissa [Nokia, 2003]. Kuvan piirtäminen ja vaihtaminen näytölle niinkin usein vaatii matkapuhelimelta paljon laskentatehoa. Uusimpien puhelimien käyttöjärjestelmät, esimerkiksi Symbian, ovat moniajavia järjestelmiä. Isolla värinäytöllä pyörivät ohjelmat ja taustatoiminnot vaativat myöskin entistä enemmän laskentatehoa. Pelit, jotka käyttävät matkapuhelimien kasvaneita näyttöjä hyväkseen, tarvitsevat paljon laskentatehoa. Mitä monimutkaisempaa grafiikka pelissä on, sitä enemmän laskentatehoa se tarvitsee. Esimerkiksi N-Gagen Tomb Raider -pelissä puhelimen näytöllä pyörii samanaikaisesti useita kolmiulotteisia objekteja ja värillisiä tekstuureita: nämä vaativat jo varsin paljon laskentatehoa [Alatalo, 2003].

Suorituskykyä voidaan optimoida sillä, että matkapuhelin tukee tiettyä teknologiaa suoraan laitteistopohjaisesti [Alatalo, 2003]. Myös erilaiset rajapinnat grafiikan piirtämistä varten helpottavat suorittimen laskentataakkaa [Kontio, 2003]. Uusimmat ohjelmointitekniikat tehostavat ja helpottavat peliohjelmoijan työtaakkaa. Esimerkiksi grafiikan piirtäminen on helpompaa ja tehokkaampaa Javan MIDP 2.0 määrittelyn ilmestyttyä kuin aikaisemmissa Java-versioissa [Forum Nokia, 2003].

Jos suorituskykyä ei ole riittävästi, siitä tulee käytettävyysongelma. Pelikokemus kärsii, matkapuhelin saattaa lukkiutua, jokin toiminto saattaa toimia väärin ja niin edelleen. Samat ongelmat ovat erittäin tuttuja pc-peleistä. Esimerkiksi Nokia 6600-puhelimen mukana tuleva flipperipeli nykii lähes aina, kun puhelin moniajaa mitä tahansa taustalla. Jos flipperiä pelattaessa on puolen

sekunnin viiveitä satunnaisesti, peli muuttuu varsin vaikeaksi. Suuri laskentateho ja muisti sekä optimoitu ohjelmisto tekevät matkapuhelimella pelaamisen toimivaksi [Maio, 2002]. Nykyisissä puhelimissa on jo niin paljon laskentatehoa vaativia pelejä, että matkapuhelinvalmistajien on kiinnitettävä entistä enemmän huomiota puhelimen suorituskykyyn.

Matkapuhelimien suorituskykyä ei voida kasvattaa kuitenkaan samalla tavoin kuin tietokoneissa tai pelikonsoleissa. Kun laskentatehoa kasvatetaan, laite tarvitsee enemmän virtaa ja se tuottaa enemmän lämpöä [Pietikäinen, 2001]. Melko suoraan voidaan todeta, että mitä vaativampaa grafiikkaa laitteen tulee tuottaa, sitä enemmän laskentatehoa se tarvitsee. Luonnollisesti myös ohjelmakoodia optimoimalla suorituskyvystä saadaan enemmän irti [Matthews, 2002]. Nykyisten matkapuhelinten laskentateho on melko hyvällä tasolla, esimerkiksi N-Gage pystyy pyörittämään varsin näyttävää grafiikkaa. Kuitenkin pelatessa N-Gagen akku loppuu muutamassa tunnissa [Nokia, 2003]. Matkapuhelinvalmistajien onkin syytä kehittää sekä ohjelmisto- että laitteistoteknologioitaan niin, että suorituskyky on mahdollisimman suuri, mutta puhelimen käyttömukavuus ei kärsi.

4. Muita ongelmakohtia ja tulevaisuudennäkymiä

Edellä käsittelemäni näppäimistö, näyttö ja suorituskyky ovat kolme pääkohtaa, jotka aiheuttavat ongelmia puhelimella pelatessa. Käsittelemäni muita, pienempiä ongelmakohtia lyhyesti. Käsittelemäni tässä kappaleessa myös matkapuhelinpelaamisen tulevaisuudennäkymiä.

4.1. Äänet, ohjelmisto ja laitteen suunnittelu

Äänet ovat tärkeä osa pelikokemusta ja viime vuosina tietokone- ja videopelien yhteydessä pelaaja on päässyt nauttimaan entistäkin laadukkaimmista äänistä [Wilde, 2003]. Matkapuhelin on kuitenkin pieni laite, eikä siihen pystytä sisällyttämään kovinkaan hienoja äänentoistolaitteita. Monikanavainen kaiutin ajaa kuitenkin asian välttävästi, paremmat äänet saadaan kuulumaan korvakuulokkeiden kautta [Skenegroup, 2003]. Kuulokkeiden käyttömahdollisuus on tärkeä seikka, jos matkapuhelimen yksi käyttötarkoitus on pelaaminen. Niiden kautta saadaan aikaiseksi laadukkaat äänet ja ne tekevät matkapuhelimella pelaamisesta ihmisjoukon seassa hyväksyttävämpää, koska kukaan ei häiriinny. Erilaisia teknologioita äänentuotantoon mobiililaitteissa on useita [Wilde, 2003]. Äänen toistaminen kuluttaa laitteen resursseja, ja varsinkin pelien yhteydessä tulisi pelikehittäjän pyrkiä käyttämään tehokkainta äänen pakkausmenetelmää ja teknologiaa.

Oli pelilaitte miten hyvä tahansa, on sille oltava laadukkaat ohjelmistot aina käyttöjärjestelmästä peleihin. Ohjelmistojen on pysyttävä pystyssä, ja niiden tulee olla helposti käytettäviä. Lukkiutumista, kaatumista ja laitteen sekoamista on vältettävä jo ohjelmistojen suunnitteluvaiheessa. Käyttäjäystävällinen järjestelmä antaa laadukkaan kuvan laitteesta. Mahdollisimman hyvään tulokseen päästään laajalla ja perinpohjaisella testauksella ennen laitteen tuomista markkinoille [Lipponen, 2001].

Matkapuhelin tulisi suunnitella niin, että sen jokapäiväinen yleiskäyttö on helppoa ja mukavaa. Nokian puhelimia on pidetty erinomaisina tässä suhteessa [Lindholm *et al.*, 2003]. Nokia N-Gagen yleissuunnittelussa on kuitenkin muutama erikoinen piirre, joita on kritisoitu laajasti. Ensinnäkin, puhelimella puhutaan niin, että sitä pidetään puhelimen kylki korvaa vasten. Sen on useassa lähteessä todettu olevan epämiellyttävää ja epäkäytännöllistä [mm. Skenegrupp, 2003]. Toinen ongelma on tähän tutkielmaan lähemmin liittyvä käytettävyysoongelma: jos N-Gageen halutaan vaihtaa peli, joudutaan irrottamaan sekä takakansi että puhelimen akku, ennen kuin päästään käsiksi pelimoduliin. Tällaiset seikat eivät välttämättä tee puhelimen käyttöä helpoksi ja mukavaksi.

Matkapuhelimen ulkoinen olemus ja käyttäjän persoonallisuuteen liittyvät seikat on myös tärkeää ottaa huomioon nykyaikasta matkapuhelinta suunniteltaessa. Persoonallinen muotoilu ja ulkoasu, sekä käyttöliittymän personoitavuus (taustakuvat, animaatiot, värit, soittoäänät ja niin edelleen) ovat osaltaan vaikuttamassa matkapuhelinten myyntilukuihin [Swantenson, 2003].

4.2. Sisällöntuotanto

Sisällöntuotannolliset seikat eivät ole varsinaisesti käytettävyysoongelmia, mutta ne ovat läheisesti sidoksissa tutkielman aiheeseen: mitä tahansa peliä ei voida kääntää tai toteuttaa matkapuhelimelle. Kuten todettua, muun muassa pieni näyttö, puutteellinen suorituskyky ja paljon normaalista pelilaitteesta poikkeava käyttöliittymä asettavat rajoituksia. Jo pelin suunnitteluvaiheessa olisi mietittävä tarkoin soveltuuko kyseinen peli lainkaan matkapuhelimelle. Esimerkiksi alunperin televisioruudulle suunniteltua Sonic-tasohyppelyä on useassa yhteydessä pidetty liian pienenä ja vauhdikkaana pelinä matkapuhelimelle [mm. Alanen, 2003].

Saman ongelman voi kääntää myös toisin päin. Osataanko matkapuhelimen tarjoamia mahdollisuuksia käyttää hyväksi? Mobiili ja verkottunut pelialusta tarjoaa ennennäkemättömiä mahdollisuuksia toteuttaa aivan uudenlaisia pelejä [Parikka, 2003]. Esimerkiksi N-Gage tarjoaa mahdollisuuden moninpeleihin paikasta ja ajastakin riippumatta [Salminen, 2003]. Nähtäväksi jää, osaavatko sisällöntuottajat hyödyntää näitä ominaisuuksia. Japanissa matkapuhelimilla

pelaaminen on jo erittäin suosittua, vanhat klassikkopelit, useat erilaiset matkapuhelinverkkoja hyödyntävät moninpelit ja kymmenet erilaiset yksinpelikonseptit ovat luoneet isot mobiilipelimarkkinat Japaniin [Collier, 2003].



Kuva 4: Japanilainen matkapuhelinpelimainos

Sisällöntuotannon rooli on ratkaiseva tekijä laitteen myynnissä. Ilman hittituotteita laite ei menesty. Pelilaitteen markkinatilanne parantuu oleellisesti, jos sille on olemassa hittituote, jota ei ole olemassa millekään muulle laitteelle. Pelimaailmassa tämä on osoittautunut todeksi useaan otteeseen [Salminen, 2003].

Perinteisesti miehet ovat pelanneet erilaisia elektronisia pelejä enemmän kuin naiset, mutta tämä seikka yritetään muuttaa matkapuhelinpelien yleistyessä. Isossa-Britanniassa matkapuhelimen omistaa 75% väestöstä. Vuoteen 2006 mennessä matkapuhelinpelien markkinoiden uskotaan olevan 211 miljoonaa puntaa Isossa-Britanniassa. Pelien kehittäjät pitävät tärkeänä sitä, että myös naiset ovat näissä markkinoissa osallisena. [Twist, 2003]

4.3. Tulevaisuudennäkymiä

Matkapuhelinpelaamisen ennustetaan kasvavan todella paljon lähi-tulevaisuudessa. Matkapuhelinvalmistajat ovat tiedostaneet tämän ja alkaneet panostaa tuotteidensa peliominaisuuksiin. Nokia N-Gagen myynti on lähtenyt nihkeästi liikkeelle, mutta vasta tuleva vuosi näyttää miten N-Gagen käy. Nokialla on varsin kovat odotukset laitetta kohtaan [The Register, 2003].

Uusia, pelaamiseen entistä paremmin sopivampia laitteita, ilmestyy markkinoille tihenevään tahtiin. Nokia julkaisee ensi vuoden toisella neljänneksellä mallin 7700, mobiilimedialaitteen. Siinä yhdistyy periaatteessa kaikki tämän tutkielman pelaamisen parantamiseksi tarjoamat ratkaisut. Iso vaakatasossa oleva näyttö, paljon suorituskykyä ja pad-tyyppinen ohjaus.

Äänet ovat mobiililaitteeksi ensiluokkaiset ja ohjelmistot ovat erittäin kehittyneitä. [Nokia, 2003]



Kuva 5: Nokia 7700

Monet muutkin valmistajat ovat pyrkimässä samoille markkinoille. Myös ohjelmistojätti Microsoft. Microsoftin käyttöjärjestelmään perustuvia Pocket PC Phone Edition ja Smartphone laitteita on saatavilla jo useita, ja niiden multimedia- ja peliominaisuuksiin on panostettu paljon [Microsoft, 2003]. Lähitulevaisuuden matkapuhelimet muistuttavat entistä vähemmän perinteisiä matkapuhelimia: isot näytöt, vaihtoehtoiset ohjainlaitteet, multimedia ja mobiilipelit ovat tulevaisuutta.

5. Yhteenveto

Matkapuhelinpelaamisen tulavaisuus näyttää siis valoisalta. Tällä hetkellä mobiilipelaaminen painottuu vielä mobiilipelikonsoleihin, ei matkapuhelmiin. Suurin osa mobiilipeleistä valmistetaan ja myydään Nintendo GameBoyle [Amaro, 2003]. Nintendon tavoitteena on myydä 20 miljoonaa GameBoy Advance -mobiilipelikonsolia meneillään olevalla tilikaudellaan, Nokian pyrkimys on myydä 6 - 9 miljoonaa N-Gagea 2004 vuoden loppuun mennessä [The Register, 2003]. Todennäköinen myyntimenestys olisi sellainen matkapuhelin, jossa kaikki tässä tutkielmassa käsitellyt käytettävyystekijät olisivat kunnossa, hinta olisi alhainen ja olisi olemassa edes yksi hittipeli, jota ei saisi millekään muulle laitteelle. N-Gage ei ole huono laite, mutta siinä on muutamia käytettävyyso ongelmia jotka saattavat pahimmillaan koitua laitteen kohtaloksi. Sekin pahentaa tilannetta, että kaikki N-Gagen pelit ovat vanhoja käännöksiä muilta laitteilta, omaa hittipeliä ei ole [Salminen, 2003].

Matkapuhelinten yhdistäminen mobiilipelaamiseen luo uusia mahdollisuuksia, koko pelaamiseen liittyvä kulttuuri voi kehittyä. Matkapuhelinten avulla voidaan toteuttaa verkottunut virtuaalipelimaailma, joka on pelaajien ulottuvilla koko ajan. Käytettävyys on iso tekijä kun seuraavan sukupolven mobiililaitteet yleistyvät, käytettävyystekijät pelaajan näkökulmasta on otettava tarkasti huomioon, jos

matkapuhelinvalmistaja haluaa tuotteensa menestyvän. Jatkan tämän aiheen tutkimista myös pro gradu -tutkielmassani.

Lähdeluettelo

- [Alanen, 2003] Ninnu Alanen, N-Gagen ensimmäinen peliaalto, *MikroBitti*, 12 / 2003, 124-125.
- [Alatalo, 2003] Tero Alatalo, henkilökohtainen tiedonanto, aiheina: *Uusien matkapuhelinteknologioiden suorituskyky. Pelien porttaaminen erimallisille näytöille*. 10.12.2003
- [Amaro, 2003] Pedro Henrique Simões Amaro, *The Clash of Mobile Platforms: J2ME, ExEn, Mophun and WGE*
(http://midlet-review.com/?content=j2me_exen_mophun_wge1)
(tarkistettu 24.10.2003)
- [Amaro and Martins, 2003] Pedro Henrique Simões Amaro, João David Dias Martins, *Mobile Benchmarking - Methods and Importance*
(http://midlet-review.com/?content=mobile_benchmarking)
(tarkistettu 24.10.2003)
- [Barbagallo, 2002] Ralph Barbagallo, *Games on the Run: BREW and J2ME*
(http://www.gamasutra.com/resource_guide/20021125/barbagallo_pfv.htm) (tarkistettu 19.10.2003)
- [C64.com, 2003] C64.com, (<http://www.c64.com>) (tarkistettu: 12.12.2003)
- [Collier, 2003] David Collier, *Mobile Games: Trends from Japan*
(http://www.gamasutra.com/gdce2003/david_collier.pdf)
(tarkistettu 30.10.2003)
- [Digitv.fi,2003] Digitv.fi, Ohjelmien kuvasuhde
(<http://www.digitv.fi/sivu.asp?path=1;2997;952;3097>)
(tarkistettu: 10.12.2003)
- [Forum Nokia, 2003] Forum Nokia, *MIDP 2.0: Working with Pixels and drawRGB*
(http://ncsp.forum.nokia.com/downloads/nokia/documents/Working_witrh_pixels_and_drawRGB_v1_0.pdf) (tarkistettu 26.10.2003)
- [Cuciz, 2002] David Cuciz, *Total Control - A History of Game Controllers*,
(<http://www.gamespy.com/hardware/june02/sticks1/>) (tarkistettu: 1.12.2003)
- [Järvinen, 2002] Aki Järvinen, *Milloin pelistä tulee mobiili?*
(<http://www.m-cult.net/mediumi/article.html?articleId=29&print=1>)
(tarkistettu 19.10.2003)
- [Junkkari, 2003] Marko Junkkari, Suomen kansantalouden toivot, *Helsingin Sanomat*, 26.10.2003, E1.
- [Kontio, 2003] Mikko Kontio, *MIDP 2.0: The Game API*
(http://www.microjava.com/articles/techtalk/game_api)
(tarkistettu 24.10.2003)

- [Lindholm *et al.*, 2003] Christian Lindholm, Turkka Keinonen, Harri Kiljander, *Mobile Usability: How Nokia Changed the Face of the Mobile Phone*. McGraw-Hill Companies, 2003.
- [Lipponen, 2001] Outi Lipponen, *Testaus – välttämätön pakko vai kilpailutekijä?* (http://www.uku.fi/kkk/it/laatu2001/materiaali/lipponen_testaus.ppt) (tarkistettu 10.12.2003)
- [Maio, 2002] Oliver Maio, *Developing for Two Phone Extremes – Comparing the Nokia 6310i and the Nokia 3650* (http://www.gamasutra.com/resource_guide/20021125/maio_pfv.htm) (tarkistettu 28.10.2003)
- [Matthews, 2002] Marcus Matthews, *Developing Action Based Mobile Games* (http://www.gamasutra.com/resource_guide/20021125/matthews_pfv.htm) (tarkistettu 28.10.2003)
- [Microsoft, 2003] Microsoft Windows mobile, (<http://www.microsoft.com/windowsmobile/default.msp>) (tarkistettu: 12.12.2003)
- [Mind Trek, 2003] *Painopisteenä mobiilit pelit –seminaari*, Tampere, 12.11.2003
- [Nokia, 2003] *Nokia – Puhelimet – Puhelinmallit* (<http://www.nokia.fi/puhelimet/puhelinmallit/>) (tarkistettu 23.11.2003)
- [Parikka, 2003] Jussi Parikka, *Liike, paikka, havainto – mobiili kuvaruutu ja pelaamisen uudet ulottuvuudet* (http://www.film-o-holic.com/widerscreen/2003/2-3/mobiili_kuvaruutu_ja_pelaamisen_ulottuvuudet.htm) (tarkistettu 24.10.2003)
- [Pietikäinen, 2001] Ville Pietikäinen, Kannettavat laitteet, kannettavien rakenne, tehonhallinta, pakkaustekniikka... (http://tisu.mit.jyu.fi/embedded/TIE345/vanhat_luentokalvot/kannettavat_9.pdf) (tarkistettu 10.12.2003)
- [Press.nokia.fi, 2000] Nokian lehdistötiedotteet, Lokakuu 04, 2000, (http://press.nokia.fi/PR/200010/815969_4.html) (tarkistettu: 13.12.2003)
- [The Register, 2003] The Register, *Nokia figures claims massive N-Gage sales*, 23.10.2003. (<http://www.theregister.co.uk/content/54/33555.html>) (tarkistettu: 7.12.2003)
- [Salminen, 2003] Kari Salminen, *Tulevaisuus pelissä*, *Suomen Kuvalehti*, 48, 28.11.2003, 34-35.
- [Skenegroup.net, 2003] *Nokia N-Gage* (<http://www.skenegroup.net/fi/artikkelit/ngage>) (tarkistettu: 1.12.2003)

- [Swantenson, 2003] Sofia Swantenson, *User Centred Design for Mobile Phone Interfaces*
(<http://www.it.uu.se/edu/course/homepage/idesign/2003/material/mobile.pdf>) (tarkistettu 20.10.2003)
- [Twist, 2003] Jo Twist, *Mobile games to 'tempt women'*
(<http://news.bbc.co.uk/1/hi/technology/3151344.stm>) (tarkistettu 28.10.2003)
- [Wilde, 2003] Martin Wilde, *Audio for Mobile devices*
(http://www.gamasutra.com/resource_guide/20030528/wilde_pfv.htm)
(tarkistettu 26.10.2003)

Tietokone musiikintekijän välineenä

Harri Pora

Tiivistelmä

Tässä tutkielmassa kartoitetaan niitä mahdollisuuksia, joita tietokone ja musiikkiohjelmistot ovat tuoneet musiikintekijöiden käytettäväksi.

Avainsanat ja -sanonnat: tietokone ja musiikki, musiikkiohjelmat, MIDI, musiikin kuvauskielet, äänenkäsittely.

CR-luokat: H.5.5

1. Johdanto

Tutkielman aihepiirinä on musiikin tekeminen tietokoneen avulla. Digitaalitekniikka on muuttanut lyhyessä ajassa niin itse musiikkia kuin sen tekemistään. Suuri osa nykyään kuulemastamme musiikista on tuotettu ainakin osittain tietokoneen myötävaikutuksella. Musiikkituotanto ei ole enää täysin riippuvainen akustisista instrumenteista ja niiden taitavista soittajista. Musiikkia voi tehdä tiettyihin käyttötarkoituksiin jopa ilman soittotaitoa, vaikkapa pelkän tietokoneen ja ohjelmiston avulla. Perinteisten soittimien mallintamisen lisäksi voidaan rakentaa myös täysin uusia instrumentteja tai pelkkiä ääniä, joita voi käyttää ohjelmallisesti musiikin rakennusmateriaalina.

Tietokoneen hyväksikäyttö ei rajoitu pelkästään uuden etsimiseen tai kokeellisen musiikin tuottamiseen. Varsinkin populaarimusiikin alueella konetta käytetään välineenä jo niin taitavasti, että musiikin ammattilainenkaan ei aina kykene valmiista äänitteestä erottelemaan ihmisen ja koneen osuutta toisistaan. Tämä ei kuitenkaan tarkoita sitä, että itse kone olisi välttämättä musikaalinen. Muusikkous on vain laajentunut sisältämään konventionaalisten musiikki-instrumenttien hallitsemisen lisäksi myös tietotekniikan hallintaa. Nykymuusikko voi tuntea tietotekniikan mahdollisuudet ja muokata niitä haluamaansa suuntaan jo yhtä perusteellisesti kuin vaikkapa fagotin soittaja, joka tuntee ruokolehden ominaisuudet äänilähteenä perinpohjaisesti.

Tämän tutkimuksen tarkoituksena on kartoittaa ja pohtia niitä uusia mahdollisuuksia ja tekniikoita, joita musiikintekijöillä tänä päivänä on. Työ on luonteeltaan kirjallisuuskatsaus, mutta osa ajankohtaisemmasta aineistosta on kerätty verkkosivuilta. Myös kirjoittajan omat musiikilliset kokeilut, joissa tietokone on ollut osatekijänä 1980-luvulta alkaen, ovat vaikuttaneet lopputuloksen muotoutumiseen.

Tutkimuskohdetta pyritään tarkastelemaan riittävän etäisyyden päässä teknisistä yksityiskohdista, jotta laaja aihealue saataisiin hallintaan. Aihealueelta löytyy myös runsaasti aikaisempaa kirjallisuutta, mutta se on yleensä keskittynyt johonkin tämän otsikon osa-alueeseen tai tekniseen yksityiskohtaan. Tässä selvityksessä pyritään muodostamaan ajankohtainen kokonaiskuva koko tutkimusalueesta.

Työn otsikkona on *tietokone musiikintekijän välineenä*. Musiikin tekemisellä tarkoitetaan niitä vaiheita, joita sisältyy luomisen alkuvaiheista valmiin teoksen nuotinnukseen tai soivaan esitykseen saakka. Kiinnostavaa on pohtia millaisia rooleja tietokoneella voi olla näissä tekemisen eri vaiheissa. Jos ajatellaan aikaa ennen tietokonetta, niin vaiheisiin tarvittiin yleensä useita tekijöitä. Nykyään on periaatteessa mahdollista hoitaa yksin koko tekemisen prosessi. Musiikin omakustantaminen onkin tullut yhä kiinnostavammaksi sitä mukaa kun tarvittavien tietoteknisten välineiden hintataso on laskenut. Kaupalliseen levitykseen tulevien äänitteiden viimeistelyvaiheissa käytetään kuitenkin yleensä ammattistudioiden monipuolisempia mahdollisuuksia [Oesch, 1998, 52].

Tutkimus on jaettu lukuihin seuraavasti: Toisessa luvussa määritellään tärkeimpiä käsitteitä ja luodaan katsaus historiaan. Kolmannessa luvussa jaotellaan musiikki tekijän näkökulmasta kolmeen erilaiseen olomuotoon. Neljäs luku käsittelee tarvittavia musiikkilaitteita ja esittelee MIDI-standardin periaatteita. Viidennessä luvussa jaotellaan musiikkiohjelmat pääasiallisen käyttötarkoituksensa perusteella erilaisiin ohjelmatyyppeihin, ja kuudennessa luvussa tehdään vielä yhteenveto koko selvityksestä.

2. Lähtökohtia

Tässä luvussa käydään läpi tutkimusaiheen rajaamisen ja ymmärtämisen kannalta tärkeimmät termit ja käsitteet. Musiikkia, sen tekemistä, ja siinä käytettyä välinettä, eli tässä yhteydessä tietokonetta määriteltäessä, ei etsitä lopullista ja ainoa totuutta, vaan käsitteitä yritetään selventää ainoastaan siinä kontekstissa kuin niitä tässä tutkimuksessa käytetään. Tietokoneen ja musiikin yhteistyöhön liittyy myös joitakin moniselitteisiä ja helposti väärinymmärrettäviä termejä, joita kokonaisuuden hahmottamiseksi yritetään hiukan selventää. Lopuksi luodaan lyhyt katsaus tietokoneavusteisen musiikintekemisen historiaan.

2.1. Musiikista

Selvityksen tutkimusasetelmasta johtuen musiikki määritellään ensisijaisesti tekijän näkökulmasta. Musiikilla on siis aina tekijä ja tekijällä puolestaan joku

musiikillinen tavoite. Tämä tavoite voi olla esimerkiksi nuottikirjoituksen muodossa oleva musiikin kuvaus, tai soiva musiikkiesitys kuten äänite. Näiden esilletuomisella tekijä voi pyrkiä pelkästään oman tunnetilansa tai mielialansa taiteelliseen ilmaisemiseen ja sen pysyvään tallentamiseen. Mutta tarkoituksena voi olla myös suurempia päämääriä, kuten yhteiskunnallisia, ideologisia tai taloudellisia. Lopullinen tuote voidaan karkeasti jakaa joko *funktionaaliseen* tai *absoluuttiseen* musiikkiin. Absoluuttisella musiikilla tarkoitetaan musiikkia, jota tehdään musiikin itsensä vuoksi, mutta funktionaalisessa musiikissa tavoitellaan myös ulkomusiikillisia päämääriä [SM, 1991]. Nämä päämäärät eivät välttämättä ylety kuulijaan saakka, sillä kuulijalla on mahdollisuus kokea musiikki aina omalla tavallaan.

Musiikin soivaan lopputulokseen, ja siitä kuulon perusteella tehtävään tulkintaan, vaikuttavat myös monet teoksen ulkopuoliset tekijät, esimerkiksi esitystekniikka ja esitystilan akustiset olosuhteet. Nämäkin ovat asioita, joihin tekijä ehkä haluaisi puuttua, ja joissa tietotekniikkaa voidaan käyttää hyväksi, mutta ne rajataan kuitenkin tämän tutkimuksen ulkopuolelle.

Musiikin määrittäminen tyyli-suunnan mukaan on erittäin hankalaa. Samoja termejä käytetään eri tavoin riippuen asiayhteydestä tai esimerkiksi siitä, onko kysymys ns. korkeakulttuurista vai populaarikulttuurista. Toisaalta termit saattavat kuvata musiikin sijaan ainoastaan välinettä, jolla musiikki on tuotettu.

Korkeakulttuurin puolella *elektronimusiikki* voidaan jakaa *nauhamusiikkiin*, jossa esitetään aikaisemmin tehty tallenne, ja *elektroakustiseen* musiikkiin, joka yhdistää esitystilanteessa luonnollisia ja sähköisiä soitinääniä. Nauhamusiikki voidaan jakaa äänilähteen perusteella edelleen *konkreettiseen* musiikkiin, joka käyttää nauhoitettuja ympäristön ääniä ja *"puhtaaseen" elektronimusiikkiin*, jossa kaikki äänet ovat sähköisesti tuotettuja. [SibA, 2003].

Tietokone-musiikiksi kutsutaan yleensä musiikkia, jonka synnyttämisessä tai esittämisessä tietokoneella on ollut ratkaiseva rooli [SibA, 2003]. Termiä on käytetty pääasiassa ns. taidemusiikin puolella. Sen sijaan *konemusiikiksi* saatetaan kutsua vastaavin menetelmin tehtyä populaarimusiikkia, esimerkkeinä vaikkapa *tekno*, *trance*, *ambient* jne. Alaa tuntemattomat eivät osaa eritellä alagenrejä ja saattavatkin kutsua teknomusiikiksi kaikkea elektronista musiikkia [Suviola ja Hinkula, 2001]. Toisaalta konemusiikki ei välttämättä ole tuotettu hiiren, näppäimistön ja tietokoneen avulla, mutta nykypäivän elektroniset instrumentit sisältävät silti vastaavanlaista tietotekniikkaa kuin tietokoneet ohjelmistoineen.

Myös aivan tavallista käyttömusiikkia tehdään tietokoneen avulla. Esimerkiksi mainosmusiikki ja televisio-ohjelmien taustamusiiikki voidaan

tehdä pelkästään tietokoneen ja ohjelmiston avulla, mutta tällä tavoin tehtyä musiikkia kutsutaan silti harvemmin tietokonemusiikiksi.

2.2. Musiikintekijästä ja tekemisestä

Tekijä on tässä tutkimuksessa yleisnimi mm. säveltäjälle, lauluntekijälle, sovittajalle, muusikolle, miksaajalle ja tuottajalle, yleensä ottaen kaikille niille, joilla voidaan ajatella olevan jonkinlainen rooli musiikin tekemisen eri vaiheissa. Tekijän käsite onkin tässä hieman laajempi kuin suomalaisen musiikin tekijänoikeusjärjestön *Teoston*, jonka tekijäasiakkaita ovat säveltäjä, sanoittaja ja sovittaja [Teosto, 2003]. Vaikka sanat ovat usein tärkeä tai välttämätön osa musiikkia, niin tässä selvityksessä niiden tekijät unohdetaan kokonaan. Sen sijaan tekijöiden joukkoon otetaan esittävien taiteilijoiden järjestön *Gramexin* asiakkaat, joita ovat äänitteen tuottajat ja muusikot [Gramex, 2003].

Periaatteessa osatehtävän tekijänä voi olla myös kone, mutta viimeistään siinä vaiheessa kun mietitään teon motiiveja, niin tekijän paikalta löytyy lopulta aina ihminen. Tosin 1960-luvulla professori Markku Nurminen ohjelmoi tietokoneen säveltämään tangoja, jotka matkivat Toivo Kärjen sävellyksiä. Eräs ohjelman säveltämä tango jopa levytettiin [Suominen, 2000, 95-96] ja sen tekijänoikeuspalkkiot osoitettiin Turun yliopiston tietokoneiden huoltotilille [Hyvönen, 2001].

2.3. Tietokoneesta

Suurimmalla osalla suomalaisista äänitysstudioista on jo tietokone osana vakiokalustoa [Oesch, 1998, 57]. Vaikka itse äänitys hoidettaisiin digitaalisesti, niin monipuoliseen kaupalliseen studioon kuuluu aina myös analogisia laitteita ja menetelmiä. Kokonaan digitaalisia studioita, ilman ainuttakaan analogista laitetta, löytyy lähinnä musiikin tekijöiden yksityiskäyttöön tarkoitetuista kotistudioista.

Ennen kuin tietokone voi käsitellä ääntä, niin se on muutettava digitaaliseen muotoon. Tämä tapahtuu A/D -muuntimen (analogi-digitaali) avulla. Nykyiset kotitietokoneet sisältävät vähintään 16-bittisen äänipiirin, joka sisältää A/D -muuntimen, tukee MIDI-standardia ja käyttää äänenmuodostukseen joko FM-synteesiä tai laadukkaampaa ääninäytteisiin perustuvaa *Wavetable-synteesiä*. Periaatteessa niillä on siis mahdollista tuottaa *CD-tasoista* (16bit/44kHz) ääntä sopivien ohjelmien avulla. Ammattimaisempaan käyttöön on saatavissa myös laadukkaampia äänikortteja.

Käyttöjärjestelmällä ja laitealustalla on merkitystä lähinnä siinä mielessä, että järjestelmästä riippuen niille on tarjolla vaihteleva määrä erilaisia musiikkisovelluksia.

2.4. Taustasta

Tietokone-musiikin syntyvaiheita elettiin 1950-luvulla, jolloin kreikkalainen Iannis Xenakis loi tietokoneen avulla musiikkia, joka koostuu erilaisista tekstuuripinnoista ja niiden välisistä vähittäisistä muutoksista. Xenakis käytti tietokonetta nimenomaan musiikin rakenteellisen sisällön luomisvaiheessa. Esittäjiä varten tarvittavan musiikin nuotinnuksen hän joutui vielä tuolloin tekemään omin käsin. [SibA, 2003].

Kölnin radioon perustettiin vuonna 1951 elektronimusiikin studio, jossa tuotettiin ääniä myös synteettisesti. Radion studiossa tehtiin nauhamusiikkia, jossa yhdistettiin luonnollisia ja synteettisiä ääniä. Tämä tekniikka oli kuitenkin varsin hidasta ja vaivalloista, eikä useimpien säveltäjien ulottuvilla ollut vastaavia mahdollisuuksia. [Aho, 1999].

Vuonna 1957 Max Mathews ohjelmoi Bellin laboratoriossa Yhdysvalloissa ensimmäisen todellisen musiikkiohjelman (Music I) IBM-tietokonetta ja assembler-kieltä käyttäen. Ohjelma sai myöhemmin uusia versioita (Music II-V), ja sitä ohjelmoitiin mm. Fortran-kielillä. [Obsolete, 2003].

Ensimmäisen kaupallisen syntetisaattorin kehitti 1960-luvulla amerikkalainen Robert Moog. Legendaarinen Moog-syntetisaattori perustui vielä analogiseen synteettiin, mutta 1970-luvulla markkinoille tuli digitaaliseen FM-synteettiin (Frequency Modulation) perustuva Synclavier, jossa oli mukana myös *sämpläys* (Sampling) mahdollisuus. [Obsolete, 2003]. Sämpläyksellä tarkoitetaan luonnollisia ääniä, jotka on nauhoitettu ja muutettu digitaaliseen muotoon, jolloin niitä voidaan toistaa muokattuina ja esimerkiksi erilaisilta korkeuksilta. FM-synteesi puolestaan perustuu taajuusmodulointiin, ja sen ääni on parhaimmilla uusien synteettisten äänien tuottamisessa, mutta aitojen soittimien matkimiseen se ei kunnolla kykene. 1980-luvun alussa Yamaha esitteli FM-synteettiin perustuvan laitteen nimeltä DX-7, joka on vielä tänäkin päivänä yleisesti käytössä. Samoihin aikoihin Yamaha esitteli myös musiikkitietokoneen nimeltä CX5 Music Computer. Laite perustui DX-7 syntetisaattoriin ja siinä oli tarvittavat musiikkiohjelmat ja erillinen monitori. Lisäksi laitteesta löytyi sovelluksia mm. tekstinkäsittelyyn, taulukkolaskentaan yms.

Tietokone alkoi yleistyä musiikintekijöiden työvälineenä vasta 1980-luvulla, kun ensimmäiset varsinaiset mikrotietokoneet tulivat markkinoille. Vuonna 1983 esiteltiin *MIDI-standardi*, jonka avulla tietokoneet ja erilaiset musiikkilaitteet kykenivät kommunikoimaan keskenään. Tämä mahdollisti jo monikanavaisen synteettisen musiikintekemisen kotikonstein. Mutta vasta 1990-luvun loppupuolella kotitietokoneiden kapasiteetti on antanut kaikille

hyvät mahdollisuudet toteuttaa musiikkia sekä synteettisen että konkreettisen äänen lähtökohdista.

Viime vuosina on kehitetty myös soittimien fysikaaliseen *mallintamiseen* (Modelling) perustuvia instrumentteja, mutta kaupallisia sovelluksia on varsin vähän tarjolla, ja ne ovat vielä kalliita. Kun sämpläys perustuu äänen tallentamiseen ja vaatii paljon muistikapasiteettia, niin mallinnuksessa pyritään tuottamaan alkuperäisen soittimen kaltainen ääni matemaattisesti laskemalla, jolloin koneelta vaaditaan massamuistin sijaan erityisesti prosessoritehoja. Mallintamista voidaan tehdä akustisten soittimien lisäksi vaikkapa analogi-syntetisaattoreista. [Smith, 1996]. Esimerkiksi Moog-syntetisaattorista löytyy erittäin korkealaatuinen ohjelmistopohjainen *virtuaali-syntetisaattori* nimeltä *Moog Modular V*.

3. Musiikin olomuotoja

Kuulijan näkökulmasta musiikilla tarkoitetaan yleensä ääntä, mutta tekijän kannalta se voi merkitä muutakin. Musiikki voi soida säveltäjän mielessä jo ennen kuin se on kirjoitettu nuoteille tai esitetty kuulijoille. Sami Nybacka [Nybacka, 2001] käyttää musiikin erilaisista ilmentymistä osuvaa termiä *musiikin olomuodot*.

Tässä tutkielmassa erotellaan nämä kolme erilaista olomuotoa *musiikin käsitteelliseen sisältöön*, jollakin merkkijärjestelmällä esitettävään *musiikin kuvaukseen* ja *musiikin soivaan eli auditiiviseen esitykseen*. Perinteisimmässä tapauksessa olomuodot seuraavat toisiaan vaiheittain, jolloin musiikin sisällön ja kuvauksen tuottavat yleensä säveltäjät ja esityksestä huolehtivat muusikot, mutta tietokonetta käytettäessä nämä roolit saattavat yhdistyä monella eri tavalla.

3.1. Musiikin käsitteellinen sisältö

Säveltäjä Jukka Tiensuu [Tiensuu, 1991b, 264] puhuu henkisestä toiminnasta, oivalluksesta ja abstraktista ideasta, jonka symboleina partituuri ja sen soiva esitys toimivat. Suomalaiset säveltäjät Kaija Saariaho ja Markus Lindberg ovat molemmat luonnehtineet tietokonetta pelkästään musiikillisten visioiden toteuttamisen apuvälineeksi [Aho, 1999]. Idea ja oivallus syntyvät säveltäjän aikaisempiin kokemuksiin perustuvien mielleyhtymien seurauksena [Heinonen, 1995].

Musiikin käsitteellinen sisältö luodaan siis tavallisesti ilman tietokonetta, mutta idean etsimisen vaiheessa tietokoneen käytöstä saattaa olla virikkeiden antajana apua.

Oman pikantin lisäyksensä musiikintekijäin soppaan heitti säveltäjä John Cage, kun hän ensi kertaa vuonna 1952 esitti teoksensa 4'33". Teos sisälsi vain taukoa. Musiikki ei hänen mielestään ole keino vastaanottaa säveltäjän ajatuksia vaan keino havainnoida maailmaa [Tiensuu, 1991a, 14-15]. Cagen mielestä kaikki on musiikkia [Aho, 1999].

3.2. Musiikin kuvaus

Säveltäjällä on varmasti polttava tarve tallentaa ideansa tavalla tai toisella pysyvään muotoon. Jos tavoitteena on funktionaalinen musiikki, niin tallennusmenetelmäksi saattaa riittää pelkkä musiikin kuvaus. Tällöin säveltäjä merkitsee ideansa muistiin käyttäen muusikoiden tuntemaa kieltä, eli yleensä nuottikirjoitusta. Muusikot puolestaan tulkitsevat säveltäjän tekemää kuvausta ja antavat esitysvaiheessa teokselle myös oman panoksensa.

Vaikka nuottikirjoitus ja muut musiikin kuvauskielet ovat aina puutteellisia, niin säveltäjän tärkeimmät tavoitteet ovat silti merkittävässä muistiin niiden avulla. Muusikoiden toteuttama teoksen esitys saattaa olla säveltäjän mielestä kompromissi, mutta se voi olla myös enemmän kuin pelkkä alkuperäinen idea.

Kuvauksen on oltava erittäin tarkka, jos esittäjänä on tietokone, sillä elävistä esittäjistä poiketen koneella ei ole sitä taustatietoa ja ymmärrystä, jota muusikot käyttävät hyväkseen tulkitessaan nuotteja.

Säveltäjä Ilhan Mimaroglu vertaa nuotteja rakennuksen suunnitelmaan tai elokuvan käsikirjoitukseen, molemmat ovat vain eräs suunnitelman kehitysvaihe [Cage, 1969]. Jotkut ajattelevat, että nuotit ovat yhtä kuin soiva musiikki. Toiset taas, että nuotinnuksella ei ole mitään tekemistä soivan musiikin kanssa. [Marco, 1969]. Musiikin kuvaus on kuitenkin välttämätön välivaihe, mikäli käsitteellistä sisältöä halutaan tallentaa tai siirtää eteenpäin joko toisille ihmisille tai koneille, ja edelleen soivaksi musiikiksi. Kuvausta tarvitaan siis kaikissa muissa tapauksissa paitsi silloin, kun säveltäjä itse esittää tai tallentaa oman teoksensa.

Musiikin kuvauskielet voidaan jakaa periaatteessa kahteen ryhmään riippuen siitä, onko ne tarkoitettu muusikoille vai koneille. Länsimaisessa musiikissa nuottikirjoituksella tarkoitetaan yleensä viivastonotaatiota, mutta muitakin merkkijärjestelmiä on käytössä. Yleensä kuitenkin ihmisille tarkoitettut musiikin kuvauskielet ovat visuaalisesti tulkittavia graafisia merkkijärjestelmiä. Poikkeuksena on sokeille tarkoitettu tuntoaistiin perustuva järjestelmä.

Koneille tarkoitettuja kuvauskieliä on lukematon määrä. Aivan kuten yleensäkin tietokoneohjelmilla, niin lähes kaikilla musiikkiohjelmilla on oma tapansa tallentaa tapahtumia tiedostoihin. Musiikin tekijän kannalta tämä on

hankalaa. Jos ohjelmia ei päivitetä lainkaan tuleviin tietokone- tai käyttöjärjestelmäversioihin, niin miten tallennettuun tietoon pääsee tulevaisuudessa käsiksi? Ainoaksi vaihtoehdoksi jäävät erilaiset konversio-ohjelmat tai standardit kuvauskielet, joita useimmat ohjelmat tukevatkin. Näitä käyttämällä kuitenkin menetetään yleensä osa siitä informaatiosta, jota musiikkiohjelmalla on synnytetty.

3.3. Musiikin auditiivinen esitys

Jos tavoitteena on absoluuttinen musiikki, ilman mitään ulkomusiikillisia päämääriä, niin säveltäjälle on tärkeitä myös se, miten tarkasti alkuperäistä ajatusta osataan tulkita. Ennen äämentallennusmenetelmien kehittymistä säveltäjän ainoa mahdollisuus kontrolloida esitystä mahdollisimman tarkkaan, oli osallistua itse soittamiseen tai orkesterin johtamiseen.

Tänä päivänä tietokoneen käyttömahdollisuudet ovat monipuolisimmillaan juuri auditiivisen esityksen tuottamisen kohdalla. Mikä tahansa digitaaliseen muotoon muunnettu ääni voi olla musiikin materiaalina. Edes säveltäjiä ja muusikoita ei välttämättä enää tarvita. Musiikin tekeminen voi olla ”leikkaa ja liimaa” tyyppistä palapelin pelaamista.

Laadukkaan auditiivisen musiikkiesityksen tuottaminen sisältää kuitenkin monia erilaisia vaiheita ja vaatii tekijältään todellista moniosaamista.

3.4. Yhteenveto

Musiikin käsitteellisen sisällön tuottaminen on ensisijaisesti säveltäjän varassa, mutta nuotinnusprosessissa tietokoneet ovat jo helposti käytettävissä. Sävelteoksen luoja pääsee siinä mielessä helpommalla, että tekemällä pelkästään musiikin kuvauksen, hän voi liikkua yleisemmällä tasolla kuin tekijä, joka pyrki kontrolloimaan tarkasti myös musiikin auditiiviseen muotoon saattamista. Vivahteikkaan soivan musiikin aikaansaamiseksi tekijän on hallittava nämä kaikki olomuodot. Soivan musiikin tuottaminen vaatii tekijältään myös muusikon virtuoosimaisia teknisiä taitoja, mutta instrumenttina on tuolloin tietokone ja musiikkiohjelmistot.

4. Musiikkilaitteet

Useimmat sähköiset musiikkilaitteet sisältävät myös MIDI-liitännän. Tietokoneen ulkopuoliset MIDI-laitteet eivät ole musiikintekijän kannalta välttämättömiä, mutta niiden avulla päästään ainakin toistaiseksi parempaan äänenlaatuun ja monipuolisempaan käytettävyyteen kuin vastaavilla ohjelmistopohjaisilla tuotteilla. Historia on osoittanut, että monet MIDI-laitteet tulevat ensin markkinoille ja vasta myöhemmin niitä matkivat MIDI-ohjelmistot. Näin on käynyt ainakin *sekvensserien* (Sequencer) ja sämpplerien

kohdalla. Sekvensserillä tarkoitetaan laitetta, joka kykenee nauhoittamaan moniraitaisesti MIDI-tietoa.

4.1. MIDI-laitteet

MIDI-laitteiden välissä tarvitaan MIDI-kaapeli, ja tietokoneessa on oltava MIDI-liitäntä ennen kuin se voi kommunikoida ulkoisten MIDI-laitteiden kanssa. Nykykoneissa liitäntä on yleensä valmiina, mutta sen voi hankkia myös lisäkorttina liitettäväksi johonkin tietokoneen erilaisista väylistä.

Ensimmäiset MIDI-ohjaimet olivat pelkkiä tietokonenäppäimistöjä tai pianon ”käyttöliittymää” muistuttavia koskettimistoja, mutta nykyisin vaihtoehtoja on jo useita. Käytettävällä ohjaimella on suuri vaikutus sekä syntyvään musiikkiin että soittimen käytettävyyteen. Esimerkiksi viululla säveltäessä syntyy erilaista musiikkia kuin kitaralla, ja sama pätee tietysti muihinkin soittimiin ja ohjaimiin.

Ohjelmistopohjaiset MIDI-ohjaimet muistuttavat aitoja esikuviaan, mutta alkuinnostuksen jälkeen niiden suosio on vähentynyt varsinkin ammattilaisten keskuudessa. Esimerkiksi mikseriohjelman säätönappien käytettävyyys hiirellä on perin kankeata verrattuna konkreettisesti kosketeltaviin säätimiin.

Nykyiset ulkoiset MIDI-ohjaimet ovat jo tavallaan *multimodaalisia*, eli ohjaustietoa voidaan syöttää useammalla eri syöttötavalla samanaikaisesti. Mutta jos multimodaalisuuden käsitettä laajennetaan vielä siten, että usean eri aistin varaan, tai vaikka liikkeentunnistusmenetelmien avulla järjestetään ohjausmahdollisuus, niin tällä tavoin luotu musiikkikin saa varmasti aivan uusia ulottuvuuksia. Multimodaalisuuden käsitettä käytti ensimmäisen kerran Richard A. Bolt [1980] kuuluisan Put-That-There järjestelmänsä yhteydessä [Raisamo, 1997].

Ulkoiset äänilähteet voivat olla joko koskettimistolla varustettuja tai pelkkiä soundimoduleja (sound module). Tietokonekäytössä käytännöllisimpiä ovat ns. *multitimbraalit* laitteet, jotka voivat soittaa samanaikaisesti useita eri instrumentteja eri MIDI-kanavilla.

4.2. MIDI 1.0 Standardi

Sähköiset musiikkilaitteet ovat perustuneet osittain tietotekniikkaan jo aikana, jolloin kotitietokoneet vasta tekivät tuloaan markkinoille. 1970-luvun loppupuolella tuli mahdolliseksi käyttää halpoja mikroprosessoreja ohjaamaan syntetisaattoreita [Roads, 1996, 973]. Tuolloin jokaisella valmistajalla oli oma toteutustapansa, eivätkä laitteet olleet yhteensopivia toistensa kanssa. Musiikkia ei voitu siirtää digitaalisessa muodossa laitteelta toiselle.

Vuonna 1981 aloitettiin eri laitevalmistajien toimesta kehittämään standardia, jonka ensimmäinen versio julkaistiin vuonna 1983. Näin syntynyt

MIDI-standardi (Musical Instruments Digital Interface) on edelleen lähes kaikkien markkinoilla olevien sähköisten musiikkilaitteiden, musiikkiohjelmistojen, sekä myös monien muiden laitteiden yhdistävänä tekijänä. Curtis Roads [1996, 975] esittelee MIDI:n mukanaan tuomia musiikillisia mahdollisuuksia seuraavasti:

1. MIDI erottaa ohjaimen ja syntetisaattorin (äänilähteen) toisistaan. Tällöin on mahdollista samanaikaisesti ohjata yhdellä ohjaimella useita eri äänilähteitä.
2. Toisaalta erottaminen mahdollistaa myös erityyppisten ohjaimien, kuten koskettimistojen, puhaltimien, rumpujen, kielisoittimien tai vaikkapa mikrofoniin käytön ohjaimena.
3. Tietokoneohjelmistot voivat ohjata äänilähteitä. Samoin myös syntetisaattoreilla synnytetty MIDI-tieto voidaan siirtää tietokoneeseen.
4. MIDI mahdollista yleisten laiteriippumattomien tietokoneohjelmien suunnittelun, kuten esimerkiksi sävellys-, sekvensseri- ja nuotinkirjoitusohjelmat.
5. Myös tietyille laitteille tarkoitettuja ohjelmia on helpompi suunnitella. Tämä on tarpeen esimerkiksi silloin, kun halutaan parempi käyttöliittymä jollekin ulkoiselle MIDI-laitteelle.
6. MIDI-tietoa voidaan syöttää myös muille laitteille kuin syntetisaattoreille. Esimerkiksi efektilaitteita, mikseriä tai valoja voidaan ohjata MIDI:n avulla reaaliaikaisesti ja synkronoituna musiikin kanssa yhteen.
7. MIDI-tietoa voidaan siirtää eri valmistajien tuotteiden välillä.

MIDI-kielen peruskäsitteitä ovat MIDI-kanavat ja MIDI-viestit. Yksinkertaisimmillaan MIDI-viesti syntyy esimerkiksi siten, että painetaan MIDI-ohjaimella yksi näppäin alas, jolloin MIDI-kaapelia pitkin lähtee vastaanottavalle MIDI-soittimelle komento *Note on*. Kun näppäin nostetaan ylös, niin lähtevä viesti on *Note off*. Sama voidaan tehdä myös ohjelmistolla ilman MIDI-ohjainta. Jos esimerkiksi nuotinkirjoitusohjelmassa viivastolle pudotetaan nuotti ja annetaan ohjelmalla komento *Play*, joka käynnistää kappaleen, niin vastaanottava MIDI-laite, oli se sitten sisäinen äänikortti tai ulkoinen soitin, soittaa äänen ja myös sammuttaa sen. Äänen automaattinen sammuminen johtuu tietysti siitä, että nuotin aika-arvo sisältää myös *Note off* -komennon.

MIDI-standardi käsittää 16 kanavaa, ja jokainen soitettavaksi tarkoitettu MIDI-viesti liitetään johonkin näistä kanavista. Jos kaikilla 16 kanavalla on eri instrumentille tarkoitettua ohjaustietoa, ja vastaanottavassa päässä todella käytetään myös 16 eri instrumenttia, niin lopputulos saattaa kuulostaa orkesterilta, mutta valitettavasti vaatii myös yhtä suuren tilan. Käytännössä näin ei jouduta kuitenkaan tekemään, koska useimmat MIDI-instrumentit ovat multitimbraaleja, ja kykenevät siis käsittelemään samanaikaisesti useita eri kanavia ja instrumentteja.

Kanavakohtaisten viestien (Channel Messages) lisäksi käytettävissä on myös muita komentotyyppisiä. Esimerkiksi laitteistolle tarkoitettujen yleisluontoisemmat komennot (System Messages), jotka liittyvät esimerkiksi ajoitukseen tai kappalevalintaan. Ne voivat olla myös laitevalmistajien itse määrittelemiä ohjauskomentoja, joilla viitataan laitteen tiettyihin ominaisuuksiin.

MIDI on laitteiden ja ohjelmistojen kannalta tiedonsiirtostandardi, tallennusformaatti ja rajapinta, mutta musiikin tekijän kannalta sitä voidaan ajatella myös yhtenä musiikin kuvauskielenä. Visuaaliset kuvauskielet tarvitsevat soittajan ja instrumentin, jotta musiikki saadaan soivaan olomuotoon. Mutta MIDI-ohjelmisto tarvitsee syntetisaattorin, sämplerin tai tietokoneen äänikortin, jotta viestit voidaan muuttaa digitaaliseksi ääneksi. Soivaksi ääneksi ne muuttuvat vasta monien vaiheiden jälkeen kaiuttimista ulos tultuaan.

Koska MIDI-standardi kehitettiin tosiaikaiseen kommunikointiin eri laitteiden välillä, niin se ei sisältänyt tallennusformaatin määrittelyä lainkaan. Aluksi eri ohjelmistovalmistajat käyttivät keskenään yhteen sopimattomia tiedostoformaatteja ohjelmissaan, ja vasta vuonna 1988 kehitettiin laajennus nimeltä *Standard MIDI Files (SMF)* [Roads, 1996, 999].

MIDI-standardia on täydennetty myös monilla muilla laajennuksilla ja täydennetään yhä edelleen *MIDI Manufacturers Association* yhteisön toimesta. [MMA, 2003].

5. Musiikkiohjelmat

Monet musiikkiohjelmat eivät sovi puhtaasti mihinkään seuraavista kategorioista, ja jotkut ovat lähes täydellisiä musiikkituotantotyökaluja, joiden avulla koko musiikintekemisen prosessi voidaan hoitaa alusta loppuun saakka. Musiikintekemiseen tarkoitettujen ohjelmien ovat seurailleet samantapaista kehityshistoriaa kuin muutkin tietokoneohjelmat. Usein ohjelmistokehitys lähtee liikkeelle jostakin erityistarpeesta ja laajenee vähitellen monitoimittamiseksi, joka pyrkii kattamaan kaikki kuviteltavissa olevat

kuluttajan tarpeet. Samat asiat ovat kuitenkin tehtävissä myös erillisillä, yhteen osa-alueeseen suunnitelluilla pienemmillä ohjelmilla. Jopa ilmaisohjelmia löytyy lähes kaikkiin tarpeisiin. Täysin välttämätön hankinta on siis ainoastaan tietokone.

5.1. Sekvensseriohjelmat

MIDI-sekvensseri on tietokoneavusteisen musiikintekemisen perustyökalu. Sekvensseri toimii samaan tapaan kuin moniraitainen ääninauhuri, mutta äänen sijaan sillä talletetaan pelkkiä musiikin ohjauskomentoja, eli MIDI-dataa. Ensimmäiset sekvensserit olivat ulkoisia laitteita ja saattoivat liittyä johonkin erilliseen musiikkityöasemaan. Monet koskettimistolla varustetut syntetisaattorit sisältävät myös sekvensserin.

Sekvensserissä jokainen instrumentti nauhoitetaan yleensä omalle raidalleen, johon voidaan liittää myös tiedot käytettävästä instrumentista ja MIDI-kanavasta. Nauhoittaminen tapahtuu joko soittamalla reaaliaikaisesti MIDI-ohjaimen avulla tai syöttämällä komentoja yksitellen pelkän ohjelman avulla. Ohjelma tarjoaa useita eri sisäänsyöttö- ja esitystapoja, joista tyypillisimpiä ovat *Event List* ja *Piano Roll*. *Event List* on listamuotoinen esitys MIDI-tapahtumista. Se näyttää kaikki MIDI-tapahtumat raitakohtaisesti, ja kokenut käyttäjä voi muokata tapahtumia suoraan listalta. *Piano Roll* -ikkuna on puolestaan visuaalinen esitystapa raitakohtaisista MIDI-tapahtumista. Siinä tapahtumat esitetään ruudukossa, jossa vaakakseli kuvaa äänen kestoa ja pystyakseli korkeutta. Tapahtumia voidaan lisätä ja muokata hiiren avulla.

Sekvensserien ominaisuuksin kuuluu myös monia tapahtumatiedon muokkaustoimintoja, joiden käyttöperiaatteet ovat hyvin samantapaisia kuin vaikkapa kuvankäsittelyohjelmissä. Ensin määritellään alue, johon komento halutaan osoittaa, ja sitten valitaan efekti ja sen parametrit. Esimerkiksi sisäänsoitetun musiikin rytmisen epätarkkuus on helppo poistaa *kvantinsointi-*komennolla (*Quantize*).

5.2. Nuotinnusohjelmat

Nuotinnusohjelmat ovat usein myös sekvenssereitä, ja nuotit voidaan niissä joko soittaa tai lisätä viivastolle graafisesti hiiren ja näppäimistön avulla. Vaikka nykyiset nuotinnusohjelmat hallitsevat myös MIDI-standardin, niin niiden päätarkoitus on kuitenkin musiikin kuvaaminen graafisessa muodossa. Toisaalta nuotinnusohjelma voi nuotteihin tottuneelle säveltäjälle olla se paras tapa tutustua myös auditiivisen musiikin tekemiseen.

Nuotinkirjoitus on taidetta jo itsessään, ja monilla säveltäjillä on oma aivan erityinen tapansa kuvata musiikkiaan. Kaupallisista tuotteista ei löydy juuri muita kuin perinteiseen länsimaiseen viivastonotaatioon perustuvia ohjelmia,

joten säveltäjät saattavat käyttää muitakin grafiikkaohjelmia musiikillisten merkistöjensä tuottamiseen. MIDI-pohjaisessa notaatio-ohjelmassa on kuitenkin se etu, että sovituksia voi kuunnella jo työskentelyprosessin aikana.

5.3. Audiosekvensserit

Jo 1920-luvun Neuvostoliitossa käytettiin musiikkiteosten äänimateriaalina konkreettisia ääniä ja hälyjä [Aho, 1999]. Näitä ensimmäisiä kokeiluja voidaan pitää lähtökohtana myös konkreettiselle nauhamusiikille. Nauhamusiikki puolestaan toimii selvänä esikuvana nykyiselle sämplepohjaiselle digitaaliselle musiikille. Viime vuosina on valokuvataiteen puolella aiheuttanut kiivasta vastustusta luontokuvien manipuloiminen tietokoneella. Vastaavia käsityksiä on esitetty myös musiikin puolella. 1970-luvulla ranskalainen Barnard Machen käytti konkreettisen musiikin yhdistelytekniikka, mutta ei hyväksynyt lainkaan alkuperäisten äänien muuntelua [SibA, 2003]. Nykypäivän tietokoneohjelmistojen käyttökelpoisimmat ominaisuudet konkretisoituvat kuitenkin juuri näissä monipuolisissa muokkausmahdollisuuksissa.

Audiosekvensseri on digitaalinen moniraitanauhuri, joka käsittelee sekä ääntä että MIDI-tietoa. Audiodata talletetaan tietokoneen kovalevyille tiedostoina, joiden koko stereofonisena on noin 10 Mb minuutti. Tämä selittää miksi audiosekvensserit ovat yleistyneet vasta viime vuosina. Käytännössä tietokoneessa on oltava useiden kymmenien gigatavujen kovalevy, jotta moniraitaisen musiikin työstäminen olisi joustavaa. Audiodatan käsittely vaatii myös runsaasti keskusmuistia ja prosessoritehoja, jotta erilaiset efektit ja muokkaustoimenpiteet saadaan sujuvasti tehtyä.

5.4. Muut ohjelmat

Ohjelmistopohjaiset sämplerit ja virtuaaliset syntetisaattorit ovat hyvä vaihtoehto silloin, kun äänikortin omat ominaisuudet äänen tuottajana tuntuvat liian rajoitetuilta. Lisäksi on saatavissa erilaisia efektilaitteita, jotka toimivat *plugin* -periaatteella monissa eri musiikkiohjelmissa.

Sävellys- säestys- ja sovitusohjelmat ovat hyvä keino lähteä liikkeelle musiikin tekemisessä. Niiden avulla musiikki voidaan generoida osittain automaattisesti. Esimerkiksi kappaleen sointurakenne, tempo, soittimet ja tyyli voidaan antaa lähtötietoina, ja ohjelma generoi erilaisista valmiista vaihtoehtoista koottua musiikkia. Lopputulos ei välttämättä ole täydellistä, mutta se on kuitenkin hyvä lähtökohta esimerkiksi sekvensseriohjelmaan siirrettäväksi, ja siellä edelleen jatkojalostettavaksi. Joissakin ohjelmissa tyyliä voi suunnitella myös itse, jolloin on helpompi myös hyväksyä musiikki omaksi tuotokseksi.

Musiikkia voi rakentaa myös palapelimäiseen "leikkaa ja liimaa" -tyyliin

ohjelmilla, jotka käsittelevät musiikkia pieninä katkelmina. Nämä luuppiohjelmat (Loop) muodostavat ruudukon, jonka soluja voidaan järjestellä halutulla tavalla.

Äänenkäsittelyohjelmat ovat ohjelmia, joita käytetään musiikin viimeistely- ja miksausvaiheissa. Ohjelmilla säädellään esimerkiksi musiikin dynamiikkaa ja taajuusaluetta. Ääntä käsitellään niissä yleensä lopullisessa stereofonisessa muodossaan ilman erillisiä työnaikaisia raitoja.

6. Lopuksi

Tutkielman aihe osoittautui lopulta liian laajaksi käytettävissä olevaan aikaan ja sivumäärään nähden. Ehkäpä liian perusteellinen taustatietojen selvittely aiheutti sen, että itse pääasialle, eli tietokoneen erilaisille rooleille ja musiikkiohjelmistojen kartoittamiselle, ei riittänyt enää kunnolla panoksia. Selvitys toimii kuitenkin tutustumisena aihealueeseen ja tuo esille alan tärkeimpiä käsitteitä. Ohjelmistopuolta olisi mielenkiintoista jatkaa varsinkin musiikin ilmaisiohjelmia tarkastelemalla. Myös ohjelmistopohjaiset virtuaaliset äänilähteet olisivat tarkemman selvityksen arvoisia.

Tietokone on monipuolinen väline musiikin tuottamisen eri vaiheissa. Käytetty väline vaikuttaa oleellisesti sekä tekemisen prosessiin että syntyvään lopputulokseen. Musiikin kuvauksessa tietokonetta käytetään mielellään apuvälineenä, mutta käsitteellisen sisällön tuottamisessa siitä ei ole juurikaan apua. Toisaalta tietokoneohjelmien avulla säveltäminen ei ole tekijän kannalta kovin mielekästäkään, sillä kukapa haluaisi antaa luomisen iloa tai kunniaa koneelle.

Arkistointia ajatellen on hyvä pitää mielessä, että kaikki tietokoneelle tarkoitetut musiikin tallennusmuodot ovat nykyisen tiedon mukaan pitkäaikaissäilytyksessä arveluttavampia kuin perinteinen paperille tulostettu notaatio.

Soivan musiikin tuottamisessa tietokone on parhaimmillaan. Se vaatii kuitenkin käyttäjältään musikaalisuuden lisäksi myös monipuolista teknistä osaamista. Sopivien ohjelmien avulla tietokoneella voi tuottaa ”muodollisesti pätevää” musiikkia vaikkapa kokonaan ilman soittotaitoa.

Viiteluettelo

[Aho, 1999] Kalevi Aho, 1999. *Miltä kuulosti 1900-luku*. Saatavana:

<http://www.tsv.fi/ttapaht/997/aho.htm> (13.12.2003)

[Bolt, 1980] Richard A Bolt, Put-That-There. *SIGGRAPH '80 Conference Proceedings*, ACM Press, 1980, 262-270.

[Cage, 1969] John Cage, *Notations*. Something Else Press, 1969.

- [Gramex, 2003] Gramex ry. Esittävien taiteilijoiden ja äänitteiden tuottajien tekijänoikeusyhdistys. Saatavana: <http://www.gramex.fi> (13.12.2003)
- [Heinonen, 1995] Yrjö Heinonen, *Elämyksestä ideaksi - ideasta musiikiksi. Sävellysprosessin yleinen malli ja sen soveltaminen Beatles-yhtyeen laulun- ja äänitysprosessiin*. Jyväskylän yliopisto, 1995.
- [Hyvönen, 2001] Eero Hyvönen (toim.), *Inhimillinen kone - konemainen ihminen*. Helsinki University Press, 2001. Ks. myös: <http://www.cs.helsinki.fi/abinfo/Databitti02/alykaskone.html> (13.12.2003)
- [Marco, 1969]. Tomas Marco. Teoksessa: *John Cage, Notations, Something Else Press*, 1969.
- [MMA, 2003] *MIDI Manufacturers Association*. Saatavana: <http://www.midi.org/> (13.12.2003)
- [Nybacka, 2001] Sami Nybacka, *Musiikin kuvauskielet*. Pro gradu -tutkielma, Jyväskylän yliopisto. Saatavana: <http://www.lib.helsinki.fi/eva/SNybackaMusiikki.pdf> (13.12.2003)
- [Obsolete, 2003] 120 Years of Electronic Music. Saatavana: http://www.obsolete.com/120_years/ (13.12.2003)
- [Oesch, 1998] Pekka Oesch, *Musiikkiteollisuus Suomessa, Rakenne työllisyys ja talous*. Taiteen keskustoimikunta, 1998.
- [Raisamo, 1997] Roope Raisamo, *Suorakäyttöisyyden lisääminen piirto-ohjelman toiminnoissa kaksikäteisellä vuorovaikutuksella*. Tampereen yliopisto, Lisenssiaattitutkimus, 1997. Saatavana: <http://www.cs.uta.fi/hci/theses/rrlistyo.pdf> (13.12.2003)
- [Roads, 1996] Curtis Roads, *The computer music tutorial*. MIT Press, 1996.
- [SibA, 2003] Sibelius-Akatemia, Koulutuskeskus, *1900-luvun musiikki*. Saatavana: <http://www2.siba.fi/historia/1900/index.html> (13.12.2003)
- [Smith, 1996] Julius O. Smith III, Physical Modelling Synthesis Update. Teoksessa: *Computer Music Journal*, 20: 2, 44-56, MIT Press, 1996. Saatavana: <http://ccrma-www.stanford.edu/~jos/pmupd/> (13.12.2003)
- [SM, 1991] *Suuri Musiikkitietosanakirja*, 4, Weilin & Göös, 1991.
- [Suominen, 2000] Jaakko Suominen, *Sähköäivo sinuiksi, tietokone tutuksi*. Nykykulttuurin tutkimuskeskuksen julkaisuja 67, Jyväskylän yliopisto, 2000.
- [Suviola ja Hinkula, 2001] Teemu Suviola ja Antti Hinkula, 2001. *Koneistofestivaalin 2000-2001 visuaalinen ilme*. Opinnäytetyö. Saatavana: <http://www.lpt.fi/mi/tutkinnot/pdf/Koneisto.pdf> (13.12.2003)
- [Teosto, 2003] Säveltäjien Tekijänoikeustoimisto Teosto ry. Saatavana: <http://www.teosto.fi> (13.12.2003)

- [Tiensuu, 1991a] Jukka Tiensuu, Mutta onko se musiikkia. Teoksessa: *Lauri Otonkoski (toim.), KLANG: uusin musiikki*. Gaudeamus, 1991.
- [Tiensuu, 1991b] Jukka Tiensuu, Nuottikirjoitus. Teoksessa: *Suuri Musiikkitietosanakirja, 4*, Weilin & Göös, 1991, 264-267.

Sähköpostipalvelun ongelmista

Janne Repo

Tiivistelmä.

Sähköpostin ongelmia ja niiden analyysiä. Roskaposti, virukset ja madot sekä uudemmat uhat. Mitkä ovat sähköpostin ongelmat ja miten ne voidaan ratkaista?

Avainsanat ja -sanonnat: Sähköpostipalvelu, sähköpostipalvelun ongelmat.
CR-luokat: C.2.2

1. Johdanto

Sähköpostin ongelmista on viime aikoina herännyt keskustelua. Suurimman julkisuuden keräävät roskaposti ja sähköpostitse leviävät virukset ja madot. Edellä mainitut aiheuttavat pahimmillaan huomattavia kustannuksia ja virukset saattavat jopa aiheuttaa tuhoja tietojen menettämisen muodossa. Sähköpostin toimintaa ei kuitenkaan yleisesti tunneta kovin hyvin, eikä mahdollisia muita sähköpostiin liittyviä ongelmia ei ole tutkittu.

Yleisesti käsitettä sähköposti käytetään viittaamaan sähköpostiohjelmiin, sähköpostipalveluja tarjoaviin palveluntarjoajiin, yksittäisiin sähköpostitileihin sekä Internetin sähköpostipalveluun. Tutkimuksen otsikko, Sähköpostipalvelun ongelmista, asettaa tutkimuksen kohteeksi nimenomaan sähköpostin Internetin palveluna ja rajaa tutkimuksen ulkopuolelle palveluntarjoajat sekä sähköpostin lukemiseen ja lähettämiseen käytettävät ohjelmistot. Sähköpostia tarkastellaan siis yleisimmällä mahdollisella tasolla. Näin pyritään saamaan mahdollisimman laaja kokonaiskatsaus sähköpostipalvelun ongelmakentästä.

Sähköpostin käyttö niin yksityiseen kuin yritystenkin kommunikaatioon on lisääntynyt muutaman viime vuoden aikana räjähdysmäisesti. Tänä aikana erilaiset sähköpostiin ja sen käyttöön liittyvät ongelmat ovat alkaneet uhata koko sähköpostijärjestelmän toimivuutta. Käsitlemme seuraavassa sähköpostin ongelmia ja puutteita. Esittelemme myös lyhyesti nykyisten sähköpostikäytäntöjen ratkaisuja näihin ongelmiin.

Lisäksi pohdimme uusia ratkaisuja sähköpostin ongelmakohtiin. Mitä menetelmiä tai tekniikoita ongelmiin on mahdollista soveltaa? Arvioimme myös mahdollisuuksia olemassa olevien käytäntöjen parantamiseksi ja uuden standardin luomisen mahdollisuutta.

2. Sähköpostin toiminnasta

Sähköposti määritellään tätä tarkastelua varten seuraavasti: "sähköposti on Internetin tekstimuotoinen, ei-reaaliaikaiseen viestintään tarkoitettu palvelu." Sähköpostin kehittämisessä on oleellista muistaa, että järjestelmän vahvuudet ovat juuri tekstimuotoisuudessa ja ei-reaaliaikaisuudessa. Nämä takaavat palvelun soveltuvuuden useille erilaisille alustoille tietokonejärjestelmistä matkapuhelimiin ja muihin uusiin viestimiin. Lisäksi tekstipohjaisuus takaa viesteille hyvän ilmaisuvoiman, etenkin suhteessa viestien kokoon. Multimedia- tai reaaliaikaiseen viestintään on Internetissä muita palveluja.

Sähköposti, kuten muutkin Internetin yleisemmät palvelut, on standardoitu. Standardoinnista vastaa Internetin standardointielin IETF (Internet Engineering Task Force), joka julkaisee standardi- ja protokollamäärittelyjä. IETF:n työryhmät kehittävät Internetin viestinnässä käytettäviä protokollia, joista tärkeimmät on korotettu standardin asemaan. Sähköpostipalvelu koostuu kolmesta erillisestä, standardoiduilla protokollilla määritellystä osasta – nämä ovat *viestin lähettäminen*, *viestin vastaanottaminen* ja *viestimuoto* (message format [Resnick, 2001]).

Sähköposti toimii yksinkertaistettuna seuraavan mallin mukaisesti. Käyttäjä tuottaa sähköpostiviestin, joka koostuu yksinkertaisimmillaan vastaanottajan osoitteesta, viestin tekstiosasta ja mahdollisista liitteistä. Käyttäjän sähköpostiohjelma muuntaa viestin viestimuotoon, lisää siihen tarvittavat otsikkotiedot ja lähettää sen eteenpäin käyttäjän määrittelemälle sähköpostipalvelimelle. Palvelin vastaanottaa viestin ja välittää sen edelleen vastaanottajan postipalvelimelle. Vastaanottajan postipalvelin tallentaa viestin odottamaan vastaanottajan yhteydenottoa. Vastaanottaja ottaa yhteyden omaan postipalvelimeensa ja noutaa viestin luettavaksi.

Edellä kuvatun kaltaisessa sähköpostiliikenteessä käytetään yleensä SMTP-protokollaa [Klensin, 2001] viestin lähettämiseen ja välittämiseen palvelimelta toiselle sekä POP3- [Myers and Rose, 1996] tai IMAP-protokollaa [Crispin, 2003] viestin vastaanottamiseen. Vaihtoehtoisesti palvelu voidaan myös toteuttaa siten, että viestin joko lähettämiseen tai vastaanottamiseen tai molempiin käytetään HTTP-protokollaa ja selainta, mutta näitä menetelmiä ei ole standardoitu, joten en käsittele niitä tässä.

Kaikki edellä mainitut standardin asemassa olevat protokollat on kehitetty jo 1980-luvun alkupuolella. Ajan mittaan protokollia on pyritty päivittämään enemmän senhetkisten vaatimusten mukaisiksi. Protokollien päivittäminen on kuitenkin hidas ja monimutkainen prosessi, joten viive ongelman tai puutteen havaitsemisen ja päivitetyn protokollaversion julkistamisen välillä on usein vuosia. Lisäksi standardien historiallinen yhteensopivuus estää merkittävien

korjausten tekemisen. Päivitetyn standardin ottaminen käyttöön vie aikaa. Näin viive yksittäisenkin puutteen korjaamisessa koko Internetin alueella on pitkä. Vaikka osa palveluntarjoajista onkin jo ottanut käyttöön turvallisuutta ja käytettävyyttä parantavia uusien päivitettyjen standardien mukaisia ohjelmistoja, suurin osa Internetin sähköpostiliikenteestä kulkee silti vanhentuneiden siirtokäytäntöjen mukaisesti.

Standardien hidas päivittäminen on jo itsessään ongelma kaikkien Internetin käyttöön liittyvien protokollien kohdalla, mutta standardeja säätävän IETF:n toiminnan muuttaminen ei sekään ole vaivatonta tai nopeaa. Käytännöllisempi ratkaisu on kehittää toimivampi protokolla tai protokollajoukko korvaamaan vanhat. Kehitystyössä voi toki hyödyntää olemassa olevien protokollien hyviä ominaisuuksia, eikä sähköpostipalvelun luonnetta ole syytä muuttaa.

3. Ongelmista

Sähköpostipalvelussa on edellisessä luvussa kuvattujen syiden takia useita ongelmakohtia ja puutteita. Tarkastellaan seuraavaksi näitä ongelmia ja tutustutaan joihinkin yrityksiin ratkaista niitä.

Sähköpostin ongelmat voidaan jakaa ulkoisiin ongelmiin, turvallisuuden ongelmiin ja viestimuodon ongelmiin. Ulkoiset ongelmat eivät liity sähköpostipalveluun vaan sen käyttöön. Yleisten ohjelmien käytettävyysongelmien, joihin emme tässä paneudu, lisäksi ulkoisia ongelmia ovat esimerkiksi roskaposti ja muu sähköpostipalvelujen hyväksikäyttö. Turvallisuuden ongelmakohtiin kuuluu salauksen puute niin viestin lähettämisen kuin käyttäjän tunnistuksenkin yhteydessä. Viestimuotoon liittyviä ongelmia ovat liitetiedostot, kansainvälisyystuki ja viestimuodon hyvin-muodostuneisuus.

Roskaposti, eli massalähetyksinä lähetettävä mainossähköposti, on viiden viimeisen vuoden aikana paisunut pahimmaksi sähköpostia rasittavaksi ongelmaksi. YK:n tuoreen tutkimuksen [UNCTAD, 2003] mukaan vuoden 2003 loppuun mennessä arviolta puolet maailman sähköpostiliikenteestä aiheutuu roskapostista. Roskaposti on todennäköisesti ongelma, josta emme koskaan pääse kokonaan eroon. Roskaposti yleensä ei ole kuitenkaan luonteeltaan sellainen ongelma, joka olisi yksinkertaista ratkaista protokollia muuttamalla. Mainossähköpostien lähettämisen estämiseen ei ole protokollatason ratkaisua. Liikenteen rajoittaminen jollakin keinotekoisella kriteerillä olisi yksi vaihtoehto, mutta kaikki tämänkaltaiset rajoitukset sotivat Internetin vapautta vastaan. Ratkaisu tähän ongelmaan löytyy muualta kuin protokollista.

Ainoa sähköpostin käytäntöihin liittyvä piirre roskapostissa on avointen sähköpostin välityspalvelimien käyttäminen roskapostin lähettämiseen. *Avoim välityspalvelin* (open relay) on SMTP-protokollan mukainen palvelin, joka ei vaadi käyttäjän tunnistamista sähköpostin lähettämiseksi. Tämä tarkoittaa, että kuka tahansa voi käyttää palvelinta sähköpostin lähettämiseen. Useimmat palveluntarjoajat ovat rajoittaneet palvelimensa käyttöä joko vaatimalla käyttäjän tunnistuksen tai hylkäämällä oman verkon ulkopuolelta tulevat lähetykset. Avoimet välityspalvelimet ovat tällä hetkellä merkittävä roskapostin aiheuttaja, mutta niiden merkitys on vähenemässä roskapostin lähettämiseen erikoistuneiden verkkomatojen yleistyessä. Tulevaisuudessa suurin osa roskapostista lähetetäänkin saastutetuilta käyttäjien työasemilta.

Sähköpostin välityksellä leviävät virukset, madot ja muut haitalliset ohjelmat ovat myös yhä kasvava ongelma. Haitallisten ohjelmien toiminta perustuu yleensä yksittäisten sähköpostiohjelmien ja käyttöjärjestelmien turvallisuusaukkoihin tai käyttäjien tietämättömyyteen, joten kyseessä ei ole varsinainen sähköpostipalvelun ongelma. Sähköpostipalvelu itsessään ei tarjoa mitään menetelmää viruksen tartuttamiseen, vaan toimii saastuneiden tiedostojen välitysreitteinä.

Varsinaisista sähköpostipalvelun ongelmista vakavin on salauksen puute. Sähköpostipalveluissa kommunikointi palvelinten kanssa ja palvelinten välillä tapahtuu salaamattomien tekstikomentojen avulla. Jopa *autentikoinnissa* eli käyttäjän tunnistuksessa käytetty käyttäjätunnus-salasana -pari lähetetään salaamattomana. Käyttäjän tietokoneen ja palvelimen välistä liikennettä seuraava voi näin saada tietoonsa kaiken tarvittavan käyttäjän sähköpostitilin käyttämiseen haitallisiin tarkoituksiin. Tietoturvaryhmä SANS Institute pitää salaamatonta, selväkielistä (clear-text) tiedonsiirtoa yhtenä Internetin pahimmista tietoturvauhkista [SANS, 2003].

Ratkaisuksi on kehitetty SASL-kirjasto [Myers, 1997], joka tarjoaa salauksen autentikointitapahtumiin. SASL ei kuitenkaan salaa varsinaista viestien siirtoa. Parempi, mutta työläämpi ratkaisu on SSL- [Freier et al., 1996] tai TLS-protokollan [Dierks and Allen, 1999] mukaisen yhteyden salauksen käyttäminen sähköpostiliikenteessä. Tämä kuitenkin vaatii palveluntarjoajalta toimenpiteitä, joten käyttäjän mahdollisuudet vaikuttaa ongelman ratkaisemiseen ovat rajoitetut. Palveluntarjoajien tuki sähköpostiyhteyksien salaukseen vaihtelee suuresti.

Sähköpostipalveluun ei myöskään sisälly viestin sisällön salausta. Näin ollen palvelimella oleva viesti on mahdollista lukea ja viestin liitteet avata yksinkertaisilla työkaluilla. Tätä ongelmaa korjaamaan on kehitetty S/MIME-protokolla [Ramsdell, 1999], joka mahdollistaa viestin ja liitteiden salaamisen.

S/MIME ei kuitenkaan itsessään sisällä mitään salausrakenteita, vaan tyytyy määrittelemään puitteet salauksen käytölle. Salaisuuden säilyminen koko sähköpostin toimitusprosessin ajan vaatii sähköpostin salaamista jollakin ulkoisella ohjelmalla. Ulkoisen ohjelman käyttö edellyttää, että osapuolilla on käytettävissään ohjelmasta yhteensopiva versio, sekä yleensä salausavaimen luomiseen ja vaihtoon liittyviä operaatioita. Tämä johtaa siihen, että varsinaisen turvallisen kommunikaation aloittaminen vaatii *metakommunikaatiota*.

Salaukseen liittyvä viestin digitaalinen allekirjoitus viestin oikeellisuuden toteutukseksi kuuluu myös S/MIME:n piiriin. Tähänkään ei tarjota valmiita työkaluja vaan luotetaan käyttäjän aktiiviseen rooliin tietoturvan alueella. Osittain tästä syystä digitaalisten allekirjoitusten käyttäminen sähköpostissa ei ole yleistynyt.

Eräs sähköpostiprotokollien piirre, joka vaikuttaa niiden kehitykseen on se, että protokollat kehitettiin alun perin sekä ohjelmallista että suoraa manuaalista käyttöä (esimerkiksi telnet-yhteyden yli) silmällä pitäen, eli protokollat mahdollistavat myös sähköpostipalvelun käyttämisen ilman tarkoitukseen suunniteltua ohjelmaa. Tämä ei välttämättä ole ongelma, mutta nykyisin suora käyttö on niin harvinaista, että ominaisuus on käytännössä tarpeeton. Se voi kuitenkin avointen välityspalvelimien tapauksessa mahdollistaa luvattoman sähköpostin lähetyksen.

Sähköpostin viestimuoto ei, kuten eivät muutkaan tietokonejärjestelmät, alun perin tukenut muita merkistöjä kuin niinsanottua US-ASCII -merkistöä. MIME-standardin [Freed and Borenstein, 1996] myötä lisätty merkistötuki tuki on toteutettu otsikkotiedostoihin lisätyn content-type -asetuksen charset-tyypillä. Tämä ei kuitenkaan mahdollista kahden eri merkistön merkkien lisäämistä tekstiin.

Liitettäessä tiedostoja sähköpostiviestiin MIME-standardin mukaisesti tiedostot muunnetaan binäärimuodosta viestimuodon vaatimaan tekstimuotoon. Muun-nosoperaatio kasvattaa tiedoston kokoa 10-25 prosenttia, muunnosalgoritmista riippuen. Muunnettu tiedosto liitetään viestiin muun tekstin jälkeen. Tätä menetelmää luotaessa on pyritty takaamaan tekstimuotoisen siirtotavan säilyminen. Suurikokoisten liitteiden käsittely tekstimuodossa on kuitenkin epäkäytännöllistä ja liitteiden varastointi vie runsaasti tilaa.

Liitetiedostot ja sisällön salaus on lisätty viestimuodon standardiin kehityksen myöhemmässä vaiheessa. Näiden lisäysten mahdollistamiseksi tehtyjen viestimuodon muutosten vuoksi viestimuodon alkuperäinen rakenne on muutettu rakenteellisemmaksi. Muutos on kuitenkin tehty säilyttäen alkuperäinen muoto mahdollisimman pitkälle. Tämän vuoksi rakenteellisuutta

ei ole voitu toteuttaa kovin kekseliäästi. Tämä rajoittaa viestimuodon kehittämistä laajennettavuuden osalta. Uusien ominaisuuksien lisääminen sähköpostipalveluun on työlästä ja hankalaa, koska viestimuoto ei ole rakenteellisesti riittävän hyvinmuodostunut. Hyvinmuodostuneisuudella (well-formedness) tarkoitetaan tiedon tallentamista siten, että tieto on hyvin jäsennettyä ja lisäysten ja muutosten tekeminen mahdollista.

Useiden muiden palveluiden tavoin sähköpostipalvelussa käyttäjä autentikoidaan yleensä käyttäjätunnus-salasana -parilla. Tämä vakiintunut käytäntö on kuitenkin osoittautunut käytännössä hankalaksi. Salasanojen muistaminen ja riittävän monimutkaisten salasanojen keksiminen salasanojen vaihtuessa säännöllisesti aiheuttaa ongelmia. Tämä ongelma on kuitenkin yhteinen kaikkien autentikointia vaativien palveluiden kanssa.

Suurin osa sähköpostijärjestelmän ongelmista, kuten salaamattomuuteen liittyvät turvallisuusongelmat, ovat asiantuntijoiden keskuudessa tunnettuja. Koska turvallisuusaukkoja ei ole laajamittaisesti käytetty haitallisiin tarkoituksiin, ne ovat jääneet vähälle huomiolle sekä mediassa että käyttäjien keskuudessa.

4. Ongelmien ratkaiseminen

Edellä perehdyttiin sähköpostin ongelmiin ja puutteisiin. Seuraavassa tarkastelemme, millaisia menetelmiä voitaisiin käyttää ongelmien ratkaisemiseen.

Kuten edellisessä luvussa todettiin, roskaposti ei suurimmaksi osaksi ole protokollatason ongelma. Roskapostiongelman tehokkain yleinen ratkaisu olisi yhtenäisen roskapostin kriminalisoivan lainsäädännön luominen, valvonta ja riittävät sanktiot. Osaltaan ongelman torjumisessa auttaa avointen välityspalvelinten sulkeminen tai niiden käytön rajoittaminen. Avointen välityspalvelinten sulkemiseen tai niiden käytön rajoittamiseen onkin kiinnitetty runsaasti huomiota mm. roskapostia vastustavien ryhmien taholta. Uutta standardia luotaessa tämänkaltaisen ongelman syntyminen voidaan estää autentikoimalla käyttäjä jokaisen sähköpostitapahtuman yhteydessä.

Virusten ja matojen torjuntaan on olemassa useita keinoja. Saapuvan sähköpostin virustarkistus palvelimella, sähköpostiohjelmaan liitetyt virustorjuntaominaisuudet, virusherkkien ohjelmien välttäminen ja tietoturvapäivitysten nopea asentaminen ovat kaikki tehokkaita keinoja virustentorjuntaan. Itse ongelma ei, kuten edellä todettiin, kuitenkaan liity sähköpostipalveluun. Sähköpostin lukemiseen käytettävien ohjelmistojen etukäteistestaus, mahdollisten löydettyjen turva-aukkojen nopea paikkaaminen ja tehokas tiedotus auttavat tämän ongelman korjaamisessa.

Salaamattoman sähköpostiliikenteen muuttaminen turvallisemmaksi vaatii liikenteen salaamista jollain riittävällä salausmenetelmällä. Yhteyden salaamiseen olisi mahdollista käyttää jotakin SSL:n tai TLS:n kaltaista salaustekniikkaa. Näin turvattaisiin sekä autentikoinnissa tarvittava käyttäjätunnus-salasana -pari sekä varsinainen sähköpostin sisältö sähköpostitapahtumien aikana.

Sisällön salaukseen liittyvän metakommunikaatio-ongelman tehokas ratkaiseminen vaatisi sähköpostijärjestelmän yhteyteen liitettävän standardisoidun julkisen avaimen salausjärjestelmän sekä avainten noutamisen joko vastaanottajan sähköposti-palvelimelta tai erilliseltä avainpalvelimelta tai avainten siirron sähköpostiviestin yhteydessä. Näin mahdollistettaisiin salauksen käyttö ilman ulkoisia ohjelmia ja ylimääräisiä avaimenvaihto-operaatioita. Myös digitaalisen allekirjoituksen yksinkertaistaminen edellyttäisi, että sähköpostijärjestelmään liitettäisiin jokin allekirjoitusalgoritmi.

Merkistöjen ongelmaa yleisesti ratkaisemaan kehitetty Unicode-merkistö [Unicode Consortium, 2003] tarjoaa käytännöllisen ratkaisun kansainvälisyystuen ongelmaan. Unicode tarjoaa käytettäväksi yhtä aikaa kaikkien merkistöjen kirjoitus- ja erikoismerkit. Se poistaa myös tarpeen muuntaa viesti merkistöstä toiseen.

Liitetiedostojen siirtäminen tekstimuodossa ja niiden säilyttäminen osana viestiä on epäkäytännöllistä etenkin liitetiedostojen koon kasvaessa. Tiedostot olisi mahdollista siirtää erillään käyttämällä binääritiedostoille tarkoitettua siirtoprotokollaa. Tuolloin myös viestien käsittely ohjelmallisesti nopeutuisi ja viestit olisi mahdollista varastoida tehokkaammin.

Hyvinmuodostuneisuuden lisäämiseksi sähköpostin viestimuotoa on muutettava paremmin rakentuneeksi. Rakenteisen informaation esittämiseen on kehitetty erilaisia tekniikoita, joista tähän tarkoitukseen sopii esimerkiksi XML (Extended Markup Language) [Bray *et al.*, 2000]. XML:n uusin versio tukee Unicodea, joten myös kansainvälisyystuen kannalta XML-kuvauskielen käyttäminen on järkevää. XML:n käyttö mahdollistaisi liitetiedostojen ja salauksessa käytettävien avaimien sekä viestin digitaalisen allekirjoituksen erottamisen tekstiosasta. Myös muiden lisäysten tekeminen sekä olemassa olevien rakenteiden muuttaminen tulevaisuudessa olisi helpompaa.

Käyttäjän autentikointi käyttäjätunnus-salasana -parilla on standardi, jota käyttävät lähes kaikki useamman käyttäjän palvelut. Lähitulevaisuudessa teknologia tulee sallimaan myös kehittyneempien ja käyttäjäystävällisempien tunnistustekniikoiden käytön, esimerkiksi sähköistä henkilötunnistetta tai biometriikkaa soveltaen. Näiden kehittyneempien autentikointimenetelmien

käytön mahdollistaminen olisi eräs turvallisuutta ja käytettävyyttä parantava tekijä.

5. Yhteenveto

Tämänhetkisessä sähköpostipalvelussa on vakavia puutteita. Sähköpostipalvelun puutteet turvallisuudessa ja laajennettavuudessa vaativat toimenpiteitä. Nykyisten käytäntöjen päivittämissyrytykset ovat tuomittuja epäonnistumaan, sillä osaan ongelmista löytyy nyt jo ratkaisu päivitettyjen protokollaversioiden muodossa, mutta päivitettyjä versioita ei ole otettu käyttöön. Sähköpostijärjestelmien päivittäminen uusien protokollien mukaisiksi vie aikaa, eikä ole mitään takuita siitä, että päivitykset saataisiin joskus koko Internetin alueelle. Mikään ei pakota palveluntarjoajia noudattamaan standardin uusinta versiota.

Merkittävin puute on salauksen puuttuminen sähköpostiliikenteestä. Tällä hetkellä sähköpostin käyttö voi palveluntarjoajasta riippuen olla mitä tahansa melko turvallisen ja täysin turvattoman väliltä. Käyttäjän mahdollisuudet vaikuttaa oman sähköpostiliikenteensä turvallisuuteen ovat hyvin vähäiset.

Kansainvälisyystukea voidaan pitää uuden standardin jonkinlaisena vähimmäisvaatimuksena, onhan sähköposti välineenä ollut olemassa jo yli kaksi-kymmentä vuotta.

Sähköpostipalvelun kehittäminen on ongelmallista. Kankeiden standardointikäytäntöjen ja standardien hitaan käytäntönottotahdin lisäksi ongelmia tuottaa sähköpostin viestimuoto. Viestimuotoon on erittäin hankala tehdä muutoksia, joten uusien ominaisuuksien lisääminen on vaikeaa, joissain tapauksissa jopa mahdotonta. Viestimuoto tulisikin suunnitella alusta uudelleen laajennettavuuden ja rakenteisuuden periaatteita korostaen. Näin sähköpostipalvelun kehittäminen olisi siltä osin helpompaa.

Näiden edellä hahmoteltujen ongelmien ja puutteiden käytännöllisin ja nopein korjausmenetelmä olisi siis uuden protokollajoukon luominen. Kokonaan uuden protokollan luominen, standardointi ja käyttöönotto on kuitenkin pitkälinen prosessi. Prosessia voi nopeuttaa käyttämällä uuden protokollan pohjana vanhojen protokollien toimivia osia sekä mahdollistamalla osittaisen yhteensopivuuden. Uuteen sähköpostikäytäntöön siirtyminen mahdollistaisi uusien ominaisuuksien käytön niitä tarvitseville, mutta takaisi silti yhteensopivuuden vanhempien standardien (kuten viestimuodon).

Uuden sähköpostistandardin kehittämisessä oleellisia peruseriaatteita ovat turvallisuus, kansainvälisyys ja laajennettavuus. Näiden periaatteiden noudattaminen auttaa paremman sähköpostipalvelun luomisessa.

Viiteluettelo

- [Bray *et al.*, 2000] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen and Eve Maler (eds.), *Extensible Markup Language (XML) 1.0*. World Wide Web Consortium, 2000.
Available as: <http://www.w3.org/TR/REC-xml>.
- [Crispin, 2003] Mark Crispin, Internet Message Access Protocol - Version 4rev1 (RFC 3501). IETF, March 2003.
Available as <http://www.ietf.org/rfc/rfc3501.txt>.
- [Dierks and Allen, 1999] Christopher Allen and Tim Dierks, The TLS Protocol, Version 1.0. IETF, January, 1999.
Available as <http://www.ietf.org/rfc/rfc2246.txt>.
- [Freed and Borenstein, 1996] Ned Freed and Nathaniel S. Borenstein, MIME Part One (RFC 2045). IETF, November, 1996.
Available as <http://www.ietf.org/rfc/rfc2045.txt>.
- [Freier *et al.*, 1996] Alan O. Freier, Philip Karlton and Paul C. Kocher, The SSL Protocol, Version 3.0. Unpublished manuscript, 1996.
Available as <http://wp.netscape.com/eng/ssl3/draft302.txt>.
- [Klensin, 2001] John C. Klensin, Simple Mail Transfer Protocol (RFC 2821). IETF, April, 2001.
Available as <http://www.ietf.org/rfc/rfc2821.txt>.
- [Myers, 1997] John G. Myers, Simple Authentication and Security Layer (RFC 2222). IETF, October 1997.
Available as <http://www.ietf.org/rfc/rfc2222.txt>.
- [Myers and Rose, 1996] John G. Myers and Marshall T. Rose, Post Office Protocol - Version 3 (RFC 1939). IETF, May, 1996.
Available as <http://www.ietf.org/rfc/rfc1939.txt>.
- [Ramsdell, 1999] Blake Ramsdell (ed.), S/MIME Version 3 Message Specification (RFC 2633). IETF, June 1999.
Available as <http://www.ietf.org/rfc/rfc2633.txt>.
- [Resnick, 2001] Peter W. Resnick, Internet Message Format (RFC 2822). IETF, April 2001.
Available as <http://www.ietf.org/rfc/rfc2822.txt>.
- [SANS, 2003] SANS Institute, The Twenty Most Critical Internet Security Vulnerabilities. Unpublished manuscript, October 8, 2003.
Available as: <http://www.sans.org/top20/>.
- [UNCTAD, 2003] E-Commerce and Development Report 2003. United Nations, November 20, 2003.

Available as:

<http://www.unctad.org/Templates/webflyer.asp?docid=4228&intlItemID=1634>

[Unicode Consortium, 2003] Unicode Consortium, The Unicode Standard, Version 4.0.0. Addison-Wesley, 2003.

Available as: <http://www.unicode.org/versions/Unicode4.0.0/>.

Terveydenhuollon tietojärjestelmien tietoturva

Jussi Saarinen

Tiivistelmä.

Tässä tutkielmassa tarkastelen miten terveydenhuollon tietojärjestelmissä toteutetaan tietoturvaan liittyvät kysymykset. Tarkasteltavia kohtia ovat erityisesti sähköisen potilaskertomuksen toteuttamiseen ja käyttämiseen liittyvät tekijät eli esimerkiksi miten käyttäjät tunnistetaan ja varmistetaan, että tiedot ovat pysyneet muuttumattomina siirtoyhteyden aikana. Lisäksi esittelen erilaisia vaatimuksia arkkitehtuureiden toteuttamiselle, millaisia arkkitehtuurivaihtoehtoja on tarjolla sekä miten käyttäjien tunnistaminen toteutetaan terveydenhuollossa, jos käytössä on verkottunut tietojärjestelmä.

Terveydenhuollon tietojärjestelmissä PKI-arkkitehtuuri näyttää suurta roolia tietoturvasuuteen liittyvissä kysymyksissä ja sen avulla toteutetaan mm. käyttäjien tunnistaminen ja tietojen eheyden tarkistaminen siirtoyhteyksien aikana. Järjestelmää käyttävien henkilöiden tunnistamisella ja pääsynvalvonnalla on suuri merkitys verkottuneissa terveydenhuollon tietojärjestelmissä, koska pääsy sähköiseen potilaskertomukseen tulee rajoittaa vain sitä todella tarvitseviin hoitohenkilöihin. Tällöinkin tulee ottaa huomioon, että nämä henkilöt pääsevät käsiksi vain niihin tietoihin, jotka ovat tarpeellisia heidän suorittamiin hoitotoimenpiteisiin.

Nykyisin on yhä monessa sairaalassa käytössä vanhoja ns. perinnejärjestelmiä ja niiden siirtäminen nykyajan vaatimusten tasolle tulee olemaan vaikea prosessi, mm. vanhojen järjestelmien suurten tietomäärien vuoksi. Tämän lisäksi terveydenhuollon tietojärjestelmien kehitykselle määritellään erilaisia vaatimuksia esimerkiksi eri lakipykälissä, jonka vuoksi alan kehitys on hyvin hidasta ja vaativaa. Terveydenhuollon tietojärjestelmien kehityksessä tulee myös kiinnittää erityistä huomiota standardointiin, mikäli halutaan että eri järjestelmät voivat toimia tehokkaasti yhteistyössä niin kansallisella kuin kansainväliselläkin tasolla.

Avainsanat ja -sanonnat: luottamuksellisuus, terveydenhuollon tietojärjestelmä, sähköinen potilaskertomus, pääsynvalvonta, tietoturva, luottamuksellisuus, todentaminen, PKI, digitaalinen allekirjoitus, PKI-arkkitehtuuri.

CR-luokat: J.3

1. Johdanto

Terveydenhuollon tietojärjestelmiä on ollut olemassa jo yli 50 vuotta ja Suomi on ollut yksi alan edelläkävijöitä aina 1950-luvulta lähtien. Ensimmäiset terveydenhuollon sovellukset olivat talous- ja palkkahallintoon, tilinpitoon ja tilastointiin liittyviä ohjelmistoja, mutta hiljalleen järjestelmiä kehitettiin myös varsinaiseen terveydenhuoltoon liittyvien tietojen käsittelemiseen. Terveydenhuollon tietojärjestelmät ovat perinteisesti olleet hyvin erilaisia eri sairaanhoitopiirien ja jopa saman sairaalan eri osastojen kesken, jonka vuoksi yhtenäistä standardia ei alalle ole päässyt muodostumaan. Tämän lisäksi käytössä on yhä nykyisin eri-ikäisiä teknologioita ja arkkitehtuureita, joka tuo lisävaikeuksia järjestelmien yhteistoiminnan kehittämiseksi [Nykänen, 2003]. Nykyisin standardointiin kiinnitetään huomiota niin kansallisella kuin kansainväliselläkin tasolla ja esimerkiksi Euroopassa terveydenhuollon tietojärjestelmien standardoinnista vastaa CEN/TC 251 eli European Standardization of Health Informatics organisaatio [European Standardization of Health Informatics, 2003].

Tämän kaltainen tietojärjestelmien kehitys on aiheuttanut runsaasti ongelmia, koska esimerkiksi tietojen siirtäminen järjestelmästä toiseen ei ole aina ollut mahdollista toteuttaa teknisesti ja verkottuminen eri osapuolten kesken on usein ollut hyvin vaikeaa. Tämän lisäksi tietojärjestelmien kehittämistä on vaikeuttanut myös lakisääteiset vaatimukset, jotka on täytynyt ottaa huomioon. Ne määräävät mm. hyvin tiukasti ketkä saavat esimerkiksi katsella potilaan tietoja. Eri tietojärjestelmien eroavaisuudet ovat pahimmillaan saattaneet vaikeuttaa potilaan hoidon toteuttamista parhaalla mahdollisella tavalla, koska kaikkia hoidon kannalta tarpeellisia tietoja ei välttämättä ole aina ollut saatavilla. Tämän hetkisten perinnejärjestelmien (legacy systems) kehittäminen vastaamaan nykyajan vaatimuksia onkin hyvin vaikea toimenpide mm. käsiteltävänä olevan suuren tietomäärän ja vanhentuvan teknologian vuoksi. Lisäksi uusien järjestelmien kehittäminen on hyvin kallista.

Tässä tutkielmassa tulen käsittelemään miten verkottuneessa terveydenhuollon tietojärjestelmässä toteutetaan tietoturvaan liittyvät kysymykset. Tällaisessa tietojärjestelmässä huomiota tulee kiinnittää mm. siihen, miten käyttäjien todentaminen ja pääsynvalvonta toteutetaan, miten tietojen siirron aikana tarkistetaan eheys ja osapuolten kiistämättömyys sekä miten tietojen saatavuus toteutetaan eri organisaatioista käsin. Verkottuneessa tietojärjestelmässä on nykyisin usein käytössä sähköinen potilaskertomus, jonka vuoksi sitä käyttävien henkilöiden käyttöoikeuksien tarkistamiseen täytyy kiinnittää erityistä huomiota.

Luotettavan tietoturvan toteuttamiseen terveydenhuollon tietojärjestelmissä käytetään yleensä digitaalista allekirjoitusta, jonka avulla pystytään selkeästi

todentamaan käyttäjiä ja varmistamaan käytössä olevien tietojen eheys. Digitaalista allekirjoitusta käytetään toteuttamaan ns. PKI-arkkitehtuuri (Public Key Infrastructure). Käytännössä digitaalista allekirjoitusta hyödyntäviä terveydenhuollon tietojärjestelmiä on nykyisin vielä hyvin vähän olemassa koko maailman mittakaavassakin. Suomi on kuitenkin yksi digitaalisen allekirjoituksen ja PKI-arkkitehtuurin käytön edelläkävijöistä terveydenhuollon alalla ja eräs tätä hyödyntävä järjestelmä on käytössä Pohjois-Karjalan sairaanhoitopiirissä.

Tutkielmani rakentuu siten, että luku 2 esittelee sähköistä potilaskertomusta eli mikä se oikeastaan on ja luvussa 3 keskitytään pääsynvalvonnan ja tietojen eheyden tarkistamisen mahdollisiin toteutusvaihtoehtoihin. Luku 4 esittelee verkottuneen tietojärjestelmän PKI-arkkitehtuurin toteutustapoja ja tutkielman yhteenveto esitetään luvussa 5.

2. Sähköinen potilaskertomus

Tämä luku esittelee perinteisen potilaskertomuksen kehityksen ja sen nykyaikaisen version eli sähköisen potilaskertomuksen. Luvun lopussa tarkastellaan lisäksi myös sähköisen potilaskertomuksen toteutuksessa huomioitavia tietoturvaan liittyviä tekijöitä.

2.1. Perinteinen potilaskertomus

Potilaskertomukseen on kiinnitetty huomiota Suomessa jo 1970-luvulta lähtien, jolloin sen kehittäminen käytännössä aloitettiin. Tällöin kehitettiin sen perusominaisuudet eli hoidon kirjaamisen, tiedon käytön ja säilyttämisen periaatteet. Suomessa ensimmäinen potilaskertomus otettiin käyttöön vuonna 1974, jolloin sen jatkokehityksen tavoitteiksi asetettiin mm. että potilaan välitön hoito on turvattava, sairauskertomusta on voitava käyttää myös myöhemmin samaa potilasta hoidettaessa ja että potilaskertomukseen kirjattuja tietoja on voitava käyttää esimerkiksi hallinnon kehittämisen apuna [Hartikainen *et al*, 2000].

Hyvä potilaskertomus on esimerkiksi tietokanta, johon talletetaan potilaan hoitoon liittyvät tiedot, kuten vaikkapa potilaan sairauteen ja sen hoitoon liittyvät toimenpiteet. Potilaskertomus on keskeinen työväline terveydenhuollon piirissä työskenteleville ja hyvin toteutettuna sen avulla voidaan parantaa hoidon suunnittelua ja toteutusta sekä hoitoyhteistyön hallintaa siten, että potilaskertomuksesta löytyvät kaikki hoitoon osallistuvat tahot, jolloin hoidon suunnittelu voidaan toteuttaa eri osapuolten kesken. Myös ajankäytön hallinta helpottuu, kun toteutettava hoito voidaan suhteuttaa potilaan aikaisempiin hoitoihin ja hänen nykyiseen elämäntilanteeseen.

2.2. Potilaskertomuksen siirtyminen sähköiseen muotoon

Siirtyminen perinteisestä potilaskertomuksesta sähköiseen potilaskertomukseen mahdollistaa monia potilaan hoitoa helpottavia asioita. Perinteisen ns. manuaalikertomuksen eli käsin kirjoitetun potilaskertomuksen pituutta rajasi usein käytetyn lomakkeen pituus. Sähköisen potilaskertomuksen etuna on, että tällainen tilanne ei ole enää ongelma. Myös sähköisessä muodossa olevien erilaisten tietojen yhdistely on mahdollista, jolloin potilasta koskevia tietoja voidaan helposti yhdistellä eri paikoista uusilla tavoilla. Aiemmin tämä oli käytännössä mahdotonta. Tämän lisäksi sähköisen potilaskertomuksen merkittävä etu on, että sähköisessä muodossa oleva tieto on heti nähtävillä koko organisaatiossa, jolloin hoidon suunnittelu ja toteutus on ainakin teoriassa helpompaa toteuttaa [Hartikainen *et al*, 2000].

Erilaisin rakenneratkaisuin voidaan rajata tiedon tarpeetonta tuottamista, jota perinteisessä potilaskertomuksessa välttämättä syntyy esimerkiksi potilaan käydessä eri sairaaloissa hoidettavana. Tämä vähentää sekä kustannuksia että antaa lisää aikaa varsinaisiin potilaiden hoitamisiin liittyviin toimintoihin. Sähköisessä muodossa olevaa tietoa voidaan paremmin analysoida ja arkistoida, jolloin se mahdollisesti parantaa hoidon suunnittelua, ohjausta ja toteutusta. Potilaisiin liittyvien tietojen tarpeeton tuottaminen ja tietojen käsittelyn vähentyminen parantavat myös potilaiden tietosuojaa ja heidän oikeusturvaansa.

Sähköinen potilaskertomus muodostuu ydinkertomuksesta ja perustason kertomuksesta. Ydinkertomus sisältää potilaan henkilötiedot sekä erinäisiä tietoja potilaan keskeisistä terveydenhoitoon liittyvistä toimista. Sen tarkoitus on antaa kokonaiskuva potilaan terveys- ja sairaushistoriasta, joka täytyy ottaa huomioon esimerkiksi uutta hoitoa suunniteltaessa. Perustason kertomus puolestaan sisältää terveydenhuollon ammattihenkilöiden tekemät hoidon suunnitteluun, toteutukseen ja arviointiin liittyvät merkinnät, jotka voivat olla mm. eri ammattiryhmien tutkimus- ja hoitosuunnitelmat tai sairauden yksityiskohittaiset seurantasuunnitelmat [Hartikainen *et al*, 2000].

2.3. Tietoturva sähköisessä potilaskertomuksessa

Sähköisen potilaskertomuksen avulla voidaan saavuttaa parempi tietoturva kuin perinteisellä potilaskertomuksella, koska potilaaseen liittyvät hoitotiedot voidaan salata esimerkiksi käyttäjätunnuksilla tai käyttämällä erilaisia salaustekniikoita. Potilaiden tietoihin ei saa päästää ulkopuolisia henkilöitä tai muita henkilöitä, jotka eivät ole hoitosuhteessa potilaaseen.

Potilastietojen saatavuus tuleekin järjestää siten, että tietyille ammattiryhmille annetaan vain hänen tehtävän hoitamiseen tarpeelliset tiedot. Sähköisessä

potilaskertomuksessa tämä on kohtalaisen helppo toteuttaa, kun taas vastaavan kaltainen rajausta perinteisessä manuaalisessa kertomuksessa olisi lähes mahdotonta. Tietoturvan toteuttaminen ja valvonta on siis paljon helpompaa saavuttaa sähköisessä potilaskertomuksessa, tosin huonosti toteutettuna sähköinen potilaskertomus saattaa mahdollistaa aiempaa laajemman potilastietojen laittoman hankkimisen.

Potilaan hoidon aikana syntyy myös monenlaista tietoa niin hoitoon kuin yleiseen hallintoon liittyen. Onkin tärkeää, että nämä eri tiedot sijoitetaan eri rekistereihin, koska esimerkiksi hallinnollista tietoa ei tarvitse useinkaan säilyttää yhtä kauan kuin hoitoon liittyvää tietoa [Hartikainen *et al*, 2000]. Potilastietojen käsittelyä ohjaavat monet eri lakisääteiset säädökset. Terveystieteiden tietojärjestelmiä suunniteltaessa huomioon tuleekin ottaa mm. laki potilaiden oikeuksista, henkilökisterilaki, henkilötietolaki, laki viranomaisen toiminnan julkisuudesta, arkistolaki, henkilökorttilaki sekä sähköisen asioinnin laki [Sosiaali- ja terveysministeriö, 1998a].

3. Pääsynvalvonta ja tietojen eheyden tarkistus

Tämä luku käsittelee terveydenhuollon pääsynvalvonnan ja tietojen eheyden tarkistamisen toteutusvaihtoehtoja, joka nykyaikaisissa tietojärjestelmissä käytännössä tarkoittaa PKI-arkkitehtuurin käyttämistä.

3.1. Pääsynvalvonnan toteuttaminen

Terveystieteidenhuollossa suurin osa kanssakäymisestä tapahtuu yhä kasvotusten esimerkiksi potilaan ja lääkärin välillä, jolloin osapuolten tunnistaminen on yleensä ongelmattonta. Erityisesti terveydenhuollon ammattilaisten välinen keskustelu tapahtuu kuitenkin jo nyt hyvin usein käyttäen tietoliikenteen mahdollistamia palveluja, esimerkiksi sähköpostitse, jolloin huomiota täytyy kiinnittää tietoturvaan liittyviin kysymyksiin. Erityisesti sähköisen potilaskertomuksen ollessa käytössä täytyy tietojärjestelmän suunnittelussa kiinnittää huomiota pääsynvalvonnan toteuttamiseen, koska mm. lakisääteiset tekijät rajaavat tiukat toimintarat. Tietoverkoissa tämän toteuttamiseen on käytössä monia eri tekniikoita ja ns. heikkoja ja vahvoja tunnistusmenetelmiä. Käytettävä tunnistusmenetelmä valitaan sen perusteella, kuinka turvallisesti järjestelmä halutaan toteuttaa.

Yleisesti ottaen heikko tunnistus tapahtuu käyttäjien tunnistamisella käyttäjätunnuksien avulla, kun taas vahva tunnistaminen eli käyttäjän todentaminen toteutetaan tunnistetietoon liittyvän salasanan tai PKI-arkkitehtuurin (Public Key Infrastructure, julkisen avaimen järjestelmä) avulla. Tunnistamisessa siis yksilöidään henkilö tietyksi käyttäjäksi ja todentamisessa vakuutu-

taan yksilöinnin oikeellisuudesta. Osapuolten todentamiseen käytetään yleensä juuri PKI-arkkitehtuuria, jossa vahva todentaminen on siihen liittyvä perustoiminto [Ruotsalainen, 2002].

3.2. PKI-arkkitehtuuri

Whitfield Diffie ja Martin E. Hellman esittelivät vuonna 1976 PKI-arkkitehtuurin, joka perustuu epäsymmetriseen salaukseen eli kahteen eri matemaattiseen avaimen [Diffie and Hellman, 1976].

Salausavaimet voidaan kryptografiassa käytännössä toteuttaa kahdella eri tavalla, käyttäen joko symmetrisiä tai epäsymmetrisiä avaimia. Symmetristä salausta kutsutaan myös salaisen avaimen algoritmiksi sekä yhden avaimen algoritmiksi. Nämä nimitykset johtuvat siitä, että symmetrisessä salauksessa sekä tiedon salaus että purkaminen toteutetaan samalla avaimella, jolloin suurimmaksi ongelmaksi syntyy avaimen välittäminen luotettavalla tavalla osapuolten kesken turvallista reittiä käyttäen. Symmetrinen salaus on ollut käytössä hyvin pitkään ja tunnetuin sitä käyttävä salausmenetelmä on ns. Ceasar-salaus, joka perustuu merkkien korvaukseen sovitun kaavan mukaisesti. Epäsymmetrinen salaus puolestaan perustuu kahteen eri avaimen, joista toinen on julkinen ja toinen salainen. Viestin salaus tehdään käytännössä siten, että yhdellä avaimella voidaan salata haluttu viesti ja toisella avaimella purkaa salattu viesti takaisin salaamattomaan muotoon. Kummastakaan avaimesta ei voida selvittää laskemalla toista avainta. Tunnetuin epäsymmetrinen salausalgoritmi on RSA, joka perustuu suurien kokonaislukujen alkutekijöihin jakamisen vaikeuteen [Kerttula, 1998].

PKI-arkkitehtuuria käytetään paitsi tiedon luottamuksellisuuden turvaamiseen, myös käyttäjien tunnistukseen sekä automaattiseen avainten vaihtoon. Se on kuitenkin hidas verrattuna symmetristä salausta hyödyntäviin järjestelmiin ja avainten pituudet ovat huomattavasti pitempiä kuin symmetrisessä salauksessa, mutta toisaalta sen avulla saadaan toteutettua paljon joustavammin useita turvallisuusratkaisuja. Muita PKI-arkkitehtuurin etuja verrattuna symmetrisen salauksen järjestelmiin ovat avainten hallitsemisen helppous ja se, ettei avaimia tarvitse vaihtaa kovinkaan usein [Menezes *et al*, 1996]. PKI-arkkitehtuurin käytännön sovellukset ovat viime vuosiin saakka olleet vain teoreettisia, koska alan standardointi on ollut kesken eikä sitä hyödyntäviä tuotteita ole ollut saatavilla [Kerttula, 1998].

3.3. Digitaalinen allekirjoitus ja PKI-arkkitehtuuri

Digitaalinen allekirjoitus on sähköisesti salausalgoritmein toteutettu allekirjoitus, jonka toteuttamiseen on käytetty PKI-arkkitehtuuria [Viestintävirasto, 2003]. Digitaalisen allekirjoituksen tavoite on varmistaa tiedon eheys ja että lä-

hetetty viesti on tullut myöhästyttä sekä että sen on lähettänyt oikea taho [Stallings, 1996].

Digitaalinen allekirjoitus vastaa Suomessa täysin perinteistä allekirjoitusta ja laki sähköisistä allekirjoituksista tuli voimaan 1.2.2003. Sähköisen allekirjoituksen laki toteuttaa Euroopan parlamentin ja neuvoston direktiivin (1999/93/EY) ja sen tarkoituksena on edistää sähköisten allekirjoitusten käyttöä, niihin liittyvien tuotteiden ja palvelujen tarjontaa sekä sähköisen kaupankäynnin ja sähköisen asioinnin tietosuoja ja tietoturva [Viestintävirasto, 2003].

Seberry ja Pieprzyk esittävät viisi vaatimusta, jotka digitaalisen allekirjoituksen täytyy toteuttaa [Seberry and Pieprzyk, 1989]. Nämä vaatimukset ovat seuraavat:

- ainutlaatuisuus eli että vain digitaalisen allekirjoituksen omistaja voi luoda saman allekirjoituksen,
- väärentämättömyys eli että avaimen väärentämisen on oltava mahdollonta,
- autentikoinnin helppous eli että vastaanottajan täytyy helposti saada selville kuka allekirjoituksen omistaja on,
- kiistämättömyys eli että avaimen omistaja ei voi jälkeempään kiistää tekemäänsä allekirjoitusta, ja
- toteuttamisen helppous eli että allekirjoituksen toteuttaminen on mahdollisimman helppoa ja halpaa.

Näiden ominaisuuksien lisäksi digitaalisen allekirjoituksen tulee mahdollistaa tiedon eheyden tarkistaminen eli se, ettei siirrettyä tietoa ole muutettu siirtoyhteyden aikana [Ruotsalainen, 2002].

Digitaalisen allekirjoituksen käyttäminen perustuu siis suurelta osin luottamukseen, jossa oletetaan että vain avaimen omistajalla on hallussaan salainen avain ja täten vain hän voi allekirjoittaa viestin tällä avaimella. Toisaalta myös vastaanottaja luottaa siihen, että hänellä on oikea lähettäjän julkinen avain käytössään. Julkisten avainten hallinnointiin käytetään yleensä luotettua kolmatta osapuolta (TTP, Thrusted Third Party) eli varmenneviranomasta, joka ylläpitää mm. rekisteriä käyttäjien varmenteista sekä sulkulistaa, johon merkitään käytöstä suljetut varmenteet [Sosiaali- ja terveysministeriö, 1998b].

Varmenne sisältää mm. varmenteen omistajan yhteystiedot ja hänen julkisen avaimen sekä mahdollisesti muita lisätietoja [Viestintävirasto, 2003]. Yleisin käytössä oleva varmennetyyppi on International Telecommunication Unionin X.509 mukainen varmenne. Se on standardoitu, mutta hyvin raskas toteuttaa ja siten huonosti suunniteltu menetelmä tietojärjestelmissä käytettäväksi. Muita

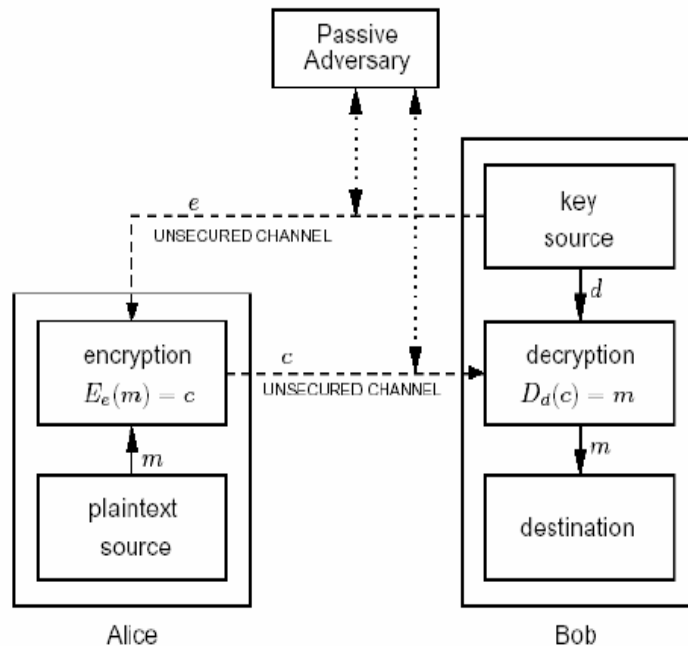
mahdollisia toteutusmenetelmiä ovat mm. SPKI- ja PGP-varmennetyypit, joiden standardointi on tosin vielä kesken. Varmenneviranomaisen varmentaa häntä hierarkiassa korkeammalla oleva varmenneviranomainen, jolloin varmenteiden myöntäjistä muodostuu puumainen hierarkia. Puun ylimpänä on ns. juurivarmentaja, joka varmentaa itse itsensä. Eri toiminta-alueiden varmenneviranomaiset harjoittavat usein ns. ristiinvarmennusta, jolloin varmenteiden toiminta-alueita voidaan helposti laajentaa [Kerttula, 1998].

Digitaalinen allekirjoitus toteutetaan käytännössä siten, että viestin lähettäjä luo tiivisteestä tietyntiivistealgoritmilla mukaisesti. Tiiviste on kuin lyhennetty versio varsinaisesta viestistä ja se salataan lähettäjän salaisella avaimella. Näin on luotu varsinaisen digitaalisen allekirjoituksen. Kun viesti on lähetetty ja käyttäjä vastaanottaa sen, laskee hän oman tiivisteensä samalla tiivistealgoritmilla kuin lähettäjäkin. Tämä laskemiseen käytettävä tiivistealgoritmi tulee käyttäjän varmenteen mukana. Tämän jälkeen vastaanottaja purkaa digitaalisen salauksen lähettäjän julkisella avaimella, josta tulee tulokseksi lähettäjän laskema tiiviste. Seuraavaksi vastaanottaja vertailee näitä kahta eri tiivistettä keskenään ja mikäli ne ovat täysin samanlaiset, on viesti aito, saatu oikealta lähettäjältä ja sen eheys on oikea [Goldwasser *et al*, 1995].

3.4. PKI-arkkitehtuurin soveltaminen käytännössä

PKI-arkkitehtuuria voidaan käyttää esimerkiksi seuraavalla tavalla:

1. Käyttäjät, esimerkiksi Alice ja Bob, luovat avainparin, jossa syntyy sekä julkinen että salainen avain.
2. Molemmat käyttäjät sijoittavat julkisen avaimensa luotetun kolmannen osapuolen ylläpitämään julkiseen varmennerekisteriin, josta kuka tahansa käyttäjä voi hakea haluamansa henkilön varmenteen. Salaiset avaimensa molemmat käyttäjät pitävät itsellään.
3. Jos esimerkiksi Alice haluaa lähettää Bobille sähköisen potilaskertomuksen, pyytää hän varmennerekisteristä Bobin varmenteen, joka sisältää mm. Bobin julkisen avaimen. Alice salaa tällä Bobin julkisella avaimella lähetettävän potilaskertomuksen sekä lähettää sen Bobille turvaamattomalla kanavalla pitkin.
4. Kun Bob vastaanottaa Alicen lähettämän potilaskertomuksen, voi hän avata sen salauksen käyttämällä omaa salaista avaintaan. Näin kukaan muu kuin Bob ei voi lukea potilaskertomusta, vaikka saisikin sen kaapatuksi turvaamattoman siirtoyhteyden aikana.



Kuva 1: Alicen ja Bobin välinen viestiliikenne järjestelmässä, jossa käytetään PKI-arkkitehtuuria [Menezes *et al*, 1996].

PKI-arkkitehtuurin tehokas soveltaminen vaatii järjestelmän läpinäkyvyyttä, sillä salausmenetelmien käytön tulee olla mahdollisimman helppoa loppukäyttäjille. Käytännössä tämä tarkoittaa sitä, että loppukäyttäjien ei tarvitse tietää miten PKI-arkkitehtuuri salaustoimenpiteet toteuttaa. Nykyisin esimerkiksi käyttäjien salainen avain onkin sijoitettuna sähköiseen henkilökorttiin, jolloin sen käyttäminen on hyvin yksinkertaista eikä käyttäjien tarvitse huolehtia salauksen teknisestä toteutuksesta.

Suomessa henkilökorttien käyttäminen on ollut mahdollista vuodesta 1999 alkaen, jolloin voimaan tuli henkilökorttilaki [Heimala ja Vestama, 2003]. Sähköistä henkilökorttia voidaan käyttää hallinnon asiointissa mikäli siihen liitetään Väestörekisterikeskuksen myöntämä varmenne. Sähköiseen henkilökorttiin talletetaan tietoja mm. varmenteen myöntäjästä, haltijan nimestä ja sähköisestä asiointitunnuksesta, kortin voimassaoloajasta, myöntäjän maatumuksesta, varmenteen sarjanumerosta sekä tiedot noudatettavasta varmennepolitiikasta ja varmenteen käyttötarkoituksesta. Organisaatiokohtaiset henkilökortit ovat yleensä voimassa 2-5 vuotta [Väestörekisterikeskus, 2003]. Sähköiset henkilökortit ovat nykyisin hyvin standardoituja ja niitä voidaan käyttää yleisimmissä käyttöjärjestelmissä. Henkilökorttien käyttäminen vaatii lisäksi joko korttikohtaisen PIN-koodin tai henkilökohtaisen salasanan [Sosiaali- ja terveysministeriö, 1998b].

Esa Kerttula esittää läpinäkyvyyden lisäksi PKI-arkkitehtuurin tehokkaalle toteuttamiselle seuraavat vaatimukset [Kerttula, 1998]:

- varmenteiden hallinnoinnin tehokkaan järjestämisen,
- varmenteiden säilytyspaikan toteuttamisen,
- varmenteiden voimassaolon kumoamisen tarvittaessa,
- käyttäjien avainten varmuuskopioinnin ja palauttamisen virhetilanteissa,
- tuen digitaalisten allekirjoitusten kiistämättömyydelle,
- automaattisen avainparien ja varmenteiden päivittämisen,
- avainhistorian ylläpidon,
- tuen ristiinvarmennukselle, sekä
- PKI-arkkitehtuurin kanssa yhteistyössä toimivan loppukäyttäjien työaseman ohjelmiston toteuttamisen.

4. Verkottunut terveydenhuollon tietojärjestelmä

Tässä luvussa käsitellään miten verkottunut terveydenhuollon tietojärjestelmä voidaan rakentaa, millaisia eri arkkitehtuurityylejä on käytettävissä ja millaisia vaatimuksia sen toteuttamiselle asetetaan.

4.1. Arkkitehtuurille asetetut vaatimukset

PKI-arkkitehtuurin rakentamisessa täytyy ottaa huomioon monia erilaisia vaatimuksia, jotka voivat olla joissain tilanteissa jopa keskenään ristiriidassa. Tällaisten järjestelmien tulee olla mm. skaalautuvia, jolloin saman järjestelmän käyttämiä varmenteita voidaan käyttää eri osapuolten välisessä kommunikoinnissa ja tällä tavoin minimoida loppukäyttäjille joustavuuteen liittyviä ongelmia. Järjestelmissä tulee myös olla tuki monille eri sovelluksille, koska tämä on järkevää jo mm. turvallisuus- ja kustannussyistä [Kerttula, 1998]. Näiden vaatimusten lisäksi huomiota tulisi kiinnittää eri varmenneviranomaisten väliseen yhteistyöhön sekä monien varmennepolitiikkojen tukemiseen, joilla voidaan parantaa eri järjestelmien yhteistoimintaa ja käytön sujuvuutta.

Suomessa sosiaali- ja terveystieteiden ministeriö on asettanut tavoitteeksi ns. saumattomien palveluketjujen kehittämisen, joka tarkoittaa terveydenhuollon palveluprosessien saannin organisaatorajoista riippumatta. Erilaisia organisaatorajoja ovat esimerkiksi julkisen ja yksityisen terveydenhuollon välinen raja. Nykyisin käytössä olevat järjestelmät on rakennettu ensisijaisesti organisaatiokeskeiseksi, jolloin ne eivät pysty tukemaan kovin hyvin tätä palveluketjumallia. Tämän mallin lisäksi tarvitaan mm. organisaatorajat

ylittävää tietoturvapoliittikkaa, infrastruktuuria sekä kaikki ammattiryhmät kattavaa tietoturvakoulutusta [Ruotsalainen, 2002].

4.2. PKI-arkkitehtuurivaihtoehdot

PKI-arkkitehtuurin mahdolliset toteutusvaihtoehdot vaihtelevat lähinnä keskitetyn ja hajautetun arkkitehtuurin välillä. Käytännössä vaihtoehdot ovat keskitetty, osittain keskitetty tai täysin hajautettu arkkitehtuuri. Varmentajat voivat eri arkkitehtuureissa myöntää varmenteita joko tarkkaan ennalta sovitulla hierarkkisella tavalla tai joustavammalla, vähemmän järjestäytyneellä tavalla [Kerttula, 1998].

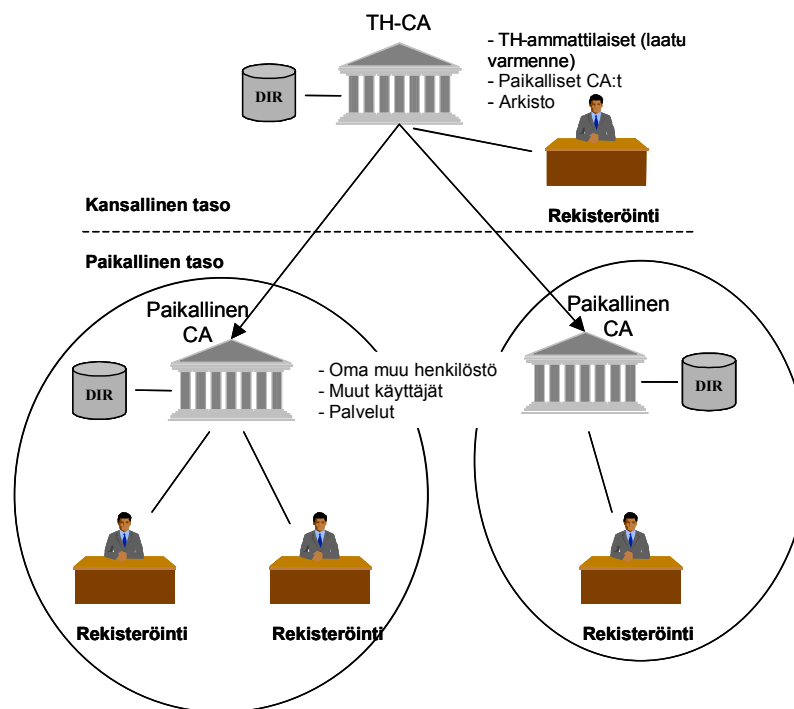
Keskitetyssä eli hierarkkisessa arkkitehtuurissa varmennepalvelut toteutetaan keskitetysti kansallisen varmenneviranomaisen kautta. Tällöin varmennepolitiikka ja muut palvelut myönnetään kaikille asiakkaille samoilla periaatteilla [Ruotsalainen, 2002]. Keskitetyn arkkitehtuurin hyviä puolia ovat yhteensopivuus, varmennepolun yksinkertaisuus sekä alhaiset kustannukset. Toisaalta mahdollisia haittoja ovat järjestelmän massiivisuus, kaikkien osapuolten vaatimusten huomioimisen vaikeus, kustannusten jakaminen osapuolten kesken sekä muutosten tekemisen vaikeus. Keskitetty arkkitehtuuri ei myöskään sovellu kovin hyvin nykyiseen terveydenhuollon toimintamalliin, jossa tällä hetkellä järjestelmiä kehitetään itsenäisesti eri sairaaloissa. Lisäksi järjestelmän juurivarmentajan salaisen avaimen murtaminen vaikuttaisi koko järjestelmän tietoturvallisuuteen merkittäväällä tavalla [Kerttula, 1998].

Osittain keskitetyssä arkkitehtuurissa toiminnot jaetaan keskitettyihin ja paikallisiin palveluntarjoajiin, jolloin varmenneviranomainen jakaa toimivaltaa organisaatioille. Jokaiseen organisaatioon tulee oma varmenneviranomainen, joka myöntää organisaatiokohtaisia varmenteita. Juurivarmentaja tarjoaa lisäksi mm. varmenne- ja korttituotantoa. Tämän arkkitehtuurin etuja ovat mm. varmennepolun yksinkertaisuus, organisaatioiden itsenäisyys, uusien palveluiden käyttöönoton helppous sekä kustannusten helppo hallittavuus. Haittoja puolestaan ovat suuri kontrollin tarve sekä suuremmat kokonaiskustannukset kuin esimerkiksi keskitetyssä mallissa [Ruotsalainen, 2002].

Täysin hajautetussa arkkitehtuurissa kaikki toiminnot toteutetaan paikallisesti, jolloin yhteisen varmennepolitiikan käyttäminen tulee todella tärkeään rooliin. Tällöin järjestelmästä syntyy verkkomainen rakennelma. Paikalliset varmenneviranomaiset voivat lisätä toimialaansa ristiinvarmentamalla toisten varmenneviranomaisten kanssa [Kerttula, 1998]. Tämän arkkitehtuurin etuja ovat mm. että jokainen varmenneviranomainen voi toimia itsenäisesti, uusien palveluiden käyttöönotto on nopeaa sekä kustannuksia voidaan hallita helposti. Arkkitehtuurin heikkouksia puolestaan ovat, että jokaisen varmenneviranomaisen täytyy toteuttaa laatuvarmentajalle asetetut tekniset ja toiminnalli-

set vaatimukset, jolloin sekä organisaatiokohtaiset kustannukset että järjestelmän kokonaiskustannukset nousevat korkeammiksi kuin keskitetyissä arkkitehtuureissa. Lisäksi varmennepolun muodostuminen ja todentaminen on vaikeampaa toteuttaa, koska varmenneviranomaisten toiminta muuttuu monelta osin monimutkaisemmaksi.

Suomessa sosiaali- ja terveysministeriön asettaman kansallisen terveysprojektin tavoitteena on ollut määrittellä kansallinen tietoturvallinen arkkitehtuuri terveydenhuollon tietojärjestelmille [Sosiaali- ja terveysministeriö, 2003]. Määrittely on vielä kesken, mutta on todennäköistä että valinnassa tullaan päätyämään kaksitasoiseen, osittain keskitettyyn arkkitehtuuriin. Tässä mallissa valtakunnallinen varmenneviranomainen luom. organisaatiossa käytettävän tietoturvapoliittikan, varmentaa alueelliset varmenneviranomaiset ja palveluntuottajat sekä toteuttaa kansainväliset ristiinvarmennukset. Alueelliset varmenneviranomaiset puolestaan vastaavat mm. kansallisen tietoturvapoliittikan toteuttamisesta sekä hallitsevat ja ylläpitävät varmennerekistereitä.



Kuva 2: ehdotus Suomessa käytettäväksi kansalliseksi PKI-arkkitehtuuriksi [Ruotsalainen, 2002].

Tämän arkkitehtuurin vahvuuksia ovat mm. että se kuvaa Suomessa nykyisin käytössä olevaa terveydenhuollon toimintamallia ja kaikilla osapuolilla on mahdollisuudet toteuttaa järjestelmänsä haluamallaan tavalla, jolloin kustannustenhallinnointi on helppoa [Ruotsalainen, 2002].

5. Lopuksi

Tutkielmassani käsittelin verkottuneen tietojärjestelmän tietoturvaan liittyviä kysymyksiä, jossa huomiota kiinnitettiin pääasiassa PKI-arkkitehtuurin toteuttamiseen. Terveystieteiden tietojärjestelmien kehittäminen tulee tulevaisuudessa yhä tärkeämmäksi, kun eri sairaanhoitopiirien järjestelmiä yritetään saada toimimaan yhdessä nykyistä paremmin. Tietoturvaan liittyvät kysymykset tulevat myös entistä tärkeämpään rooliin, kun eri järjestelmät toimivat keskenään tietoverkkojen avulla.

Tietoturvan toteuttamiseen tulee kiinnittää erityistä huomiota ja se vaatii paljon käytännön osaamista, suunnittelua ja standardointia. Eri järjestelmien yhteensovittaminen ei kuitenkaan tapahdu kovin nopeasti, koska nykyisten perinnejärjestelmien sisältämien tietojen siirtäminen uusiin järjestelmiin ei tule olemaan kovin yksinkertainen operaatio. Tämä johtuu mm. käytössä olevasta vanhasta teknologiasta sekä käsiteltävien tietojen suuresta lukumäärästä. Osa tiedoista on myös olemassa vain paperiversioina, jolloin niiden mahdollinen digitoiminen täytyisi myös hoitaa kuntoon. Suurin hidastava tekijä tulee kuitenkin todennäköisesti olemaan korkea hinta, koska uusien järjestelmien kehittäminen tulee olemaan hyvin kallista ja halukkaita maksamiehiä ei varmaankaan kovin helposti löydetä. Kehittäminen tulisikin hoitaa keskitetysti valtion taholta, jolloin Suomeen saataisiin käyttöön yhtenäinen järjestelmä.

PKI-arkkitehtuuria käyttävien järjestelmien yleistymisen tärkeimpänä edellytyksenä voidaan pitää yhteisesti sovittujen standardien kehittämistä, koska vain siten voidaan erilaiset järjestelmät saada toimimaan yhdessä tehokkaalla tavalla. Tällainen toimintamalli vähentää kustannuksia ja lisää luotettavuutta eri osapuolten välillä. PKI-arkkitehtuuriin perustuvien järjestelmien ylläpitäjillä tulee myös olla ymmärrys järjestelmään kohdistuvista riskeistä sekä miten riskien jakaminen eri osapuolten kesken toteutetaan.

Järjestelmien kehittämisessä tulee kiinnittää myös huomiota loppukäyttäjien tarpeisiin ja kehittää esimerkiksi käyttöliittymät mahdollisimmat helppokäyttöisiksi. Tämä on erityisen tärkeää, koska järjestelmien loppukäyttäjät eivät todennäköisesti tule olemaan tietotekniikan ammattilaisia tai omaamaan aiempaa kokemusta tietojärjestelmien käytöstä.

Viiteluettelo

[Diffie and Hellman, 1976] Whitfield Diffie and Martin E. Hellman, New Directions in Cryptography. *IEEE Transactions on Information Theory*, 1976. <http://citeseer.nj.nec.com/diffie76new.html>, tarkistettu 23.11.2003.

- [European Standardization of Health Informatics, 2003]. European Standardization of Health Informatics järjestön kotisivut. <http://www.centc251.org/>, tarkistettu 14.12.2003.
- [Goldwasser *et al*, 1995] Shafi Goldwasser, Silvio Micali and Ronald L. Rivest, A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM Journal on Computing*, 1995. <http://citeseer.nj.nec.com/goldwasser88digital.html>, tarkistettu 14.12.2003.
- [Hartikainen *et al*, 2000] Kauko Hartikainen, Anita Kokkola ja Ritva Larjomaa, Elektronisen potilaskertomuksen sisältömääritykset. *Osaavien keskusten verkoston julkaisuja*, 2000. <http://hosted.kuntaliitto.fi/skriptit/tyk/Docs/Elekt-potilas-julk.pdf>, tarkistettu 26.10.2003.
- [Heimala ja Vestama, 2003] Heli Heimala ja Riina Vestama, *Sähköinen Asiointi Hallinnossa*. Helia, 2003.
- [Kerttula, 1998] Esa Kerttula, *Tietoverkkojen Tietoturva: Tarpeet ja Teknologiat, Internet-tietoturva, Suljettujen IP - Verkkoyöjen Tietoturva, Julkisen Avaimen Infrastrukturi*. Liikenneministeriö, 1998.
- [Menezes *et al*, 1996] Alfred Menezes, Paul van Oorschot ja Scott Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996. <http://www.cacr.math.uwaterloo.ca/hac/>, tarkistettu 31.10.2003.
- [Nykänen, 2003] Pirkko Nykänen, Terveystenhuollon tietöjärjestelmät seminaarin materiaali. <http://www.cs.uta.fi/is/terveysseminaari.html>, tarkistettu 23.11.2003.
- [Ruotsalainen, 2002] Pekka Ruotsalainen, Ehdotus sosiaali- ja terveydenhuollon sähköisen asioinnin arkkitehtuuriksi - terveydenhuollon PKI arkkitehturi. *Osaavien keskusten verkoston julkaisuja*, 2002. <http://www.oskenet.fi/uploads/a5z5nyx9nztg.doc>, tarkistettu 26.10.2003.
- [Seberry and Pieprzyk, 1989] Jennifer Seberry and Josef Pieprzyk, *Cryptography – an Introduction to Computer Security*. Prentice hall, 1989.
- [Sosiaali- ja terveysministeriö, 1998a] Sosiaali- ja terveysministeriö, Sosiaali- ja terveydenhuollon tietoteknologian hyödyntäminen, Osa II Tietosuoja ja tietoturva. *Sosiaali- ja terveysministeriö*, 1998. <http://www.oskenet.fi/uploads/bj2ygh8mt.pdf>, tarkistettu 26.10.2003.
- [Sosiaali- ja terveysministeriö, 1998b] Sosiaali- ja terveysministeriö, Sosiaali- ja terveydenhuollon tietoteknologian hyödyntäminen. Osa II Tietosuoja ja tietoturva – raportin liite: tietoturvateknologian yleiskatsaus. *Sosiaali- ja terveysministeriö*, 2003. <http://www.oskenet.fi/uploads/u7rwcucqqptgb.pdf>, tarkistettu 26.10.2003.
- [Sosiaali- ja terveysministeriö, 2003] Sosiaali- ja terveysministeriön kansallisen terveydenhuoltoprojektin kotisivut.

<http://www.stm.fi/Resource.phx/hankk/hankt/terveysprojekti.htx>, tarkistettu
15.12.2003.

[Stallings, 1996] William Stallings, *Practical Cryptography for Data Internetworks*.
IEEE Computer Society Press, 1996.

<http://www.stm.fi/Resource.phx/hankk/hankt/terveysprojekti.htx>, tarkistettu
15.12.2003.

[Viestintävirasto, 2003] Viestintäviraston kotisivut tietoturvasta.
<http://www.ficora.fi/suomi/tietoturva/>, tarkistettu 31.10.2003.

[Väestörekisterikeskus, 2003] Väestörekisterikeskuksen ylläpitämät kotisivut
sähköisestä henkilökortista. <http://www.sahkoinenhenkilokortti.fi/>, tarkistettu
14.12.2003.

Kevyt agenttialusta heterogeenisessä ympäristössä

Rami Saarinen

Tiivistelmä

Hajautettujen järjestelmien kehittyminen ja agenttiohjelmoinnin esiinmarssi mahdollistaa uusien ohjelmointimetodologioiden käyttöönoton, kuten agenttiohjelmoinnin. Vuosien varrella on rakennettu useita, menestyksekkäitäkin, agenttialustoja. Useammat alustat kuitenkin on rakennettu jonkin tietyn käyttöjärjestelmän tai ohjelmointikielen varaan. Vastaavasti ollessaan kieli- ja käyttöjärjestelmäriippumattomia, agenttialustat ovat monimutkaisia ja rakenteeltaan epäselviä.

PROACT hankkeen yhteydessä syntyi tarve yleiskäyttöiselle agenttialustalle, joka olisi helposti käytettävä ja laajennettava. Alustan tulisi myös olla mahdollisimman kieli- ja käyttöjärjestelmäriippumaton, jotta olemassa olevia ohjelmistoja, eritoten tuntopalautelaitteiston rajapinnat, voitaisiin helposti integroida agenttiympäristöön. Alustavan vertailun jälkeen päädyttiin rakentamaan oma, kevyt, agenttialusta. Tässä paperissa esittelemme itse rakennetun alustan ja vertaamme sitä muutamiin olemassa oleviin alustoihin.

Avainsanat ja -sanonnat: Agenttiohjelointi, tietoverkot, hajautetut ohjelmisot, tekoäly.

CR-luokat: 61.2, 69.7

1. Johdanto

Tekniikassa on viime vuosien aikana menty aimo harppauksin eteenpäin. Tietoverkkojen yleistyminen ja nopeutuminen tuo uusia mahdollisuuksia toteuttaa ohjelmistoja ja antaa avaimet olemassa olevien ohjelmistokehityksen ongelmien ratkaisemisen. Tietoverkkojen yleistymisen myötä mm. hajautetuilla järjestelmät ovat saaneet huomiota enenevässä määrin. Hajautetuilla ohjelmistoilla saadaan prosessorikuormaa jaettua ympäri tietoverkkoa täten nopeuttaen suoritusajkoja. Lisäksi hajautusta voidaan käyttää hyväksi uuden ohjelmistoparadigman, agenttiohjelmoinnin, kehittämiseen.

Tietoverkkojen yleistyessä myös muu tietotekniikka on jatkanut kehityskulkuaan. Lähiverkoista on tullut hyvin heterogeenisiä käyttöjärjestelmien, käytettävien ohjelmien ja ohjelmointikielten mukaan. Suuri heterogeenisuuden aste vaatii standardointia eri ohjelmistojen välisen

yhteistyön aikaansaamiseksi. Myös laitepuolella on edistytty. Näppäimistö ja hiiri ovat saaneet kilpailijoita mm. laitteista, jotka antavat tuntopalautea itse laitteen kautta. Tuntopalaute avaa ovet tietotekniikan laajemmalle käytölle ja tuo tietotekniikan sellaisten erityisryhmien ulottuville, kuten sokeat, joiden tietokoneen käyttö muuten olisi kovin rajallista.

Suomen Akatemian rahoittamassa PROACT hankkeessa suunnitellaan 6-11 vuotiaille sokeille lapsille opetusohjelmaa maailmankaikkeuteen liittyvistä asioista. Ohjelma rakennetaan erityiseen tuntopalauteympäristöön, missä käytetään PHANTOM tuntopalaute laitetta. Samalla on tarkoitus luoda parempia valmiuksia ja arkkitehtuureja tulevien tuntopalauteprojektien rakentamiseen.

Ohjelma haluttiin hajauttaa lähiverkon koneille prosessorikuorman jakamiseksi ja agenttiparadigman hyödyntämiseksi. Olemassa olevia alustoja vertailtaessa kävi ilmi että yksikään niistä ei täyttänyt vaatimuksiamme; Jotkut olivat liian monimutkaisia, toiset vaativat jatkuvaa ylläpitoa ja jotkut olivat liian joustamattomia. Täten haluttiin rakentaa oma yleiskäyttöinen agenttialusta, joka olisi mahdollisimman kevyt, yksinkertainen, helposti laajennettavissa, kieli- ja käyttöjärjestelmäriippumaton ja, mikäli mahdollista, myös jonkin standardin mukainen.

Tässä raportissa esittelemme aluksi joitakin olemassa olevia hajautettujen järjestelmien ja agenttiohjelmoinnin standardeja. Tämän jälkeen esittelemme kaksi olemassa olevaa agenttialustaa, jotka ovat menestyneet omalla sarallaan. Esittelemme myös oman arkkitehtuurimme ja vertailemme sitä olemassa oleviin.

2. Standardeista

Verkkoliikenteen, ohjelmointikielten ja -paradigmojen ja hajautettujen järjestelmien kehittyessä on laadittu useita standardeja laite- ja ohjelmistoyhteensopivuuden aikaansaamiseksi. Nykyiset agenttialusta jossain määrin nojaavat tai ainakin ottavat kantaa seuraavaksi esiteltäviin standardeihin.

2.1. OMG CORBA

CORBA (Common Object Request Broker Architecture) on OMG:n standardi hajautetuille järjestelmille [CORBA, 2002]. CORBA:a käyttävä ohjelma voi operoida toisen CORBA:a tukevan ohjelman kanssa verkon yli. CORBA pyrkii abstrahoimaan objektien välisen liikennöinnin niin että toteutuksen ohjelmointikieli tai alla oleva käyttöjärjestelmä ei vaikuta prosessiin. Objektien

välinen kommunikaatio tapahtuu IIOP (Internet Inter-ORB Protocol) protokollalla.

Jonkin toiminnan tarjoava palvelinobjekti julkaisee toiminnan määritteen IDL kielellä. Asiakasobjekti käyttää tätä IDL määrettä toiminnan kutsumiseen ja palvelinobjekti tulkitsee asiakkaan toimintopyynnön määritteestä luotavan IDL rungon mukaan. Asiakas tekee toimintopyynnön kuin käsittelisi paikallista oliota, mutta kutsu välittyykin *objektipyynnön välittäjälle* (Object Request Broker), joka ohjaa pyynnön asianmukaiselle objektille tai toiselle välittäjälle IIOP:ta käyttäen.

2.2. KQML ja KIF

DARPA:n³ tukema Knowledge Sharing Effort on poikinut mm. standardit KIF ja KQML [Patil *et al.*, 1992]. Ensinnäkin KIF (Knowledge Information Format) kuvaa yhteisen kielen eri tahojen väliselle tiedon jakamiselle ja kuvaamiselle. KIF perustuu ensimmäisen asteen predikaattilogiikkaan. KQML [Finin *et al.*, 1994] vuorostaan on kolmikerroksinen kommunikointiprotokolla ts. protokollan viesti voidaan jakaa kolmeen loogiseen osaan. Protokolla sisältää kommunikaatio-, viesti- ja sisältötason. Kommunikaatiotasossa määritellään viestin alkeisominaisuuksia, kuten lähettäjän ja vastaanottajan tiedot. Viestitasolla kuvataan mm. mikäkommunikatiotapahtuma on kyseessä. Sisällössä vuorostaan on itse välitettävä informaatio jollakin formaalilla tavalla kuvattuna (kuten KIF). Myöhemmät sovellukset ja standardit, kuten FIPA:n ACL omaavat paljon yhteistä KQML:n kanssa.

2.3. FIPA

FIPA (Foundation of Intelligent Physical Agents), tuottoa tuottamaton organisaatio, perustettiin vuonna 1995. Sen tarkoituksena oli laatia standardit agenttijärjestelmien väliselle kommunikaatiolle ja täten se ei ota kantaa agentin, tai edes agenttialustan, sisäiseen rakenteeseen. FIPAn ensimmäinen virallinen spesifikaatio julkaistiin vuonna 1997. Nykyään itseään FIPA-yhteensopiviksi ilmoittavat alustat viittaavat yhteensopivuudessaan yleensä vuoden 1997 spesifikaatioon, vaikka vuonna 1998 ilmestyi uusi spesifikaatio ja vuonna 2000 ilmestyi FIPA:n abstrakin arkkitehtuurin määrittely. Vuoden 2000 määrittely poikkeaa aiemmista määrittelyistä ja uusilta FIPA -alustoilta vaaditaan yhteensopivuutta vuoden 2000 määrittelylle.

³Defence Advanced Research Projects Agency

2.3.1. FIPA97

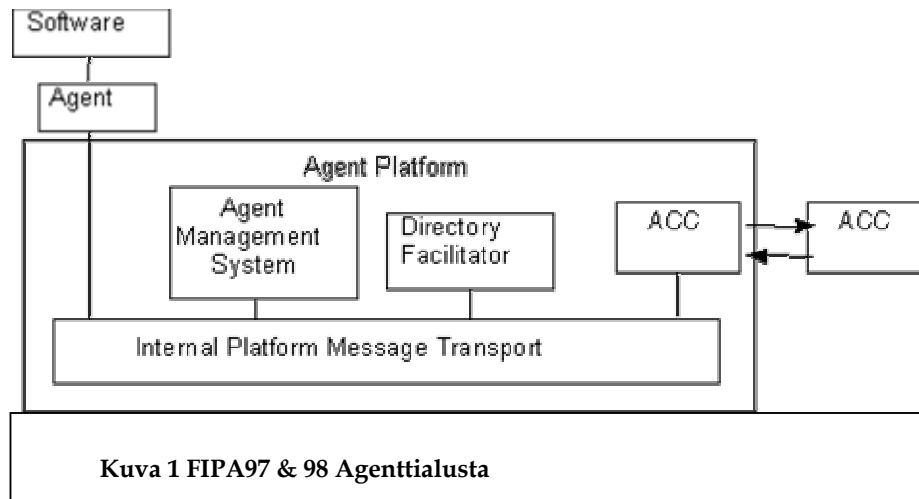
FIPA97 koostuu useista määrittelyistä, joista kolme luokitellaan pakollisiksi yhteensopivuuden saavuttamisen kannalta. Pakollisiin spesifikaatioihin kuuluva osa 1 [FIPA00019, 2001] kuvaa agenttien ylläpitoon kuuluvia asioita, kuten agenttien välistä kommunikaatiota, agenttien luontia ja agenttien sijaintiin liittyviä asioita. Määritelmän mukaan agenttialustan tulisi omata seuraavat ominaisuudet [O'Brien and Nicol, 1998]:

- alustan tulisi tarjota viestipalvelu, joka on luotettava, tarkka ja asianmukainen,
- agentilla tulisi olla nimi, joka mahdollistaa viestin lähettämisen oikealle taholle,
- alustan tulisi havaita ja raportoida virheistä (esimerkiksi viestin lähetyksen epäonnistuminen) lähettävälle taholle, ja
- alusta ei saisi olla riippuvainen yhdestä kommunikointitavasta (esim. HTTP, SMTP jne.) .

FIPA97 yhteensopivassa alustassa tulisi olla seuraavat agenttien ylläpitoon ja niiden väliseen kommunikointiin keskittyneet pakolliset komponentit, jotka itsessään tulisi olla agenteja ja jotka kommunikoivat ACL:n (ks. seuraava kappale) avulla:

- Keltaiset sivut (Directory Facilitator, DF), joka tarjoaa rekisteröimis- ja tiedustelupalvelut agenteille. DF:n avulla agentti voi ilmoittaa (ts. rekisteröidä) tarjoamansa palvelut muille agenteilla. Agentti voi myös etsiä jonkin palvelun tarjoavia agenteja keltaisten sivujen avulla.
- Agenttien ylläpitopalvelut (Agent Management Services, AMS) ylläpitävät tietoa mm. alustalla olevista agenteista, niiden ainutlaatuisista nimistä ja niiden sijainnista ja osoitteista⁴. AMS tarjoaa myös mahdollisuudet agenttien luomiseen, tuhoamiseen, ja rekisteröintiin. AMS pitää myös kirjaa agenttien alustalta poistumisesta ja sille saapumisesta.
- Agenttien kommunikaatiokanava (Agent Communication Channel, ACC) vastaa viestien välittämisestä toisille agenttialustoille (ja myös saman alustan sisällä). ACC:n tulee tukea vähintään IIOP protokollaa ollakseen FIPA yhteensopiva. Tosin alusta saa käyttää mitä tahansa protokollaa alustan sisäiseen liikennöintiin.

⁴Agenttialusta voi olla hajautettuna usealle koneelle lähiverkossa.



Osassa 2 [FIPA00003, 2001] kuvataan agenttien välinen kieli, ACL. Haluttiin kieli, joka on tarpeeksi rikas, jotta sillä voitaisiin kuvata sellaisia asioita kuin agenttien uskomuksia ja haluja. Rikas kieli antaisi agenteille kyvyn välittää toisilleen tietoa maailman tilasta. KQML hylättiin vaihtoehtona siksi että siitä puuttui riittävän formaali semantiikka [O'Brien and Nicol, 1998], se oli liian avoin tulkinnoille ja siinä ei määritelty tapaa virallisesti suostua johonkin kommunikointitapahtumaan (l. agentti ei tiennyt suostuuko toinen agentti esim. pyyntöön). ACL:n määrittely voidaan jakaa viiteen tasoon:

- Protokolla määrää säännöt, jotka vaikuttavat agenttien välisen dialogin rakenteeseen. Protokolla siis määrittää viestien jakson, mikä kuvaa jotain tiettyä dialogia kokonaisuudessaan. Protokolla auttaa agenteja ennustamaan tulevia, dialogiin kuuluvia, viestien olemusta helpottaen näin agenttien välisen kommunikaation luomista.
- Kommunikointitapahtumat (Communicative Act) ovat agenttien välisen dialogin rakennuspalikoita määrittäen aiotun kommunikaation tyypin. FIPA97 määrittelee useita sekä yhdistelmä- että alkeiskommunikaatiotapahtumia. Yhdistelmätapahtuma koostuu useammasta alkeistapahtumasta. Esimerkiksi *pyyntö* on alkeistapahtuma, kun taas *myöntyminen* on yhdistelmätapahtuma. Myöntyminen voi siis koostua useammasta alkeistapahtumasta. Agentti voi esimerkiksi myöntyä ja välittää samalla lisäinformaatiota myöntymisestä, tai agentti voi myöntyä, jos jotkin reunaehdot tavoitetaan. Jokaiselle tapahtumalle on tarkkaan määritelty merkitys.
- Viestintä määrittelee agentin viestin metainformaation. Esimerkiksi tieto lähettäjästä, vastaanottajasta ja viestin sisällöstä kuuluvat metainformaatioon.

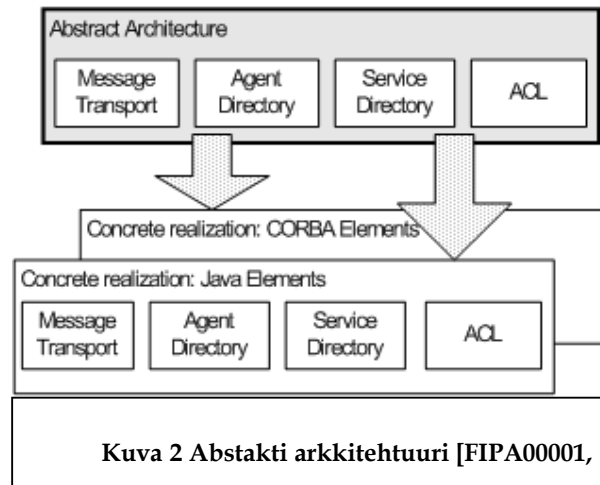
- Sisällön kieli vuorostaan määrittelee viestin sisällön kieliopin ja semantiikan. Esimerkiksi sisällön kieleksi voitaisiin määrittellä Prolog, jos sisältö koostuisi Prolog -kielen imaisista.
- Ontologia vastaa viestin sisällön sanasto, termien ja käsitteiden määrittelystä.

Lopulta osassa 3 [FIPA00017, 2001] kuvataan agenttien ja ei-agenttikomponenttien yhteensovittamista. FIPA97 määrittelee yleisen tavan rakentaa ei-agenttimaisen ohjelmiston ympärille erityisen kääreen, mikä antaa agenteille mahdollisuuden käyttää ohjelmistoakuten agenttia. Spesifikaatio määrittelee erityisen agentin, ARBn (Agent Resource Broker), joka tarjoaa DF:n kaltaiset palvelut ei-agenttimaisille ohjelmistokomponenteille.

2.3.2. FIPA Abstrakti Arkkitehtuuri

Abstrakti arkkitehtuuri [FIPA00001, 2002] sai alkunsa vuonna 1999, kun FIPA päätti omaksua holistisen näkökohdan agenttialustan määrittelyyn. Sen sijaan että aikasempaa määritelmää olisi jatkettu iteratiivisesti kehittämällä tarvittavia osa-alueita, päätettiin edetä tarpeellisten arkkitehtuurin elementtien tunnistamisella ja yleisten, jaettujen, ominaisuuksien kartoituksella. Laadittiin Abstrakti Arkkitehtuuri, mikä toimii suuntaa-antavana spesifikaationa konkreettista agenttialustaa luodessa. Määritelmässä kuvataan useita pakollisia ominaisuuksia, palveluja yms., jotka konkreettisen alustan tulee toteuttaa ollakseen Abstraktin Arkkitehtuurin mukainen. Kuten aina FIPA:n polttopiste on agenttien ja agenttialustojen yhteistoiminnan mahdollistaminen, eikä se täten ota kantaa itse agenttien tai alustan rakenteeseen. Tässä kerrataan lyhyesti Abstraktin Arkkitehtuurin pakolliset elementit.

FIPA97 tai FIPA98 määritelmien mukaiset agenttialustat pystyvät, tietyn rajoituksen, yhteistoimintaan Abstraktin Arkkitehtuurin määrittelyn mukaan rakennetun alustan kanssa käyttäen viestinkuljetuskanavia (gateway). FIPA2000 määrittelyn mukainen alusta pyytää toimimaan rajoituksetta viestinvälityskanavan kautta. Abstrakti Arkkitehtuuri käyttää hyväkseen aiempia määrittelyjä. Esimerkiksi FIPA97:n ACL ja kommunikointiprotokollat ovat mukana määrittelyssä.



2.3.3. Agent-directory

Jokaisen agenttialustan on toteutettava *agenttien hakemistopalvelu*, jonka tehtävänä on ylläpitää tietoa agenteista ja niiden ominaisuuksista. Kirjautuessaan palvelujen juureen agentti välittää ainutlaatuisen nimensä ja agentin *paikallistimen*. Paikallistin (agent-locator) sisältää yhden tai useamman *viestinvälityskuvauksen* (transport-description), joka koostuu viestinvälitysmekanismien tyypistä, osoitteesta ja mahdollisesti mekanismien lisäominaisuuksista. Tämän lisäksi agentin kuvaus voi sisältää muuta informaatiota⁵, esimerkiksi tietoa agentin tarjoamista palveluista tai agentin käytön rajoituksista. Agentti voi muuttaa tietojaan ja myös poistaa tietonsa palvelusta milloin tahansa. Muut agentit voivat suorittaa hakuja agenttien hakemistopalvelussa, ja täten käynnistää dialogeja toisten agenttien kanssa. Agentin ei ole pakko kirjautua hakemistopalveluun, vaan agentti voi esimerkiksi tarjota paikallistintaan yksityisesti suoran toiselle agentille.

2.3.4. Service-directory

Palvelujen hakemistopalvelussa pidetään kirjaa mahdollisista palveluista, joita on tarjolla agenttiympäristössä. Siinä missä agenttien hakupalvelu tarjoaa keinot löytää tiettyjä agenteja, antaa hakemistopalvelu samat mahdollisuuden palvelujen suhteen. Palvelun määrittelyssä on mm. seuraavat tiedot: sen ainutlaatuinen nimi, tyyppi ja sen paikallistin. Kuten agenttienkin kanssa, paikallistin sisältää yhden tai useamman määritteen palvelun allekirjoituksen tyypistä, allekirjoituksesta ja osoitteesta. Allekirjoituksesta ja sen tyypistä

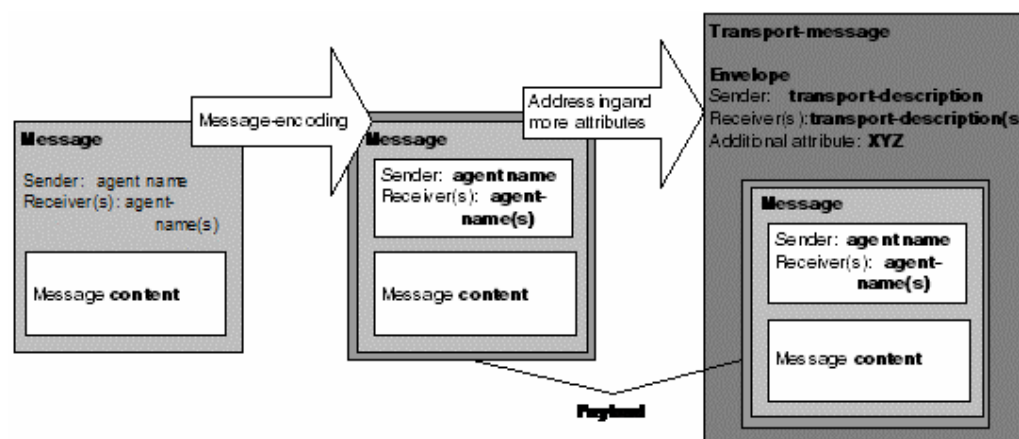
⁵lisäinformaatio kuuluu spesifikaation valinnaisiin, ei pakollisiin osiin.

selviää palvelun ositten tulkinta ja käyttötapa. Palvelun määrittelykin voi sisältää palveluun liittyviä lisäominaisuuksia. Palvelun ei ole pakko ilmoittaa palveluistaan palveluhakemistolle. Palvelu voi muokata tai poistaa palvelumäärittelyään. Lisäksi agentit ja toiset palvelut voivat suorittaa hakuja palveluhakemistoon. Palveluhakemiston palvelut ovat erillisiä sovellustason palveluita. Agenttien ei odoteta kirjaavan palveluitaan palveluhakemistoon.

Käynnistyessään agentille pitää antaa palveluiden *juuri* (*service-root*), joka on joukko palvelun määrittelyjä. Juuren avulla agentti pääsee käsiksi muihin agentteihin ja palveluihin.

2.3.5. Agenttien välinen viestintä

Valmistellessaan viestiä toiselle agentille, se laatii ACL:llä viestin, joka sisältää mm. lähettäjän nimen, vastaanottajan tai vastaanottajien nimet, viestin sisällön ontologiakuvauksen ja itse viestin sisällön. Lähettävällä agentilla on vastaanottavan agentin viestinvälityskuvaukset, joita agentti käyttää viestin muuttamiseen lähetyskelpoiseksi. Tämä tapahtuu pyytämällä *koodauspalvelua* (ks. alla) muodostamaan viestistä *kuorman* (payload) asianmukaisella koodauksella valittuun viestinvälitystapaan nähden. Tämän jälkeen kuorman liitetään *kirjekuori*, joka sisältää mm. tietoa käytetystä koodaustavasta ja lähettäjän ja vastaanottajan viestinvälityskuvaukset. Agentti antaa täten muodostetun *kuljetusviestin* (transport-message) viestinkuljetuspalvelun hoidettavaksi.



Kuva 3 Viestin muodostus [FIPA00001, 2002]

Viestin sisältöön FIPA ei ota kantaa, mutta määrää sisällön olevan jonkin sisältökielen, esimerkiksi FIPA-SL, FIPA-RD, FIPA-KIF tai FIPA-CCL, mukainen. Jokin sisältökieli on pakollinen.

Agenttialustan tulee toteuttaa koodauspalvelu, joka vastaa agenttien välisten viestien muuttaminen ja muutosten purkamisen valitun viestintätavan mukaan. Koodaamisen lisäksi tarjotaan palveluita viestin koodauksen että koodauspalvelun tukemien koodaustapojen selvittämiseen. Koodauspalvelu **voi** myös tarjota viestien salaus- ja varmennuskeinoja.

Viestinkuljetuspalvelu (message-transport-service) vastannee aiempien spesifikaatioiden ACC:tä. Se toimittaa viestejä agenteille. Viestin lähetyksen lisäksi agentti voi sitoa ja vapauttaa jonkin tietyn viestinvälitystavan halutessaan sille tulevien viestien tulevan juuri kyseisellä viestinvälitystavalla. Viestinkuljetuspalvelun ei välttämättä tarvitse olla erillinen palvelu, vaan esimerkiksi alustassa, joka tukee vain yhtä kuljetustapaa voidaan käyttää esimerkiksi käyttöjärjestelmän rakenteita, kunhan palvelun perusominaisuudet toteutetaan.

3. Agenttialustat

Tässä luvussa esitellään kaksi toisistaan huomattavasti eroavaa agenttialustaa. Ensinnä tutustumme JADE:en (Java Agent Development Environment), joka on FIPA standardia noudattava agenttialusta ja tämän jälkeen saa vuoron OAA (Open Agent Architecture), joka noudattaa omaa rakennettaan. Nämä kaksi alustaa valittiin mm. siksi että molemmilla on jo vankka historia takanaan⁶, ne omaavat monipuoliset ominaisuudet ja niillä on laajat käyttäjäkunnat.

3.1. JADE

JADE [Bellifemine *et al.*, 2001] on Javalla toteutettu FIPA2000 yhteensopiva agenttialusta⁷, joka luonnollisesti toteuttaa standardin vaatimat hakemistopalvelun (DF), ylläpitopalvelun (AMS) ja kommunikaatiopalvelun (ACC).

Varsinainen alusta voi olla hajautettuna usealle koneelle lähiverkossa. Jokaisella alustaan kuuluvalla koneella on käynnistettynä ns. agenttisäiliö (AgentContainer), joka tarjoaa agenttien ajoympäristön, säiliön agenttien hallintapalvelut ja viestipalvelut alustan sisällä. Kaikki kommunikaatio käydään viestien välityksellä ja viesti on, alustan sisäisessä liikenteessä, Java olio `ACLMessage`. `ACLMessage` on ACL:n oliomuotoinen esitys. toiselle.

⁶OAA on yli 10 vuotta vanha

⁷Tämä esittely kuitenkin pohjaa aikaisempaan JADE:n versioon, joka oli FIPA97 yhteensopiva. Uudessa versiossa ei ole sellaisia muutoksia, jotka näkyisivät tämän esittelyn tasolla l. perusrakenteet ovat samat.

Erityinen säiliö, Front End, toimii yhteyspintana muihin agenttialustoihin ja vastaa myös hakemistopalveluista ja ylläpitopalveluista. Tavallisia säiliöitä voidaan lisätä ja poistaa alustasta heti kun Front End on käynnistynyt. Tämä tapahtuu JADEn rekisteröinti-protokollalla, jossa käynnistyvä säiliö rekisteröi itsensä Front Endin ylläpitopalvelun kanssa. Tällöin säiliö lisätään säiliötauluun (Agent Container Table). Front End vastaa agenttien ainutlaatuisesta nimeämisestä, samalla se pitää myös tietoja alustan agenteista, tiedot säilytetään globaalissa agenttitaulussa (Agent Global Definition Table).

Säiliö, oli se sitten tavallinen tai Front End, pyrkii käyttämään parasta mahdollista yhteysprotokollaa viestin välitykseen. Viestintä alustojen välillä käydään IIOP:lla (käyttäen siis CORBA:a). Alustan sisällä käytetään, viestin välittämiseen, Javan RMI:tä (Remote Method Invocation), jos vastaanottava agentti on eri säiliössä kuin lähettävä. Javan tapahtumaa vuorostaan käytetään, jos vastaanottaja on samassa säiliössä. Tapahtuma sisältää ACLMessage -olion.

Säiliö [Bellifemine *et al.*, 1999] puskuroi toisten säiliöiden osoitetietoja välttyäkseen jatkuvilta kyselyiltä hakemistopalvelulta (joka saattaa olla toisella koneella). Jos lähetys puskuroituun osoitteeseen epäonnistuu, hakee säiliö vastaanottavan säiliön osoitetiedot Front Endin globaalista agenttitaulusta ja päivittää puskuritietonsa sen mukaan.

JADEssa on valmiina muutamia erityisiä agenteja. Etätarkkailuagentti (Remote Monitoring Agent, RMA) on graafisen käyttöliittymän tarjoava agentti, jonka avulla alustan tilaa ja sisältöä voidaan tarkkailla ja kontrolloida (uusien agenttien luonti, niiden tuhoaminen, säiliöiden sulkeminen jne.). Nuuskija-agentilla (Sniffer) voidaan vuorostaan tarkkailla agenttien välistä liikennöintiä graafisesta käyttöliittymästä.

3.1.1. JADE:n agentit

Agentti on tavallinen Java luokka. Agentti luodaan periyttämällä oma agentti JADEn tarjoamasta abstraktista agentista. Jokainen agentti omaa yhden viestijonon, johon agentille tarkoitetut viestit saapuvat. Agentti tosin voi käydä useampaa dialogia samanaikaisesti.

Jokainen agentti ajetaan omassa säikeessään. Yleisenä sääntönä on että agentilla saa olla vain yksi säie, vaikka mikään ei estä tämän säännön rikkomista. Perimmäinen syy sääntöön on se, että jos säikeiden määrää ei millään tavalla pyritä rajoittamaan, voi säiliöllä, jossa on useita agenteja, olla tehokkuusongelmia.

Säikeiden sijasta agentille määritellään useita *käyttäytymistapoja* (behaviours). Yksi käyttäytymistapa voi olla suorituksessa kerrallaan ja agenttiin on sisäänrakennettu skedulointi, joka vastaa ajettavan käyttäytymistavan valinnasta. Agentti voi lisätä ja poistaa käyttäytymistapoja

ajon aikana esimerkiksi uuden dialogin käymiseksi. Jokaisella agentilla on valmiina käyttäytymistavat agentin rekisteröintiin, rekisteröinnin purkamiseen ja viestien lähettämiseen.

Käyttäytymistapoja on kahdelaisia [Bellifemine *et al.*, 1999]. Yksinkertaiset käyttäytymistavat katsotaan atomisiksi olennoiksi ja täten ne suoritetaan keskeytyksettä alusta loppuun. Yksinkertaiset tavat jakaantuvat vain kerran ajettaviin ja silmukkamaisiin (jotka ovat siis ikuisessa silmukassa). Monimutkainen käyttäytymistapa koostuu yksinkertaisista ja toisista monimutkaisista tavoista. Monimutkaisetkin tavat jakaantuvat kahteen ryhmään: järjestyksellisiin tapoihin, joissa monimutkaisen käyttäytymistavan sisältämät tavat ajetaan järjestyksessä, ja epädeterministisiin tapoihin, joissa suoritukseen valitaan käyttäytymistapa epädeterministisen ajo-algoritmin mukaan.

3.1.2. Esimerkki

Eräässä JADE esimerkeistä (ThanksAgent) agentti luo muutamia muita samanlaisia agenteja samaan säiliöön⁸ ja lähettää niille tervehdysviestin. Tämän jälkeen agentti määrittelee itselleen syklikäyttäytymisen (Cyclic Behaviour), joka on yksinkertainen silmukkamainen käyttäytymismalli. Mallin sisällä agentti kuuntelee saapuvia viestejä. Jos vastaanotettava viesti on tervehdysviesti, lähettää agentti vastauksen. Jos kyseessä on vastaus, lähettää agentti kiitosviestin. Muut agentit tekevät samoin. Koska muita käyttäytymismalleja ei ole, on syklinen malli suorituksessa jatkuvasti.

3.2. OAA

OAA:n (Open Agent Architecture) pääsuunnitteluperiaattena on ollut agenttialustan mukauttaminen heterogeeniseen ympäristöön [Martin *et al.*, 1999]. Tämän lisäksi agenttialustaan haluttiin rakentaa joitakin liitutaulutekniikan (Blackboard), liikkuvien objektien ja kommunikoivien agenttien vahvimpia puolia. Liitutaulutekniikoissa agentit jakavat yhteisen työtilan, liitutaulun, jonka avulla ne voivat tehdä yhteistyötä ja tiedon jakamista. Liikkuvilla objekteilla tarkoitetaan, luonnollisesti, paikasta toiseen liikkuvia oliota tai agenteja ja kommunikoivat agentit vuorostaan käyvät dialogeja yhteisellä kielellä ja ontologialla.

Lisäksi on haluttu rakentaa arkkitehtuuri, joka on joustava sekä ajon aikaisilta ominaisuuksiltaan (agenteja voidaan lisätä ja poistaa ajon aikana), sovelluskehityksen aikana että agenttiympäristön eri tahojen

⁸ uudet agentit eivät luo uusia agenteja, kiitos luokassa määritetylle staattiselle muuttujalle, jonka kaikki saman luokan agentit jakavat

yhteistyöinteraktioiden rakenteessa. Alustan siis haluttiin tarjoavan hyvät ominaisuudet erilaisten yhteistyötapahtumien luontiin mahdollisimman helposti ja tehokkaasti. Arkkitehtuurissa haluttiin myös määrätä agenttien rakennetta sopivasti, muttei liikaa. Jos kommunikointiin olisi käytetty KQML:ää, se ei vaatisi agentin rakenteelta juuri mitään. Toisaalta, toisen merkittävän agenttimetodologian BDI [Rao and Georgeff, 1995], joka perustuu agentin uskomuksille (Beliefs), haluille (Desires) ja tavoitteille (Intentions), katsottiin olevan joihinkin tarkoituksiin pakottamaltaan rakenteeltaan liian vaativa. Päädyttiin ratkaisuun, joka sijaitsee KQML:n ja BDI:n välimaastossa. Tiedostettiin myös valmiiden ohjelmistojen ja komponenttien olemassaolo -- oli ne sitten omia tai kaupallisia -- ja haluttiin mahdollistaa niiden käyttö agenttiympäristössä.

Ihmiskeskeisiä käyttöliittymiä ovat myös keskeinen osa OAA:ta. Agenttialustan tulisi tarjota käsitteellisesti luontevat tavat kanssakäymiseen useiden hajautettujen komponenttien kanssa. Käyttäjän siis pitäisi pystyä suorittamaan tehtäviä tietämättä tarkemmin eri agenteista tai niiden ominaisuuksista. Käyttäjä katsotaan myös yhdeksi, joskin etuoikeutetuksi, agentiksi. Myös agenttien (sekä ihmis- että koneagenttien) välistä yhteistyötä tuetaan.

Alustan tulisi myös täyttää realistiset käyttövaatimukset olemalla helposti käyttöönotettavissa, uudelleenkäytettävä komponenttiensa puolesta, kevyitä mobiilialustoja (PDA:t, puhelimet jne.) tukeva ja mahdollisimman käyttöjärjestelmä ja ohjelmointikieliriippumaton.

3.2.1. ICL

Agenttien välinen liikennöinti käydään OAA:n omalla kielellä ICL:llä (Interagent Communication Language), joka muistuttaa rakenteeltaan Prolog -kielen predikaattikutsua. Yleisin viestintämuoto on *tapahtuma*, jota käytetään aina agenttien aktiviteettien ilmaisusta agenttien väliseen kommunikointiin saakka. Esimerkiksi sähköpostin lähetystä X:ltä Y:lle pyytävä agentti kutsuu OAA:n agenttikirjaston *oaa_solve* funktiota, joka siis mahdollistaa palvelupyynnöt, esimerkiksi seuraavalla tavalla:

```
oaa_post_solve(send_message(email,Y,X),[])
```

Kommunikaatiossa tapahtuma vastaa esimerkiksi JADE:n ACLMessage -oliota. Jokaisella tapahtumalla on tyyppi (*oaa_post_solve*), joukko parametreja (tässä tapauksessa tyhjä lista, []) ja sisältö (*send_message(email,Y,X)*). Koska OAA on käsitteellisesti ja jossain määrin myös olemukseltaan hyvin lähellä Prolog -kieltä, nähdään sisältö, erityisesti palvelupyynnössä, jonkinlaisena tavoitteena tai Prologin predikaattikutsuna, joka tulee saavuttaa.

Tyyppi ja parametrit toteuttavan ICL:n keskusteluprotokollan. Dialogiin liittyvien asioiden määrittely siis tapahtuu oikeanlaisella tapahtumatyyppin ja parametrijoukon valinnalla. Parametreja on kahdenlaisia. Palauteparametreilla palvelua pyytävä taho voi saada tietoa siitä miten annettu tavoite on suoritettu. Pyytjä voi esimerkiksi saada tietoon, mitkä agentit hoitivat tavoitteen suoritusta ja kuinka kauan suoritus kesti. Neuvoa antavat parametrit taas asettavat rajoitteita tai ohjeita jonkin tavoitteen suorittamiselle. Lähettäjä voi esimerkiksi asettaa *time_limit* parametrin määräämään ajan, kuinka kauan tavoitteen suoritusta odotetaan.

Tapahtuman sisältö vuorostaan voi koostua erilaisista tavoitteista, laukaisimista ja välitettävästä tiedosta (oli se missä muodossa tahansa). Sisältö on suunniteltu Prolog -kielen laajennukseksi hyödyntäen näin Prologin erityispiirteitä, kuten yhdistämistä (unification). Sisältö voi olla myös ns. yhdistelmätavoite, joka on erillisten tavoitteiden joukko. Tavoitteiden yhdistelyyn voidaan käyttää seuraavia operaatioita: pilkku on konjunktio, puolipiste disjunktio ja nuoli on ehdollinen suoritus (jos-niin). Lisäksi on rinnakkainen disjunktio, mikä mahdollistaa disjunktoiden suorittamisen eri agenteilla samanaikaisesti. Lopuksi jokaisella alitavoitteella voi olla osoiteattribuutti ja alitavoitteeseen liittyvien parametrien joukko. Osoite tarkoittaa yhden tai useamman agentin, joille alitavoitteen suoritus sallitaan.

3.2.2. Välittäjät

OAA on perusrakenteeltaan keskitetty järjestelmä⁹, jossa kaikki liikenointi tapahtuu yhden tai useamman välittäjän (Facilitator) kautta molempiin suuntiin eli vaaditun tapahtuman tulos palautetaan myös välittäjän kautta. Välittäjän tehtävänä on koordinoita agenttien välistä kommunikointia ja yhteistyössä tapahtuvaa ongelman ratkaisua. Tämän aikaansaamiseksi välittäjä pitää kirjata agenttien tarjoamista palveluista. Lähettävän agentin ei välttämättä tarvitse tietää tietyn palvelun tarjoavasta agentista mitään¹⁰, vaan välittäjän tehtävänä on löytää oikeat tahot agentin lähettämän pyynnön toteuttamiseksi.

Välittäjä pystyy myös käsittelemään yhdistelmätavoitteita. Yhdistelmätavoitteiden käsittelyssä välittäjä sekä delegoi osatavoitteita oikeille tahoille, optimoi osatavoitteita mahdollisimman tehokkaasti ratkaistavaksi ja tulkitsee ratkaistujen osatavoitteiden tulokset muodostaen niistä kokonaisuuden, joka palautetaan palvelua vaatineelle taholle. Jos kyseessä on muu kuin yhdistelmätavoite, voi välittäjä käyttää sisällön informaatiota

⁹Joskin OAA antaa myös mahdollisuuden agenttien väliselle suoralle liikennöinnille

¹⁰Agentti toki voi antaa vastaanottajan tarkan osoitteen ja viestin parametrit tarpeen vaatiessa

hyväkseen pyyntöä delegoidessa, jos sisältö on kuvattu ICL:llä. Välittäjää käytetään usein myös agenttien välisenä globaalina tietovarastona.

3.2.3. OAA:n agentit

OAA arkkitehtuurissa on yleisesti määritelty kolme erilaista agenttiluokkaa. Luokittelu tosin ei ole kuin neuvoa antava eikä sitä siis ole arkkitehtuuriin mitenkään kiinnitetty. Sovellusagentit tarjoavat palveluiden joukkoja kuten sähköposti-, puheentunnistus- ja matkanvarauspalveluja. Meta-agentit avustavat välittäjää toisten agenttien toimintojen koordinoinnissa lisäämällä sovellus- ja aihekohtaisia tietoja ja päättelytapoja. Viimeisenä on käyttöliittymäagentit, jotka toimivat mm. eri modalitteettien, eli syöte- ja toistotapojen, tulkinnassa ja niiden yhdistämisessä.

Agentit siis eivät pelkästään käytä välittäjän palveluita hyväkseen vaan luonnollisesti myös tarjoavat palveluita muiden agenttien käytettäväksi. Käynnistyessään agentti kirjaa tarjoamansa palvelunsa välittäjälle. Agentti voi lisätä, poistaa tai muokata palveluitaan suorituksen aikana. Esimerkiksi agentti voi määrittellä palvelun yksityiseksi, jos se ei halua palvelua käytettävän muualta kuin agentista itsestään käsin.

3.2.4. Palvelut

Agentin tarjoama palvelu (solvable) voi olla joko informaatio tai suorituspalvelu. Suorituspalvelussa agentti suorittaa jonkinlaisen selkeän toiminnon, kun taas informaatiopalvelussa tarjotaan käytettäväksi jotakin informaatiokokoelmaa. Molemmat palvelutyypit ovat toiminnallisesti ja määrittelyiltään erilaisia, mutta kutsuvalle agentille niillä ei ole eroa käyttötavoissa.

Palvelun määrittely sisältää tavoitteen ja käyttöoikeuksien ja parametrien joukot. Tavoite on ICL:n mukainen ja tarjoaa loogisen esityksen tarjottavasta palvelusta. Parametreissa välitetään esimerkiksi tieto siitä onko kyseessä informaatio vai suorituspalvelu. Lisäksi voidaan määrittellä palvelu henkilökohtaiseksi antamalla asettamalla yhdeksi *parametriksi* "private". Käyttöoikeuksilla voidaan luonnollisesti rajata palvelun käyttöä vain tietyille agenteille.

3.2.5. Laukaisimet

Agentit (siis myös käyttäjät) voivat asettaa laukaisimia jonkin tietyn tapahtuman seuraamiseen. Laukaisimen voi asettaa itseensä, toiseen agenttiin tai välittäjään. Asettaessaan laukaisinta agentti määrittelee jonkin reunaehdon ja tapahtuman minkä tulee tapahtua reunaehdon täytyttyä. Esimerkiksi käyttäjä

voisi asettaa sähköpostiinsa laukaisimen, joka ilmoittaisi puhelimitse tietyn sähköpostin saapumisesta.

Laukaisimia on neljänlaisia. *Kommunikaatiolaukaisin* sallii minkä tahansa ulos- tai sisäänpäin menevän tapahtuman tarkkailua. *Informaatiolaukaisin* sallii jonkun tietovaraston tarkkailua. Informaatiolaukaisimella voidaan tarkkailla mm. jonkin tiedon lisäämistä, poistamista tai muuttamista. *Tehtävälaukaisimen* avulla agentti voi tarkkailla jotakin sisääntulevaa tapahtumaa määrittelemällä ICL:n mukaisen tavoitteen. Esimerkiksi yllä mainittu sähköpostiesimerkki on tehtävälaukaisin. *Ajastinlaukaisimet* tarjoavat keinot tarkkailla ajallisia tilanteita (klo. 15.00, kolmen minuutin välein jne.).

Laukaisimet toteutetaan informaatiopalveluina. Käsitellessään laukaisinta tahon, johon laukaisin on asetettu ei tarvitse ymmärtää kuin laukaisimen ehto-osa. Lauetessaan kohdeagentti lähettää laukaisimen suoritusosan takaisin välittäjälle, joka jälleen delegoi tavoitteen asianmukaisille tahoille.

3.2.6. Esimerkki

OAA toimii hyvin samalla tavalla kuin Prolog etenee ohjelman suorituksessa. Periaatteessa kyse on yhdistämisestä (unification). Esimerkkiympäristössämme on kolme agenttia ja välittäjä. Ensinnäkin on käyttöliittymäagentti, joka toimii käyttäjän ja muun arkkitehtuurin yhdistäjänä. Tietokanta-agentti tarjoaa tuiki tarpeelliset faktat. Puheentunnistinagentti kuuntelee puhuttuja käskyjä ja luonnollisen kielen tulkkiagentti tulkitsee tällaisten kommentojen sisällön.

Sanoessaan "Lähetä sähköpostia Joukolle ensi viikon kokouksesta" tapahtuu seuraavaa. Käyttöliittymäagentti ohjaa puheen välittäjälle pyytäen sen tulkintaa. Välittäjä ohjaa sen puheentunnistimelle, josta palautetaan välittäjälle tunnistettu viesti ICL:nä. Välittäjä edelleen siirtää viestin tulkille, joka muodostaa lauseesta seuraavan tapahtuman:

```
send_message(email,Jouko, [subject("ensi viikon kokous")])
```

Tässä vaiheessa välittäjä voi etsiä **Joukon** koko nimen, tai vakkapa sähköpostiosoitteen järjestelmästä riippuen, tietokanta-agentilta. Näin välittäjä on valmis suorittamaan seuraavan tapahtuman:

```
send_message(email,"Jouko Jokinen", [subject("ensi viikon kokous")])
```

Lopulta tapahtuma ajautuu sähköpostipalvelulle, joka kaivaa lähettäjän nimen tietokanta-agentilta ja lähettää viestin ihan tavallisen UNIX *sendmail* ohjelman kautta (Tosin sähköposti olisi tässä tapauksessa aika lyhyt).

4. Oma Agenttiarkkitehtuuri

Itse kehitetyn agenttiarkkitehtuurin perimmäisenä suunnittelukriteerinä on alustan helppo muokattavuus ja käytettävyys. Olisi tarkoitus että

yksinkertaisen hajautetun agenttijärjestelmän saisi rakennettua muutamassa tunnissa perehtymättä alustan rakenteeseen tarkemmin. Alustasta ohjelmoijan ei välttämättä tarvitse tietää kuin menetöt viestien lähettämiseen ja vastaanottamiseen. Alustaa suunniteltaessa siitä pyritty muodostaa sellainen, että myöhemmin olisi mahdollista saavuttaa yhteensopivuus FIPA:n Abstraktin Arkkitehtuurin kanssa.

Kuten JADE:ssa, voidaan agenttialusta hajauttaa usealle lähiverkon koneelle erillisten agenttisäiliöiden (AgentContainer) muodossa. Erillinen viestikanava (MessageChannel) vastaa mm. hakemistopalveluista. Sekä viestikanavalla että säiliöillä on yksinkertainen konfigurointitiedosto, jonka muokkaaminen riittää niiden toimintakuntoon saattamiseksi.

Viestintä tapahtuu pääosin keskitetysti viestikanavan kautta (ellei vastaanottaja ole samassa säiliössä), siten että agentti antaa viestin säiliölle, joka vuorostaan lähettää sen viestikanavalle. Viestikanava lähettää viestin edelleen säiliölle, jossa vastaanottaja sijaitsee ja lopulta vastaanottava säiliö toimittaa viestin vastaanottavan agentin postilaatikkoon. Eri säiliöissä olevien agenttien välillä kulkevat viestit ovat tekstimuotoisia XML viestejä, jotka koittavat parhaansa mukaan noudattaa FIPA:n ACL määritelmää. Saman säiliön sisällä viesti siirtyy Message luokkan oliona, joka on ACL viestin olioesitys.

Mentäessä monimutkaisempiin järjestelmiin tulee pakolliseksi antaa agentille mahdollisuudet omien tietoliikenneyhteyksien muodostamiseen sillä viestikanavasta muodostuu nopeasti järjestelmän pullonkaula. Arkkitehtuurin tulee siis tarjota helpot rajapinnat yleisimpien tietoliikenneprotokollien käyttöön (TCP/IP, UDP jne.), jolloin agentti voisi aloittaa dialogin viestikanavan avulla ja sopimukseen päästyään agentit voisivat keskustella suoraan keskenään.

Koska tarkoituksena on myös olemassa olevien ohjelmistojen ja haptisen laitteen, PHANToMin, integroiminen agenttiympäristöön, haluttiin järjestelmästä mahdollisimman käyttöjärjestelmä- ja ohjelmointikieliriippumaton. Tämä saavutettiin hylkäämällä agenttialustoille tyyppillinen mahdollisuus siirtää agentti alustasta tai säiliöstä toiseen verkon yli. Lisäksi viestit eri säiliöiden välillä kulkee FIPA ACL:n XML muotoisena esityksenä.

Periaatteessa on mahdollista luoda agenttisäiliö millä tahansa ohjelmointikielellä, missä on vähintäänkin rajapinnat verkkoliikenteeseen TCP/IP -protokollaa käyttäen. Kun säiliö on valmis, voidaan kyseisellä kielellä keskittyä agenttien luontiin tarvitsematta ottaa kantaa enää tietoliikenneprotokolliin yms. Agenttien on myös mahdollista avata omat

kommunikointikanavat, mutta silloin se jää agentin suunnittelijan harteille. Tällä hetkellä säiliö on toteutettu sekä Pythonilla että C++:lla.

Ajonaikaista kokonaan uusien agenttityyppien (siis ei agenttien, vaan agenttityyppien) lisäämistä järjestelmään ei tueta, ellei käytetty ohjelmointikieli itsessään sitä tue. Tällöin kuitenkin on kyse erityispiirteestä, jota on käytettävä varovasti. Agentteja vuorostaan voidaan luoda joko säiliön käynnistyksen yhteydessä (määriteltynä säiliön konfigurointitiedostossa) tai ajon aikana esim. erillisellä viestillä agenttisäiliölle.

JADE:n tapaan ajatuksena on ollut että ohjelmoijan tarvitsisi vain erikoistaa ja laajentaa tietyn määrätyn toiminnallisuuden toteuttavaa primitiiviagenttia tarpeidensa mukaan. Olio-ohjelmointikielissä tämä tietenkin tarkoittaa perimistä, muissa kielissä luultavasti uudelleenmäärittelemistä, korvaamista tms. Yksinkertaisimmillaan ohjelmoija vain perii abstraktin agenttiluokan ja uudelleenmäärittelee kriittiset (abstraktit) metodit, kuten run, terminate jne. Tällöin agentin toteuttaminen on yksinkertaista ja helppoa.

Järjestelmä takaa postipalvelut (sekä viestien lähettäminen että vastaanottaminen), mutta dialogien ja kommunikaatioprosessien hallinta jää agentin suunnittelijan huoleksi. Samalla tavoin viestien kehys on ennalta määrätty, mutta varsinaiseen sisältöön ei oteta kantaa (muutamia yleisluontoisia viestejä lukuunottamatta). Jälleen on agentin suunnittelijan itse toteutettava ne toiminnot mitä agentilta haluaa¹¹.

Koska alusta on ajateltu käytettäväksi suljetussa tutkimusympäristössä, ei siinä ole huomioitu tietoturvallisia näkökohtia. Tällä hetkellä agenttien oikeellisuutta tai oikeuksia ei tarkisteta missään vaiheessa.

4.1. Message Channel

Viestikanava (MessageChannel) vastaa hakemistopalveluista. Käynnistyessään sekä säiliöt että agentit kirjautuvat viestikanavalle. Viestikanava vastaa uusien säiliöiden nimeämisestä ja säiliö saakin ainutlaatuisen nimensä kirjautumisprosessin päätteeksi. Agentti vuorostaan lähettää käynnistyessään tarjoamiensa palveluiden tiedot. Jatkossa sekä agentit että säiliöt myös poiskirjautuvat lopettaessaan ja agentit myös ilmoittavat mahdollisista tilanmuutoksista (pysäytetty, aktiivinen jne.) .

Viestikanava pitää kirjaa olemassa olevista säiliöistä ja niiden tukemista tietoliikenneprotokollista¹², agenteista ja missä säiliössä ne sijaitsevat, ja tarjottavista palveluista. Yleisin viestikanavan tarjoama palvelu on tietyn palvelun tarjoavien agenttien listan tarjoaminen agenteille. Eli agentti voi kysyä

¹¹Olisihan kovin rajoittavaa koittaa määrätä agentin ominaisuuksia arkkitehtuurissa

¹²Tällä hetkellä vain TCP/IP

viestikanavalta, mitkä agentit tarjoavat esimerkiksi lokipalvelua. Vastauksena agentti saa listan (tekstimuotoinen XML viesti) palvelun tarjoavista agenttien tunnuksista, jota se voi käyttää jatkossa esimerkiksi kommunikaatiotapahtumia käynnistettäessä. Viestinanavalta voidaan pyytää myös kaikkien palveluiden listaa, jossa XML-muodossa on listattu palvelun tarjoavat agentit palveluittain.

Yleensä vastaanottavista agenteista tiedetään tunnistin, joka koostuu agentin nimestä, "@" -merkistä ja osoitteesta. Osoite on säiliön nimi¹³, joka on yleensä muotoa "containerX", missä X on juokseva numero; täten siis "[xzy@container1](#)" on agentin xyz tunniste. Lähettäessään viestiä agentti yleensä liittää vastaanottajaksi agentin tunnoksen. Viestikanavan huoleksi jää viestin välittäminen oikealle taholle (ks. kuva 3). Sen lisäksi että agentti voi lähettää viestejä toisille agenteille tuntiessaan niiden tunnisteiden, voi agentti myös lähettää viestin tietyn palvelun tarjoaville agenteille (kysymättä agenttien listaa etukäteen) tai viesti voidaan lähettää myös kaikille agenteille (Broadcast).

Viestikanava mahdollistaa myös viestin lähettämisen agentille josta tiedetään vain nimi (ei siis tunnistetta). Vastaava lähetystapa on tuiki harvinainen -- alustahan ei tue agenttien liikkuvuutta -- joten sitä tullaan vain käyttämään hyvin erityisten agenttien kanssa. Esimerkiksi voidaan ajatella että järjestelmässä on joitankin kiinteitä agenteja, joilla on ennaltamäärätty ainutlaatuinen nimi. Tällöin voidaan lähettää viestejä pelkällä nimellä ja ei haittaa vaikka kyseinen agentti suljettaisiin toisella alustalla ja käynnistettäisiin toisella.

4.2. Agent Container

Agenttisäiliö (AgentContainer) toimii agenttien ajo-alustana ja tarjoaa yksinkertaiset postipalvelut agenteille. Postipalveluihin kuuluu mm. agentin omakohtainen postilaatikko ja yksinkertaiset viestilähetysoperaatiot. Säiliö myös vastaa agenttien luomisesta, tuhoamisesta ja nimeämisestä.

Käynnistyessään säiliö kirjautuu viestikanavalle ja saa tältä ainutlaatuisen nimen. Kirjautumisen yhteydessä säiliö myös välittää tukemansa viestintäprotokollat viestikanavalle. Tämän jälkeen säiliö käynnistää kaikki ne agentit mitkä oli määritetty käynnistettäväksi säiliön konfiguraatitiedostossa.

Säiliön yksi tarkoitus on abstrahoida matalan tason tietoliikenne ja säietoiminnot siten että agentin ei välttämättä tarvitse ottaa kantaa niihin. Siksi säiliölle voidaankin määritellä useita eri tietoliikenneprotokollia, mitä sen pitäisi kyetä tukemaan (nykyään vain TCP/IP tuettu). Teoriassa on siis mahdollista pistää säiliö kuuntelemaan TCP-liikennettä tietyssä portissa ja

¹³Tämä saattaa tosin muuttua siinä vaiheessa kun halutaan liittää alusta muiden alustojen seuraan

UDP-liikennettä toisessa (ja tulevaisuudessa myös joitain ylemmän tason protokollia, kuten HTTP:tä, SMTP:tä jne.). Tällöin viestin lähettäjä (yleensä viestikanava) voisi koittaa toista tietoliikenneprotokollaa toisen pettäessä. Säiliö itsessään toimii juuri näin lähettäessään viestiä viestikanavalle¹⁴.

Agenttia ajetaan erillisessä säikeessä ja se, ohjelmoijan niin halutessa, saa käynnistää uusia säikeitä. Agentin kuitenkin tulee itse tällöin vastata lisäsäikeiden tuhoamisesta agentin elinkaaren päättyessä. Olisi kuitenkin suotavaa, että uusia säikeitä ei luotaisi tuhottomasti, sillä -- kuten JADE:n tapauksessa -- se saattaa johtaa vakaviin tehokkuusongelmiin.

Säiliön tarjoma postipalvelu sisältää metodin *sendMessage* viestien lähettämiseksi, joka saa parametrinaan Message tyyppisen olion, josta tarvittaessa (jos vastaanottaja on toisessa säiliössä) luodaan tekstimuotoinen ACL-XML viesti. Lisäksi jokainen agentti saa oman postilaatikon. Säiliö tarjoaa yksinkertaisen tavan käsitellä postilaatikkoa metodilla *getMessage*, joka poistaa viestijonon ensimmäisen viestin ja antaa sen agentille. Jos tämä toiminnallisuus ei riitä, agentti voi käsitellä postilaatikkoa myös suoraan ja saa tällöin monipuoliset manipulointiominaisuudet käyttöönsä.

Säiliö vastaa agentin tunnusteen luonnista, eli agentti määrittelee Messageluokkaan lähettäjäksi nimensä, mutta säiliö lisää siihen puuttuvan "@"-merkin ja säiliön nimen. Tällöin agentin ei tarvitse huolehtia moisista teknisistä yksityiskohdista. Agentti tosin saa, halutessaan, selville isäntäsäiliön nimen erityisellä metodilla *getContainerName*.

4.3. Agentit

Agentti on tavallinen olio (jos kyseessä on C++ tms. Oliokieli), jota suoritetaan omassa säikeessään. Agentti saa käynnistyessään ainutlaatuisen nimen agenttialustalta, jota se käyttää kirjautuessaan viestikanavalle¹⁵, ja viitteen säiliöön, jotta se voi käyttää säiliön viestipalveluja. Käynnistyksen yhteydessä agentille voidaan myös antaa parametreja. Parametrit välitetään, kielikohtaisesti, jonkinlaisessa nimi-tyyppi-arvo rakenteessa. Esimerkiksi C++:ssa parametri on erillinen luokka joka sisältää merkkijonot nimelle ja arvolle ja kokonaisluvun tyyppin kuvaamiseksi.

Mm. Pythonin ja C++:n tapauksessa agentin tulee uudelleenmäärittellä metodi *run*, joka vastaa säikeen suoritusilmukkaa. Poistuessaan siis *run* metodista, ellei agentti ole käynnistänyt muita säikeitä, agentti tuhoutuu. Yksinkertaisimmillan ohjelmoija siis kirjoittaa *run* metodin, joka tekee

¹⁴vain TCP/IP on tuettu tällä hetkellä.

¹⁵kirjautuminen ei ole pakollinen toimeenpide, mutta silloin muut agentit eivät saa tietoon agentin tarjoamia palveluita

vaadittavat toiminnot ja lakkaa tämän jälkeen olemasta. Jos agentti tuhoaa itsensä, se voi kutsua säiliön *agentTerminated* metodia informoidakseen säiliötä poistumisestaan.

Järjestelmän tarjoama agenttien yläluokka tarjoaa muutamia valmiita toimintoja. Tuhoutuessaan agentti automaattisesti poiskirjautuu viestikanaavasta ja ilmoittaa tuhoutumisesta säiliölleen¹⁶. Tämän lisäksi agentille voidaan määritellä palvelut metodilla *addService* ja palvelut voidaan rekisteröidä viestikanaavassa metodilla *registerServices*.

Kun halutaan lähettää viesti, luodaan Message olio ja määritellään tarpeelliset kentät. Tämän jälkeen viesti-olio annetaan säiliön huolehdittavaksi kutsumalla metodia *sendMessage*. Säiliö muuttaa viestin ACL XML muotoon tarvittaessa.

4.4. Kontrolliluokka ja PHANToM

Kontrolliluokka on erityinen -- varsinkin PROACT hankkeessa käytettävä -- komponentti, joka sisältää erikoistetun agenttisäiliön ja tarjoaa dynaamisen ja nopean tavan sovellusten luomiseen. Kontrolliluokka on rakennettu käytettäväksi kolmiuloitteisessa tuntopalauteympäristössä (PHANToM & ReachinAPI). Tässä ympäristössä sovellusohjelma voidaan nähdä ohjelman tilojen verkko- tai puumaisena rakenteena, ohjelman kulun ollessa siirtymä verkon solmusta toiseen. Lisäksi jokainen solmu itsessään on oma 3D maailmansa, joka ladataan erillisestä VRML määrittelytiedostosta. Kontrolliluokka siis lataa dynaamisesti ohjelman määrittelyn tiedostosta, jossa määritellään ohjelman rakenne ja ne tiedostot mistä 3D maailmat löytyvät, ja rakentaa siitä ns. navigaatiopuun. Navigaatiopuun avulla voidaan rakentaa minkälaisia ohjelmia tahansa lähtien lineaarisista ohjelmista aina verkkomaisiin rakenteisiin.

Kontrolliluokalla on vapaa pääsy niin 3D maailman rajapintoihin kuin tuntopalautelaitteen ominaisuuksiin ja kontrollointiin. Jatkossa kontrolliin rakennetaan monipuoliset mahdollisuudet erilaisten tuntopalaute toimintojen aikaansaamiseksi sekä hyvät rajapinnat sovelluksen kontrollointiin ja sovelluksen ja 3D ympäristön tietojen saamiseen ja asettamiseen.

Erikoistettu säiliö on siis suunniteltu toimivaksi kontrolliluokan ja haptisen laitteen, PHANToM:in kanssa. Phantomsäiliö on tavallisen säiliön kaltainen, mutta siinä säiliön agenteille annetaan viite myös kontrolliluokkaan, jonka kautta säiliön agentit voivat suoraan vaikuttaa mm. tuntopalauteeseen, tilojen siirtymiin ym. sovellustason tapahtumaan. Agentit voivat myös välittää tietoa sovelluksesta muille agenteille. Erikoistetussa säiliössä tulee olemaan

¹⁶ei vielä toteutettu

pääsääntöisesti edustaja-agentteja, jotka mahdollistavat yhteyden sovelluksen ja muiden agenttien välillä. Muut agentit siis kommunikoivat edustajien kanssa, jota vuorostaan käsittelevät sovellusta annetun viitteen kautta.

4.5. Esimerkki joustavuudesta

Yksinkertaisen pikaviestimen laatiminen agenttialustalla on varsin helppoa. Jokaisen käyttäjän koneella on suorituksessa oma agenttisäiliö, joka saa alustustiedostossa, *containerspec.xml*, viestikananvan IP-osoitteen ja portin lisäksi yhdeksi käynnistettäväksi agentiksi pikaviestimen mahdollistavan agentin määrittelytiedoston nimen.

```
<container>
  <server ip="xxx.xxx.xxx.xxx" port="xxxx"/>
  <port port="xxxx"/>
  <agents>
    <agent file="im-agent.xml"/>
  </agents>
</container>
```

Säiliö tulkitsee *im-agent.xml* tiedoston, luo halutun agentin ja lähettää alustuksessa mahdolliset agentin parametrit. *Im-agent.xml* näyttää tältä:

```
<agent class="AllAgents.ImAgent">
  <parameter type="string" name="user" value="Rami"/>
</agent>
```

ImAgent on uusi luokka, joka tyyppilliseen tapaan johdetaan abstraktista *Agent* luokasta¹⁷. *ImAgent* saa siis alustuksessaan erillisessä määrittelytiedostossa annetut parametrit. Tässä tapauksessa parametrina tulee käyttäjän nimi.

Käynnistyessään *ImAgent* liittää käyttäjän nimen **palveluksi** ja rekisteröi itsensä viestikanavaan. Seuraavaksi agentti pyytää kaikkien palveluiden listaa viestikanavalta. Koska agenttien tarjoamat palvelut ovat käyttäjien nimiä, voidaan tätä tietoa käyttää suoraan hyväksi viestin välityksessä samalla kun listasta saadaan myös vastaanottavan tunnisteet.

ImAgentille luodaan jonkinlainen käyttöliittymä, minkä avulla käyttäjä voi sekä lukea että lähettää viestejä haluamilleen tahoille. Lisäksi agentti hakee

¹⁷Tässä nojataan c++:n rakenteisiin

tietyn väliajoin uudelleen kaikkien palveluiden listaa, jolloin tavoitettavissa olevien henkilöiden lista on ajan tasalla.

Viesti toiselle käyttäjälle voi kaikessa yksinkertaisuudessaan sisältää lähettäjän ja vastaanottajan tietojen lisäksi vain varsinaisen viestin sisältöosiossa. Viesti voidaan lähettää useammalle henkilölle samanaikaisesti.

Todellisessa sovelluksessa pitäisi tietoturva toteuttaa jollakin tavalla. Voitaisiin esimerkiksi tehdä niin että erillinen kirjautumisagentti otaisi agenttien kirjautumispyynnöt vastaan. Kirjautumisagentilla olisi tiedot sallituista käyttäjistä ja heidän salasanoistaan. Käyttäjän nimen lisäksi agentin kirjautumisviesti sisältäisi käyttäjän salasanan, jonka ImAgent voi kysyä käyttäjältä itseltään. Tämän jälkeen agentit voisivat tiedustella käyttäjien listaa kirjautumisagentilta tai sitten (tietoturvallisuuden kannalta huonompina vaihtoehtona) kirjautumisagentti voisi ohjata sallitun agentin käyttäjän nimen palveluksi viestikanavaan ja agentit toimisivat kuten edellä.

Koska alustamme ei rajoita viestin sisältöä, voitaisiin yhtä hyvin välittää vaikkapa ääntä viestin sisällössä. Tällöin tosin olisi järkevämpää lähettää viesti suoraan vastaanottajalle kuin että sen lähettäisi viestikanavan kautta. Lisäksi ääntä kannattaisi lähettää vuona (stream) eikä yksittäisenä viestinä. Tämä toki on mahdollista agenttialustallamme, kunhan vuon aikaansaava ja sen käsittelyä koskeva tekniikka toteutetaan agentissa.

5. Vertailua muihin alustoihin

Tässä kappaleessa vertaillaan oman arkkitehtuurin eroja aiemmin esiteltyihin arkkitehtuureihin nähden. Samalla pohditaan miksi kyseiset alustat eivät kelvanneet harkittaessa PROACT projektin agenttialustan toteutusta. Vertailussa kannattaa ottaa huomioon sekä JADE:n että OAA:n elinkaari. Ne ovat olleet aktiivisessa kehityksessä vuosia ja täten ne ovat huomattavasti laajempia ja vakaampia kokonaisuuksia.

5.1. JADE vs. Oma Alusta

JADE rakenteeltaan ja ominaisuuksiltaan olisi voinut hyvinkin olla erinomainen vaihtoehto agenttialustaksi. Rakenteeltaan JADE on helposti ymmärrettävä ja yhdenmukainen. Ei siis ihme että oma alusta on paljon velkaa JADE:lle. Käytännössähän rakenne on hyvinkin samankaltainen lukuunottamatta JADE:n käyttäytymismalleja ja hieman erilaista DF:n (viestikanava), ACC:n (viestikanava) ja alustan sisäisen liikenteen toteutusta (XML ACL).

Ylläpidoltaan tai asennukseltaan JADE ei vaadi liikoja. Kaikki tarvittava tulee yhdessä paketissa. Käyttäjä ei tarvitse mitään erityisiä tekniikoita, kuten

JavaBean alustoja tai LDAP palvelimia. Lisäksi JADE tarjoaa FIPA2000 yhteensopivuuden ja luotettavat yhteydet agenttialustojen välillä. Omalla alustalla on vielä paljon kehitettävää vastaavanlaisten toiminnallisuuksien saavuttamiseksi.

Pääasiallisesti JADE ei sopinut alustaksemme siksi, että haluttiin agenttiympäristö, jossa voisi ottaa käyttöön valmiina olevia ohjelmistoja. Erityisesti haluttiin integroida tuntopalautelaitteen rajapinnat, jotka on toteutettu C++:lla. Vain-Java toteutukseen suostuminen merkitsisi myös monien muiden ohjelmistojen poisjäämistä, kuten Prologin, Lispin ja CLIPS:n. Olisihan tietysti mahdollista JNI:n avulla kirjoittaa kääreet ohjelmistokomponentin päälle, mutta tämä tarvitsisi tehdä jokaiselle lisättäväksi haluttavalle ohjelmistolle (ja joissain tapauksissa projektikohtaisestikin). Ohjelmiston kehittyessä käärettä olisi korjattava ja rakennettava yhä uudelleen. Tämä ei käy, jos halutaan omaksua ratkaisu, jota voidaan käyttää yleisluonteisesti eri projekteissa.

Lisäksi ei ollut varmuutta siitä miten Java ja C++ suorittavat yhteispeliä JNI:n avulla vaativassa tilanteessa, jossa ajossa on 3D tuntopalauteympäristön sekä reaaliaika- että ns. scene-graph prosessi.

5.2. OAA vs. Oma Alusta

OAA vuorostaan vastaa suunnitteluperiaatteiltaan omaa alustaamme. Myös OAA on suunniteltu heterogeeneisiin ympäristöihin, joissa on paljon valmiita ohjelmistoja. Kuten aiemmin todettiin, yksi OAA:n päämääristä onkin sulauttaa valmiit ohjelmistot agenttialustaan.

OAA on kuitenkin jo vanhahko ohjelmisto siten että sen olemus ei muistuta nykypäivän olio-ohjelmointi paradigmaa. Sen sijaan OAA:n juuret ovat syvällä Prolog -kielen ominaisuuksissa. Tällä on sekä hyvät että huonot seuraukset. OAA:n vahvuudet ovat Prologin erityispiirteiden hyväksikäyttö. Eritoten yhdistämisen vahvuudet tulee hyvin esiin, esimerkiksi vaatimalla palvelua käskyllä `oaa_solve(send_message(Kind,From,To),[])`. Prologin tapaan isot alkukirjaimet ovat muuttujia ja jos esimerkiksi Kind muuttujaa ei ole sidottu (ts. sille ei ole annettu arvoa), voi järjestelmä tarjota kaikkia mahdollisia viestinvälitystapoja viestin välitykseen (tai sitten valitaan ensimmäinen vaihtoehto). Toinen vahvuus on prosessoinin jakaminen rinnakkaiseen suoritukseen, joka tehostanee suoritusta entisestään.

OAA:n heikkoudet tulikin jo samalla mainittua. Jos ei ole tutustunut Prologiin ja sen suorituskoneistoihin yllä mainittu yhdistämisesimerkki ei kerro lukijalle juuri mitään. Alustan käyttäjän siis tulee osata Prologia ja sen mukauttamista OAA:n hieman vaikeaselkoiseen ja hankalaan syntaksiin. OAA siis vaatii todellista perehtymistä alustan rakenteeseen, ennen kuin alustaa voi

alkaa käyttämään. Erikoinen syntaksi tuo lisätaakkaa agenttien rakentamiselle. Lisäksi OAA:n saattaminen erittäin tehokkaaksi vaatii juuri järjestelmän läpikotaista tuntemista.

OAA:n agenttimalli on myös hyvin suorituskeskeinen. Siinä missä JADE:lla tai omalla alustalla voidaan luoda aidosti dialogia käyviä agenteja, OAA tuntuu kallistuvan erityisten toimintojen, tavoitteiden, ratkaisemiseen. Eli suuri osa toiminnasta tuntuisi olevan tietyn päämäärän aikaansaaminen ei agenttien välinen dialogi ja tiedonvaihtotapahtuma, joka saattaisi olla parempi malli tässä tilanteessa. Omalla alustallamme voimme toteuttaa kaikki OAA:n pääominaisuudet.

6. Lopuksi

Omassa agenttialustassa on vielä paljon kehitettävää. Tässä vaiheessa on aika kartoittaa kaikkien mahdollisten kommunikaatioprosessien rakenne ja luonne yhteneväisten rajapintojen aikaansaamiseksi. Lisäksi alustaa tulee iteratiivisesti muokata yleisemmäksi ja kohti FIPA:n Abstraktia Arkkitehtuuria.

Tekniseltä kannalta katsottuna seuraava vaihe Python säiliössä on sen kehittäminen ajan tasalle C++ säiliöön nähden. C++ säiliötä tulisi muuttaa pois template -luokista ja toteuttaa useampi luokka rajapinnan tyyllisenä. Moni luokka siis saa käyttävän luokan tyyppin templaatissa ja parametrinaan viitteen käyttävään luokkaan. Tämän sijaan luokkaa käyttävän luokka voisi periä virtuaalisen luokan, jossa on nimetty ne metodit joita käytetty luokka tulee kutsumaan, Javan liittymien tapaan. Lisäksi kaikkille luokille voisi määrätä isäntäluokan, jolloin takaisinkutsuminen onnistuisi tyyppinmäärittelyttä.

Uskon että alusta tulee hoitamaan tehtävänsä PROACT projektissa riittävällä varmuudella ja nopeudella. Uskon myös että, mikäli alustaa kehitetään edelleen, siitä voi olla paljonkin hyötyä useammassakin projektissa; Oli kyseessä tuntopalaute- tai jokin muu projekti. Alusta rakenteelta muuttuu yhä enemmän joustavammaksi ja uusien ohjelmointikielien mukaantulo lisää alustan potentiaalisia käyttötapoja.

7. Viitteet

[Bellifemine *et al.*, 1999] Fabio Bellifemine, Agostino Poggi, Giovanni Rimassa, JADE -- A FIPA-compliant agent framework. *Proceedings of PAAM'99*, 97 - 108.

[Bellifemine *et al.*, 2000] Fabio Bellifemine, Agostino Poggi, Giovanni Rimassa and Paola Turci, An Object-Oriented Framework to Realize Agent Systems. *Proceedings of WOA2000 Workshop*, 52 - 57.

- [Bellifemine *et al.*, 2001] Fabio Bellifemine, Agostino Poggi, Giovanni Rimassa, JADE: a FIPA2000 compliant agent development environment. *Proceedings of the 5th international conference on Autonomous agents*, 216 - 217.
- [CORBA, 2002] Common Object Request Broker: Core Specification. Draft, 2002. Available as http://www.omg.org/technology/documents/corba_spec_catalog.htm.
- [Finin *et al.*, 1994] Finin Tim, Fritzon Rich, McKay Don and McEntire Robin, KQML -- A Language and Protocol for Knowledge and Information Exchange, *13th International Distributed Artificial Intelligence Workshop*, 28 - 30.
- [FIPA00001] FIPA Abstract Architecture specifications. Draft, 2002. Available as <http://www.fipa.org/specs/fipa00001/SC00001L.html>.
- [FIPA00019] FIPA 97 Part 1 Version 2.0: Agent Management Specification. Draft, 2001. Available as <http://www.fipa.org/specs/fipa00019/OC00019A.html>.
- [FIPA00003] FIPA 97 Part 2 Version 2.0: Agent Communication Language. Draft, 2001. Available as <http://www.fipa.org/specs/fipa00003/OC00003A.html>.
- [FIPA00017] FIPA 97 Part 3 Version 1.0: Agent Software Integration Specification. Draft, 2001. Available as <http://www.fipa.org/specs/fipa00012/OC00012A.html>.
- [FIPA, 2002] Standard Specifications. Draft, 2002. Available as <http://www.fipa.org/repository/standardspecs.html>.
- [Martin *et al.*, 1999] D. L. Martin, A. J. Cheyer and D. B. Moran, The Open Agent Architecture: A framework for building distributed software systems, *Applied Artificial Intelligence: An International Journal*. Volume 13, Number 1 - 2, 91 - 128.
- [O'Brien and Nicol, 1998] P. D. O'Brien and R. C. Nicol, FIPA -- towards a standard for software agents, *BT Technology Journal*, Volume 16:3, 51-59.
- [Patil *et al.*, 1992] Patil Ramesh S., Fikes Richard E., Patel-Schneider Peter F., McKay Don, Finin Tim, Gruber Thomas, Neches Robert, The DARPA Knowledge Sharing Effort: Progress Report, *Proceedings of the Third International Conference of Principles of Knowledge Representation and Reasoning*.
- [Rao and Georgeff, 1995] A. Rao and M Georgeff, *BDI agents from theory to practice. Technical Note 56, AAIL, April.*

3D-veisto-ohjelman työkalujen käytettävyys

Nina Sainio

Tiivistelmä.

Tutkimuksessani tarkastelen 3D-maailman veisto-ohjelmaa ja kolmiulotteisuuden hahmottamista. Tutkin erityisesti Tampereen yliopiston tietojenkäsittelytieteiden laitoksella, TAUCHI-yksikössä tehtyä prototyyppiä ja sen nykyisiä, jo olemassa olevia hahmotustyökaluja. Vertailen työkalujen toimintaa eri tehtävien aikana. Tutkimukseni painottuu käytettävyystestiin, jonka tulosten perusteella voin päätellä, miten nykyiset hahmotustyökalut toimivat.

Avainsanat- ja sanonnat: 3D-Veistäminen, Suorakäyttöisyys, Kaksikäisyys
CR: H. 5.2, I. 3.4

1. Johdanto

Ajatus 3D-veistämisestä on peräisin jo 1970-luvun lopulta. Tuolloin keskivertotietokoneet eivät olleet riittävän tehokkaita pyörittämään monimutkaisia ja tarkkoja 3D-malleja reaaliaikaisesti. 3D-mallit olivat hyvin yksinkertaisia rautalankamalleja [Parent, 1977] ilman pinnoitteita ja varjostuksia, jotka ovat nykyään hyvin arkinen osa tietokonegrafiikkaa ja 3D-mallintamista. Rautalankamalleista siirryttiin 1990-luvun alkupuolella jo varsinaiseen veistämiseen, kun Galyean ja Hughes [1991] kehittivät pursottamiseen tarkoitetun työkalun.

Veisto-ohjelmat eroavat tavallisista 3D-mallinnusohjelmista siten, ettei käyttäjän tarvitse tuntea matematiikkaan tai ohjelmointiin perustuvaa mallinnusmenetelmää tai ohjelman syvällistä käyttöä, vaan hän voi toimia kuten taiteilija, luoden esineitä ja palikoita käsillään ja reaalielämää vastaavilla työkaluilla. Veisto-ohjelmien lopputuotokset eivät yleensä ole matemaattisesti yhtä tarkkoja kuin yleisempien 3D-mallinnusohjelmien tuotokset [Galeyan and Hughes, 1991].

Veisto-ohjelmien käyttötapa tekee niistä niin sanottuja suorakäyttöisiä ohjelmia. Suorakäyttöisessä ohjelmassa komennot suoritetaan fyysisinä toimintoina tai napinpainalluksina ilman, että käyttäjän tarvitsee hakea komentoja valikoista tai paleteista. Toiminnot ovat nopeita, toisiaan täydentäviä ja helposti peruutettavissa. Lisäksi suoritettujen toimintojen tulokset näytetään käyttäjälle reaaliaikaisesti [Shneiderman, 1983]. Ohjelman käyttäminen kuvaa mahdollisimman hyvin todellista veistämistilannetta. Veistäminen tapahtuu vain tietokoneen avulla.

Henkilöille, jotka eivät ole kovin tottuneita käyttämään tietokonetta, on tärkeää, että ohjelmat ovat mahdollisimman toimivia ja ymmärrettäviä. Toisaalta kokeneille käyttäjille on tärkeää, että he voivat toimia tehokkaasti muun muassa erilaisia oikopolkuja käyttäen [Nielsen and Molich, 1990]. Hyvin toimivat toiminnot ja työkalut ovat ohjelmissa oleellisia miellyttävän käyttökokemuksen saamiseksi kenelle käyttäjälle tahansa. Kolmiulotteiseen maailmaan siirryttäessä toimintojen, ohjaimien ja työkalujen käytettävyyden ainoastaan korostuu, sillä 3D-maailma tuo uusia haasteita, joihin käyttäjän on keskityttävä.

Tietokoneruudulla on vaikea hahmottaa kolmiulotteista avaruutta näytön kaksiulotteisuudesta johtuen [Nielsen and Olsen, 1987]. Kolmiulotteisen avaruuden ymmärtäminen vaatii tietoa siitä, miten päin maailmaa tai kappaletta käyttäjä tarkastelee. Ihminen havainnoi jokapäiväisessä elämässään hyvin kolmiulotteisuutta, mutta abstraktimpaan 3D-ympäristöön siirryttäessä ihmisen on vaikea ajatella etäisyyksiä ja varsinkin syvyyttä [Hinckley *et al.*, 1994]. Veisto-ohjelmassa pitäisi kuitenkin pystyä havainnoimaan kaikkia kolmea suuntaa tarkasti, vaikka maailmaa pyöriteltäisiin koneen ruudulla alkutilanteen jälkeen. Tutkimustulokset 3D-maailmasta ja sen ymmärtämisestä ovat tärkeitä, jotta voitaisiin tehdä parempia ohjelmia ja työkaluja, jotka helpottavat kolmiulotteisen maailman ymmärtämistä.

Veisto-ohjelma voidaan toteuttaa muun muassa niin, että ohjaukseen käytetään yhtäaikaista molempia käsiä. Tutkimuksessani käytetty prototyyppi on esimerkiksi kaksikätsiä hyödyntävästä veistosovelluksesta. Maailmaa ja veistettävää palikkaa voidaan tutkia näin luonnollisesti käännettä molempia käsiä hyväksi käyttäen. Palikan pyörittely toisella kädellä ja itse veistotyökalun liikuttaminen toisella voi tuottaa käyttäjälle aluksi lievää hämmennystä, mutta myöhemmin käsien rinnakkaiskäyttö ja yhteistyö tuntuneen luontevalta. William Buxton ja Brad A. Myers [1986] huomasivat tutkimuksissaan, että kahden käden yhteiskäyttö ja ohjaimien oppiminen mahdollistavat tehokkaamman työskentelyn kuin jos sama työ tehtäisiin vain yhtä kättä käyttämällä. Tehokkuus lisääntyy molempia käsiä käytettäessä, jolloin käyttäjällä turhaa aikaa ei kulu siihen, että hän yrittäisi yhdellä kädellä tehdä kahdelle kädelle tarkoitettua työmäärää.

Veisto-ohjelmat eivät ole vielä laajasti yleistyneet, eikä olemassa olevien sovellusten toiminnot ja ohjaimet ole vielä yhtenäistyneet. Ohjaimina veisto-ohjelmissa käytetään niin tavallista hiirtä ja näppäimistöä kuin keskivertokäyttäjälle vieraampia välineitä, kuten pallohiirtä, kuuden-suunnanohjainta tai Phantomia [Sensable 2003a]. Muun muassa tämän hetken markkinoiden tunnetuin ja monipuolisin veisto-ohjelma FreeForm käyttää tuntopalautteen antavaa Phantomia ohjauksessaan [Sensable, 2003b]. On muistettava, että kaksikätsiä ja

ohjainvalinnat saattavat aiheuttaa tyypilliseen PC-työskentelytapaan tottuneelle, aloittelevalle veisto-ohjelman käyttäjälle aluksi kognitiivisia lisäponnisteluja, kun uudet ohjaimet ja työskentelytavat vaativat keskittymistä. Käytettävyys kaikinensa on huomattavan tärkeää, eikä ohjelma saisi tuottaa käyttäjälle lisähuolta huonojen ohjaimien ja työkalujensa takia.

Tässä tutkimuksessa tulen tarkastelemaan, miten käyttäjät hahmottavat veisto-ohjelmassa kolmiulotteisen maailman ja miten he pystyvät toimimaan käyttäen ohjaamiseen yhtäaikaaisesti molempia käsiään. Tutkimuksessani käytetään Tampereen yliopiston tietojenkäsittelytieteiden laitoksella suunniteltua ja toteutettua veisto-ohjelmaa [Parviainen, 2004]. Pyrin selvittämään, kuinka paljon tietoa käyttäjä tarvitsee maailmasta, jotta hän ymmärtäisi etäisyyksiä ja suuntia, sekä miten ohjelman perustoiminnot toimivat. Ohjelma on vielä prototyyppi vaiheessa ja tutkimuksen tuloksia voidaankin käyttää ohjelman jatkokehityksessä.

Seuraavassa kappaleessa on tarkemmin esitelty yleisesti veisto-ohjelmien syntyyn ja kehitykseen vaikuttavia teorioita. Tutkimukseni kannalta tärkein asia on se, miten ihmiset havaitsevat 3D-maailmaa. Käsittelen erikseen myös suorakäyttöisyyden, kaksikäätisyyden ja metaforien merkityksen ohjelmille. Luvussa kolme kerron tarkemmin testattavana olevasta prototyypistä ja ongelmista, jotka tiedostetaan ennen testien tekemistä. Luku neljä kokoaa yhteen testausilanteeseen liittyvät asiat: testitehtävät, testauksessa käytetyt hypoteesit ja esimerkiksi testihenkilöiden kuvaukset. Viidennessä luvussa tarkastellaan käytettävyystestistä saatuja tuloksia ja lopuksi on yhteenveto koko tutkimuksen tuloksista ja onnistumisesta.

2. Veisto-ohjelmien taustaa

Veisto-ohjelmat voivat olla monella tapaa erilaisia kuin monet tutut tietokone-sovellukset. Veisto-ohjelmilla on yleisesti ottaen taustalla muutamia peruspiirteitä, jotka ovat muokanneet niistä nykypäivän veisto-ohjelmien kaltaisia; veisto-ohjelmat ovat 3D-ohjelmia, suorakäyttöisiä ja joissakin tapauksissa niitä käytetään yhtäaikaisesti molempia käsiä käyttäen. Kolmiulotteisuuden havainnoiminen on ihmisille arkipäiväistä. Kuitenkin kaksiulotteisessa maailmassa, kuten tietokoneruudulla, kolmiulotteisen kappaleen hahmottaminen on hyvin erilaista kuin mihin jokapäiväisessä elämässä ihminen on tottunut. Tässä luvussa keskitytään edellä mainittujen piirteiden tuomiin ongelmiin, mahdollisuuksiin ja etuihin. Luvun lopuksi käsitellään vielä hieman metaforia, jotka ovat oleellinen osa suorakäyttöistä käyttöliittymää ja näin ollen samalla veisto-ohjelmia.

2.1. Suorakäyttöisyys käyttöliittymässä

Suorakäyttöisyys on saanut alkunsa tarpeesta määritellä uudenlainen käyttöliittymä. Ben Shneiderman otti suorakäyttöisyyden määritelmän ensimmäisen kerran esille [1982], kun hän mietti tehokkaampia, parempia ja helpompia tapoja vaikeiden komentopohjaisten käyttöliittymien käyttämiseksi. Shneiderman määritteli suorakäyttöisyyden käyttöliittymässä seuraavasti:

- Monimutkaisten komentokielisten käskyjen sijaan tehtävät suoritettaisiin napin painalluksilla ja muilla fyysisillä toiminnoilla, kuten osoittamalla.
- Toimintojen tulokset näkyisivät välittömästi käyttäjälle.
- Toimintojen tulisi olla nopeasti lisättävissä ja näkyvillä sekä tarpeen tullen ne voisi peruuttaa.

Shneidermanin mukaan suorakäyttöisyys vanhoihin käyttöliittymiin verrattuna lisää muun muassa käyttäjien tunnetta siitä, että he hallitsevat järjestelmää ja että he pystyvät oppimaan ohjelman uudet piirteet helposti. Käyttäjät olivatkin ensimmäisistä suorakäyttöisistä käyttöliittymistä innoissaan ja halukkaita kertomaan kokemuksistaan ohjelman uudeltaisesta käytöstä muille käyttäjille. Nykypäivän keskiverto tietokoneenkäyttäjälle tällaisia onnistumisen elämyksiä tuskin samoissa määrin tulee, sillä suorakäyttöiset käyttöliittymät ovat levinneet kaikkialle ja ovat jokaisen käyttäjän ulottuvilla. Suorakäyttöisen käyttöliittymän esiinmarssi ja tietokoneiden nopea yleistyminen todistaneekin muun muassa sen, että komentopohjaisten käyttöliittymien käyttö on ollut hankalaa, eivätkä käyttäjät ole pystyneet tekemään asioita siten kuin olisivat kenties halunneet.

Käyttöliittymän käyttö helpottuu, kun ohjelman toiminnot tuodaan näkyville, kuten Shneiderman suosittelee. Tällöin niitä ei aina tarvitse etsiä valikoista tai antaa käskyjä kirjoittamalla jokin monimutkainenkin komento. Shneidermanin mukaan aloittelevat käyttäjät oppivat nopeasti ohjelman käytön, kun aluksi ei tarvitse opetella suurta joukkoa komentoja, mutta myös koneiden käyttäjien työteho paranee toimintojen muuttuessa helpommiksi ja nopeammiksi. Myöskään ohjelman käyttöä ei tarvitse pelätä, kun virhetoinnon voi aina peruuttaa tai muuten korjata missä ohjelman suoritusvaiheessa tahansa. Veisto-ohjelmien peruserätyyteen kuuluu tietynlainen vapaus tekemisen ja luovuuden suhteen, jolloin suorakäyttöisyys on erinomainen pohja veisto-ohjelmille.

2.2. Kaksikäisyys käyttöliittymässä

Tietokoneiden työ- ja kotikäytön lisääntyessä tietokoneen ohjausmenetelmät ovat kuitenkin pitkään pysyneet samoina. Jokapäiväisessä elämässä keskivertokäyttäjä käyttää ainoastaan näppäimistöä ja hiirtä – näitäkin usein eri aikaan. Ei liene liioiteltua väittää, että uudenlaiset työtavat, kuten molempien käsien yhtäaikainen ohjaus, voivat hämmentää aluksi käyttäjiä, jotka ovat vuosia tottuneet ajattelemaan näiden ohjaimien olevan ainoita mahdollisia ohjausvälineitä. Tulevaisuudessa ohjaimet muuttuvat ja niistä voidaan tehdä esimerkiksi sellaisia, jotka tukevat ihmisen luonnollista tapaa käyttää tehtävässään kahta kättä.

Buxton ja Myers [1986] tutkivat ensimmäisinä, miten kaksikäisyys vaikuttaa käyttöliittymiin. He tutkivat, millaisia eroja kaksikäisyydellä on yksikäisyyteen verrattuna ja saavutetaanko kaksikäisyydellä mitään etuja. Heidän mukaansa kaksikäisyys lisää tehokkuutta, kun molempia käsiä voidaan käyttää samanaikaisesti. Tällöin turhaa aikaa ei kulu, kun yhdellä kädellä yritettäisiin tehdä molempien käsien työmäärää. Buxton ja Myers osoittivat pienellä kokeella, että ihmisille on luonnollista käyttää kahta kättä yhtäaikaisesti, vaikka käsien toimintaedellytykset ovat hyvin erilaiset. Kaksikäisyyden tuoma mahdollinen lisäponnistelu liittyykin enemmän uudenlaisten ohjaimien oppimiseen kuin molempien käsien yhtäaikaiseen käyttöön.

Leganchuk *et al.* ovat päätyneet Buxtonin ja Myersin kanssa samansuuntaisiin tuloksiin [1999]. He toteavat, että kahden käden käyttö tuo kahdenlaisia etuja: kognitiivisia ja manuaalisia. Työn tekeminen nopeutuu, kun työmäärä jakautuu kahdelle kädelle yhden sijaan. Heidän mukaansa kognitiivinen kuorma taas pienentyy, kun käyttäjän ei tarvitse mielessään hahmottaa asioita niin paljon kuin jos tehtävä tehtäisiin vain yhtä kättä käyttämällä. Ihmiselle on tyypillistä hahmottaa ja havainnoida avaruudellisia asioita myös muiden aistien kuin näköaistin avulla. Tutkimuksissaan muun muassa Hinckley *et al.*, [1994] totesivatkin kahden käden käytön lisäävän käyttäjän kolmiulotteisen maailman ymmärrystä ja hahmottamista.

Monet arkipäiväiset asiat tehdäänkin molempia käsiä käyttämällä. Esimerkiksi taiteilija pitää veistettävää puupalaa heikommassa kädessään ja veistotyökalua hallitsevassa kädessään. Heikommalla kädellään hän kääntää puupalaa, jotta vahvempi käsi voisi kaivertaa helpommin ja tehdä tarkemman työn. Tällöin heikomman käden toiminta ikään kuin siirtyy vahvemmalle kädelle [Guiard, 1987]. Yleensä heikomman käden tehtävät ovat myös epätarkempia, jolloin se toimii hallitsevan tukena, apuna ja viitekehystenä. Hallitsevan ja heikomman käden toimintaedellytykset ovat siis huomattavan erilaiset, mutta yhdessä ne pystyvät toimimaan luontevasti parina ja tekemään monimutkaisiakin asioita.

Fittsin vuonna 1954 julkaiseman teorian mukaan kohteen osoittamiseen kulutettavan ajan voi ennustaa, kun tunnetaan kohteen koko ja etäisyys [Fitts, 1954]. Fittsin laki perustuu yksiulotteiseen osoittamiseen ja sitä onkin laajennettu myöhemmin toimimaan kaksiulotteiseen maailmaan [MacKenzie and Buxton, 1992]. Mackenziin ja Buxtonin johtamaa laajennettua lakia voitaisiin soveltaa myös kolmiulotteiseen maailmaan [Amento *et al.*, 1996]. Amento *et al.* mukaan lakia voidaan yleistää kahden käden käyttöliittymiin ja tapauksiin, joissa kohdetta pyritään osoittamaan molempien käsien yhteistyön tuloksena tai siten, että molemmat kursorit osoittavat samaan kohteeseen. Fittsin laajennettu laki on toimiva ja sillä on voitu suunnitella toimivampia käyttöliittymäosia veisto-ohjelman kaltaisiin käyttöliittymiin.

2.3. Kolmiulotteisuuden havaitsemisen teoriaa

Edellisessä aliluvussa tuli aiemmin jo esille, että ihminen hahmottaa monin eri asioin kolmiulotteista maailmaa – ei vain näköaistin avulla. Ihmisille kolmiulotteisen maailman havainnointi on helppoa ja jokapäiväistä [Hinckley *et al.*, 1994], eikä hän ilman kolmiulotteisuuden hahmottamisen kykyä edes selviä arkipäiväisistä asioista. Esimerkiksi jo kotona liikkuminen tai liikenteen seassa autoilu vaativat etäisyyksien havaitsemista.

Kaksiulotteinen tietokoneruutu muuttaa 3D-kappaleen hahmottamisen täysin. Hinckley *et al.* [1994] mukaan ihminen on kokenut havainnoija, kun kyse on todellisen elämän hahmottamisesta. Ihminen kerää tietoa ympärillä olevasta maailmasta ja pystyy muodostamaan yhteyksiä kokemustensa perusteella. Hinckley *et al.* toteavat, että vaikeudet kolmiulotteisen maailman hahmottamisesta alkavat, kun kappaletta käsitellään abstraktissa 3D-ympäristössä, kuten tietokoneruudulla. Tietokoneruutu kaksiulotteisuudestaan johtuen ei näytä kappaletta ”oikein”. Käyttäjä ei pysty kiertämään kappaletta taakse, eikä havainnoi sen syvyys-suuntaista pituutta yhtä helposti. Kappaletta voi toki kääntää ruudullakin ja siten tarkastella sen takapuolta, mutta käyttäjä ei pysty ottamaan sitä käteensä eikä esimerkiksi tuntoaistinsa perusteella pysty muodostamaan samanlaista kokonaiskuvaa kappaleen muodosta.

Ihminen tarvitsee siis jonkinlaista tukea havainnoidakseen kolmiulotteisuutta oikein. Tyhjä tietokoneen ruutu ei yksinään anna minkäänlaista tukea syvyys-suuntaiselle tarkastelulle. Hinckley *et al.* ovat tutkineet asioita, jotka helpottavat syvyysnäkemistä 3D-ohjelmissa. Tällaisia helpottavia asioita ovat muun muassa:

- viitteet avaruudellisiin tiloihin. Kaksiulotteisella monitorilla voidaan esimerkiksi syvyyden havaitsemista parantaa lisäämällä ohjelman näyttöön kolmatta ulottuvuutta kuvaava perspektiivi.

- kaksikäisyys. Kaksikäisessä käyttöliittymässä käyttäjä pystyy kokemaan itsensä paremmin osaksi 3D-maailmaa, jolloin hän oman kehonsa avulla pystyy tulkitsemaan avaruutta paremmin.
- moneen aistiin perustuva palaute. Esimerkiksi tunto- tai äänipalautteen lisääminen antaa paremman ja luontevamman kokonaiskuvan maailmasta.
- ehdottomat tai suhteelliset toiminnot. Hinckley *et al.* tarkoittavat ehdottomilla toiminnoilla esimerkiksi sitä, että työkalun ohjaimella on tietty kiinnitetty paikka reaali maailmassa, joka vastaa 3D-maailman työkalun paikkaa ohjelmassa [Galyean and Hughes, 1991]. Näin ollen aina tiedetään, mitä kohtaa ohjelmassa muutetaan. Suhteellisilla he tarkoittivat esimerkiksi tilanteita, joissa käyttäjä voi ottaa veistettävän kappaleen virtuaaliseen käteen ja pyöritellä sitä, miten haluaa. Muokattavan kohdan paikka kappaleessa, eli samalla työkalun paikka, määritellään virtuaalisen kappaleen ja virtuaalisen työkalun välisen suhteen avulla. Tällöin reaali maailman työkalun ja kappaleen paikalla ei ole merkitystä. Paikka ei siis ole kiinteä, jolloin työstäminen on vapaampaa.
- fyysiset rajoitteet ja edut. Fyysiset palautteet vähentävät kognitiivista kuormaa. Käyttäjän ei tarvitse näin ollen ymmärtää rajoitteita ja palautteita, kun hän voi kokea ne fyysisinä tuntemuksina.

Kolmiulotteisen maailman kokeminen ja ymmärtäminen ovatkin kaksi eri asiaa. Tietokoneohjelmissa ei välttämättä aina muisteta ottaa huomioon kaikkea, mikä vaikuttaa kolmiulotteisuuden parempaan havainnointiin. Havainnointiin vaikuttaa muun muassa kappaleiden päällekkäisyys, varjostukset, suhteellinen koko, perspektiivi ja pintakuvioidin väriliukumat [Haber and Hershenson, 1973]. 3D-maailmassa kappaleet sijaitsevat eri tasoissa ja ovat siis toistensa takana. Kun siirrytään toiseen paikkaan tai käännetään maailmaa, käyttäjä voi päätellä kappaleen olevan kauempana kuin toinen kappaleen ollessa etummaisena osittain peittävä. Etummaisesta kappaleesta muodostuu takana olevien päälle varjoja valon suunnasta riippuen. Käyttäjä voi päätellä myös etäisyyksiä esimerkiksi siitä, että normaalisti samankokoiset kappaleet ovatkin jossakin tilanteissa erikokoisia - pienempi on kauempana, kuten reaalielämässä kauempana oleva esine näyttää olevan pienempi kuin lähellä oleva. Kaikkien näiden asioiden tulisi olla kunnossa, jotta havainnointi olisi mahdollisimman luonnollista.

Tutkimukseni veisto-ohjelmassa on pyritty miettimään sitä, miten käyttäjä pystyisi havainnoimaan nimenomaan kolmiulotteisuutta paremmin ympäristössä, jossa viitteitä kolmiulotteisuudesta ei juuri ole. Ohjelmaan on

yritykseen tehdä työkaluja, joilla havainnointia on pyritty parantamaan. Työkaluja on muun muassa erilaiset koordinaatistot, laatikot ja tasot. Myöhemmin lisättävä tuntopalautte lisänee ymmärtämistä entisestään, kun aistimus ei perustu enää vain näköaistiin.

2.4. Metaforat ja niiden merkitys

Shneiderman [1982] määritteli suorakäyttöisyyttä muun muassa niin, että tehtävien toiminnot suoritettiin esimerkiksi painamalla nappeja ja että näiden toimintojen on oltava helposti käyttäjän käytettävissä. On luonnollista, että tällöin käyttöliittymäsuunnittelussa oli kiinnitettävä huomiota uudenlaiseen tiedon esitystapaan.

Komentokieliset käskyt korvattiin metaforilla, vertauskuvilla. Metaforat ovat esimerkiksi nappeja ja käyttöliittymän osia, joilla on tietty tehtävä ohjelmassa ja jotka joko ulkoisesti tai toiminnallisesti muistuttavat normaalielämän aitoja tilanteita. Esimerkiksi tiedosto, jota ei enää tarvita voidaan heittää tietokoneella *roskakoriin*, kuten tavallisessakin elämässä tai veisto-ohjelman puukappaletta voidaan veistää taltan näköisellä työkalulla, kuten miehet ovat vuosisatoja tehneet. Carrol *et al.* [1988] mielestä metaforien yksi merkittävä etu onkin siinä, että niitä käyttämällä voidaan hyödyntää käyttäjien aikaisimpia tietoja ja kokemuksia asioiden yhteyksistä ja toiminnasta. Sen sijaan, että vähennettäisiin ohjelman monimutkaisia ja laajoja käyttötapoja, voitaisiin heidän mukaan ohjelmaan lisätä tuttuja elementtejä, toimintoja ja menetelmiä.

Veisto-ohjelma asettaa 3D-ohjelmana myös metaforille uusia haasteita. Metafora käsitettä voidaan laajentaa ohjaimiin asti normaalielämän tietokoneen ruudulla tapahtuvien toimintojen lisäksi. Veisto-ohjelmissa voi olla esimerkiksi kaiverrus- ja pursotustikkuna tuntopalautteen antava Phantom [Sensable, 2003a], joka kuvaa hyvin realistisesti aitoa veistämistilannetta. Veistettävää kappaletta voidaan ohjata myös esimerkiksi hiirellä ja jollakin 3D-ohjaimella. LeBlank *et al.* [1991] ovat tutkineet hiiren ja Spaceballin yhteiskäyttöä. Spaceball on ohjaimena kuudensuunnanohjain, jolla voidaan liikuttaa kappaletta myös syvyys suunnassa. Spaceballin avulla kappaleen pyörittäminen voidaan tehdä heikommalla kädellä, jolloin itse työkalun käyttäminen jää vahvemman käden tehtäväksi. Näin käyttäjä voi ikään kuin pitää kappaletta heikommassa kädessään ja kääntää sitä tarvittaessa hallitsevan käden vapauduttua työstämään kappaletta. Tätä samaa metaforaa on käytetty myös testattavanani olevassa ohjelmassa.

2.5. Yhteenveto

Vaikka kaksikäisyys ja 3D-havainnointi ovatkin ihmiselle luonnollisia asioita, on niiden käyttäminen keinotekoisessa ympäristössä usein erilaista ja vaikeampaa kuin mihin arkielämässä olemme tottuneet. Kaksiulotteisessa ympäristössä on vaikea havainnoida kolmiulotteista maailmaa, jollei minkäänlaista tukea havaitsemiseen saada esimerkiksi ohjelman näkymästä. Veisto-ohjelmaa suunniteltaessa ja toteutettaessa on muistettava muutamia perusasioita: ohjauksen tulisi olla helppo, syvyyssäkemistä tuettava ja käyttöliittymän muutenkin kaikin osin toimiva. Ohjausta voidaan parantaa vain löytämällä oikea yhdistelmä ohjausvälineistä. Ohjelman on sisällettävä avaruuden tiloja tukevia viittauksia ja toimintojen on oltava yksinkertaisia sekä helposti käyttäjän saatavilla.

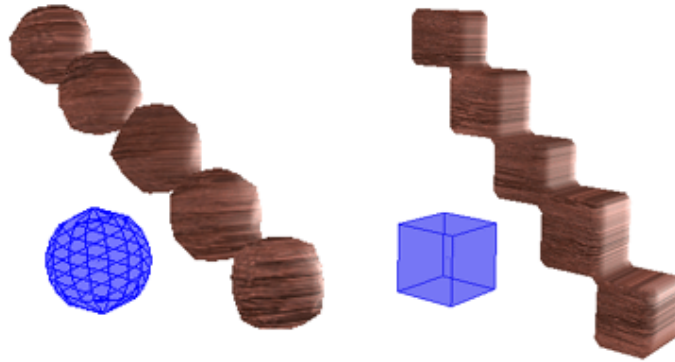
3. Testiohjelma

Tässä kappaleessa on tarkoitus esitellä hieman tutkimuksen kohteena olevaa prototyyppiä ja sen piirteitä. Testien pohjana ovat prototyypin testien aikainen tila ja sen toiminnot. Koska kyseessä on kehityksen aikainen työversio, ei ohjelmaan ole testeihin mennessä ehditty tekemään kaikkia toimintoja siinä laajuudessa ja määrin, joita sekä testin suorittaja että ohjelman suunnittelija olisivat halunneet.

3.1. Sculpting-veisto-ohjelma

Tutkimuksessani testataan Tampereen yliopiston tietojenkäsittelytieteiden laitoksella tehtyä veisto-ohjelmaa [Parviainen, 2003]. Prototyyppi toimii tavallisella hiirellä ja kuudensuunnanohjaimella, Magellanilla [3Dconnexion, 2003]. Prototyyppi on siis sovellus, jota ohjataan kahdella kädellä yhtäaikaaisesti.

Ohjelmaan on toteutettu veisto-ohjelmien perustyökaluista pursotus- ja kaiverrustyökalut. Molempia työkaluja voi käyttää sekä pallo- että laatikkotilassa, eli lisättävä tai poistettava osa on joko pallon tai laatikon muotoinen alue (Kuva 1). Työkalujen kokoa voi muuttaa. Näillä työkaluilla käyttäjä pystyy jo veistämään ja pursottamaan realistisen näköisiä esineitä ja asioita.



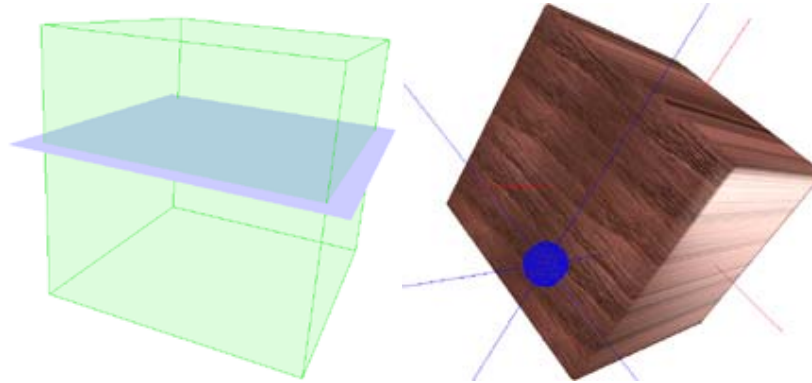
Kuva 1. Pursotuksessa käytetty pallo- ja laatikkotyökalu

Veistämisen voi aloittaa joko pursottamalla materiaalia tyhjälle *työpöydälle*, joksi tässä kutsun ohjelman veistoaluetta, tai valitsemalla aloituskappaleen, jota voi veistää edellä mainituilla työkaluilla. Aloituskappaleen muodon voi valita halutun malliseksi riippuen siitä, mitä on tekemässä. On mielekkäämpää valita aloituskappaleeksi pallo kuin neliö, jos on tekemässä omenaa.

Ohjelmaan on lisätty 3D-maailman havaitsemista auttavia työkaluja (Kuva 2). 2D-maailmassa, kuten esimerkiksi paperilla, kolmiulotteista avaruutta on totuttu kuvaamaan xyz-koordinaatistolla. Tätä menetelmää on käytetty ohjelmassa hyväksi kahdella eri tavalla. Tarvittaessa käyttäjä voi asettaa työpöydälle koordinaatiston, joka kääntyy maailmaa käännettäessä. Tämä koordinaatisto pysyy aina kiinteästi kiinni origossa, joka on työpöydän keskipisteessä. Toinen xyz-koordinaatisto voidaan kiinnittää työkaluun, jolloin käyttäjä pystyy hahmottamaan, milloin työkalu koskettaa muokattavaa osaa kappaleessa.

Havainnointia helpottamaan ohjelmaan on luotu rajauslaatikko. Rajauslaatikko on 3D-kappale, joka koordinaatistoon verrattuna havainnoi paremmin etäisyyksiä ja syvyyksiä. Koordinaatisto näyttää vain suoria, kun rajauslaatikko tasoja. Samalla se näyttää veistoalueen rajat. Tulevaisuudessa rajauslaatikkoa tullaan kehittämään havainnollisemmaksi.

Rajauslaatikosta ja koordinaatistosta on syntynyt ajatus hahmotustasosta. Tason avulla käyttäjä pystyy paremmin hahmottamaan sen, mitkä osat kappaleesta ovat samassa tasossa. Työkalun avulla uuden materiaalin pursottaminen tai vanhan poistaminen käy helposti etäisyyksien ja syvyyksien selkeydessä. Myöhemmin tätäkin työkalua tullaan kehittämään esimerkiksi niin, että tasoa vasten voi asettaa pursotettavia osia ja tasata erillisiä kappaleita samaan tasoon.



Kuva 2. Hahmottamista helpottavia työkaluja: hahmotuslaatikko ja -taso sekä origokeskeinen koordinaatisto ja työkaluun liitettävä koordinaatisto

3.2. Ennen testiä tunnetut ongelmat

Koska testattava ohjelma on prototyyppi, on se huomattavan monilta osiltaan vielä vajavainen. Ongelmat pyritään korjaamaan mahdollisimman nopeasti jatkokehityksen alkaessa. Muutoksia ei kuitenkaan ennätetty tehdä ennen näitä testejä. Seuraavassa esitellään muutamia ongelmia, joita valmiissa ohjelmassa ei olisi.

Suurin ongelma on tällä hetkellä ohjainten huono käyttömenetelmä. Esimerkiksi työkalua liikuteltaessa tulee olla oikea hiiren nappi pohjassa ja kaiverrettaessa vasen hiiren nappi. Mikäli käyttäjä haluaa kaivertaa ja liikuttaa samanaikaisesti kappaletta, pitää hänen pitää molempia hiiren nappeja alhaalla yhtäaikaaisesti. Tämä on kognitiivisesti raskasta, eikä suositeltavaa. Tämä ominaisuus pyritään korjaamaan ensitilassa. Korjaus voi tapahtua kehittämällä parempi menetelmä hiiren käytölle tai korvaamalla ohjain jollain muulla ohjaimella.

Muita ohjaukseen liittyviä, tiedostettuja ongelmia on esimerkiksi työkalun koon muutos ja hahmotustason liikuttaminen. Muutos ja liikuttelu tapahtuvat Magellanin napeista. Tällä hetkellä nappeja ei voi pitää pohjassa, jolloin koko muuttuisi tasaisesti, vaan nappeja täytyy tällä hetkellä naputtaa useasti koon muuttuessa asteittain. Korjausehdotuksena voisi olla ohjelmaan jonkunlaisten liikusäätimien tekeminen tai Magellanin uudelleen ohjelmointi.

Tällä hetkellä hiiren kursori ja työkalu ovat kaksi eri asiaa. Kursoria ei kuitenkaan vielä ole poistettu näytöltä käyttäjän veistäessä tai pursottaessa työkalulla. Kursori ja työkalu on haluttu pitää erillään, jotta työkalu voitaisiin jättää työpöydälle sillä välin, kun käyttäjä menee valikkoon muuttamaan toimintoja. Kursorin näkyminen sekaannuttaa vain tällä hetkellä käyttäjää, joka on tottunut seuraamaan nimenomaan kursoria, eikä työkalua.

Ohjelmakoodi ei keskeneräisyytensä takia ole optimoitu. Sen takia ohjelman kuvien tarkkuus on huono. Myöhemmin algoritmeja parannetaan ja tarkkuutta lisätään. Tällöin kuvista saadaan entistä siistimpiä ja tarkempia. Ohjelmakoodin parantuessa piirtoaluetta voidaan myös suurentaa.

Prototyypin perspektiivi on tällä hetkellä liian jyrkkä eli kappale on hyvin lähellä katsojaa. Kappale pitäisi viedä kauemmaksi katsojasta, jotta perspektiivi ei olisi ylikorostunut. Kauemmaksi viety kappale tosin näyttää pienemmältä, jolloin piirtoaluekin näyttäisi pienemmältä.

3.3. Yhteenveto

Tampereen yliopiston tietojenkäsittelytieteiden laitoksella, TAUCHI-yksikössä tehty veisto-ohjelma on vielä prototyyppi. Prototyypissä on paljon tiedostettuja ongelmia, joita tullaan myöhemmin korjaamaan ja muuttamaan. Prototyyppi ongelmistaan huolimatta on jo toimiva, itsenäinen ohjelma, jolla pystyy veistämään yksinkertaisia asioita ja jolla voidaan testata käyttäjien toimia ja 3D-maailman hahmottamista.

4. Veisto-ohjelman työkalujen ja 3D-hahmottamisen testaus

Tässä kappaleessa käsitellään käytettävyydestään vaikuttaneita tekijöitä. Ensimmäisessä luvussa on kerrottu testin tarkoitus ja ennen testiä asetetut hypoteesit. Myöhemmin luvussa kerrotaan testiin osallistuneista henkilöistä ja miten ja miksi testitehtävät on valittu.

4.1. Testien tarkoitus ja hypoteesit

TAUCHIn prototyyppiin on luotu kolmiulotteisen maailman hahmottamiseen kehitettyjä työkaluja. Testieni tarkoituksena on tutkia hahmotustyökalujen toimintaa ja kolmiulotteisuuden hahmottamista. Tässä hahmotustyökaluiksi ajatellaan origokeskeinen koordinaatisto, työkaluun liitettävä koordinaatisto, rajauslaatikko ja hahmotustasot (Kuva 2). Pursotus- ja veistotyökaluista on tehty läpinäkyviä ja kolmiulotteisia hahmottamisen helpottamiseksi, mutta tässä testissä näihin ei niinkään kiinnitetä huomiota, vaan keskitytään edellä mainittujen työkalujen toimintaan.

Testitehtävät on valittu siten, että käyttäjiltä saataisiin kattava tulos ohjelman kolmiulotteisuuden hahmottamista helpottavista työkaluista. Testihenkilöt on jaettu kahteen eri ryhmään. Ensimmäisessä ryhmässä tehtävät aloitetaan niin, ettei käyttäjällä ole käytössään hahmotustyökaluja. Heille annetaan työkalut käyttöön testin kuluessa eteenpäin. Työkalut saatuaan he tekevät samat tehtävät uudelleen, nyt siis apuja käyttäen. Toinen ryhmä aloittaa testin päinvastoin. Heille annetaan alussa kaikki mahdolliset työkalut käyttöön

ja testin puolivälissä ne viedään heiltä pois. Tällaisella testiryhmän jakamisella voidaan välttyä oppimisen aiheuttamalta tulosten vääristymiseltä.

Voisi olettaa, että kolmiulotteisuuden hahmottamista helpottavat työkalut parantavat työnlaatua ja lisäävät käyttäjän onnistumisen iloa. Työkalujen käyttö ei välttämättä nopeuta työtä, sillä työkalujen avulla käyttäjät voivat tehdä työnsä tarkemmin ja paremmin. Siksi aikaa saattaa kulua enemmän, kun käyttäjät keskittyvät paremmin työhönsä. Käyttäjien mieltymystä työkaluihin ja niiden tarpeellisuutta tutkitaan viimeisissä tehtävissä sillä, että käyttäjälle annetaan vapaus valita, käyttääkö hän aputyökaluja vai ei veisto-ohjelmaa käyttäessään. Mikäli testihenkilö vapaaehtoisesti haluaa valita helpottavat työkalut, on hän selvästi mieltynyt käyttämään niitä, tuntee tarvitsevansa hahmottamiseen tukea ohjelmalta ja uskoo työkalujen auttavan hahmottamisessa.

Työn nopeutta, laatua ja yrityskertojen määrää voidaan tarkkailla ohjelmaan tehdyn lokin ja käyttäjille tehdyn haastattelun ja kyselyn avulla. Lokiin kirjataan kaikki käyttäjän toiminnot ja se, kuinka kauan kuhunkin vaiheeseen on aikaa kulunut. Oletan, että hahmotustyökalut parantavat testitehtävissä vaadittavaa osoitustarkkuutta ja näin parantavat työn laatua. Kuten edellä tuli esille, aika ei välttämättä ole hyvä suure mittaamaan työkalujen toimivuutta. Testeissä tuleekin seurata, miten käyttäjät suhtautuvat veistämiseen kokonaisuutena. Jos käyttäjä suorittaa kaikki tehtävät yhtä suurella vakavuudella, oli kyseessä tehtävä työkalujen kanssa tai ilma, on oletettavaa, että tulos parantuu, kun käyttäjän kognitiivista kuormaa vähennetään visuaalisin vihjein siitä, miten 3D-maailma tietokoneruudulla käyttäytyy.

4.2. Testihenkilöt

Testiini osallistui 12 henkilöä. Koehenkilöt ovat kaikki tottuneita tietokoneen käyttäjiä ja näin ollen kaikki kuuluvat selvästi veisto-ohjelman mahdolliseen käyttäjäryhmään. Veisto-ohjelman kaltaisia ohjelmia käyttävät henkilöt luultavimmin harvoin ovat täysin vasta-alkajia tai esimerkiksi lapsia, sillä maailmaa on aluksi erittäin vaikea hahmottaa. Vaikka kaikki käyttävätkin säännöllisesti tietokonetta, voi testihenkilöistä tehdä selvän jaon kokeneisiin ja normaalikäyttäjiin.

Koehenkilöt ovat nuoria, alle 30-vuotiaita, jotka yleensäkin ovat innokkaita kokeilemaan uusia asioita ja ohjelmia. Testihenkilöiden keski-ikä oli 24 vuotta. Testihenkilöistä kaksi on naisia ja loput miehiä. Koehenkilöiden ryhmässä on kaksi graafikkoa, jotka ainakin voisivat käyttää veisto-ohjelmaa myös arkielämässään. He ovat käyttäneet aiemmin esimerkiksi vanhempia 3D-mallinnusohjelmia, kuten 3D Studio MAXia.

Koehenkilöt jaettiin kahteen eri testiryhmään. Ryhmät pyrittiin jakamaan testihenkilöistä tiedettyjen ennakkotietojen ja heidän 3D-käyttökokemusten perusteella mahdollisimman tasaisesti. Esimerkiksi edellä mainitut graafikot olivat eri ryhmissä. Molemmissa ryhmissä oli siis kuusi testihenkilöä.

4.3. Valmistelut ennen testiä ja testitehtävät

Ennen testiä käyttäjiä opastetaan ohjelman käyttöön. Heille kerrotaan edellä mainituista ongelmista, jotta he eivät kiinnittäisi niihin niin suurta huomiota testiä tehtäessä. Heille esitellään vieraat ohjaimet ja kerrotaan veisto-ohjelman pääperiaatteet. Ennen varsinaista testiä testihenkilöt saavat kokeilla vapaasti ohjelman toimintoja hetken ajan itsekseen totutellakseen hieman ohjaukseen.

Seuraavassa esitellään testitehtävät ja niiden tarkoitus. Tässä käydään läpi ainoastaan ensimmäisen ryhmän testit läpi. Toinen ryhmä tekee samat testit, vain eri järjestyksessä vääristymien ehkäisemiseksi.

Testit on jaettu kahteen eri osioon. Ensimmäiseen osioon kuuluu kuusi ensimmäistä tehtävää, joilla kontrolloidaan työkalujen tarpeellisuutta ja toimivuutta. Tehtävästä seitsemän eteenpäin testataan käyttäjien innokkuutta käyttää työkaluja ja heidän kykyä hahmottaa maailmaa ja ohjelmaa. Testit aloitetaan helpoimmalla tehtävällä. Siinä kuution muotoista kappaletta käsitellään niin, että jokaisen sivun keskipisteeseen käyttäjä kaivertaa suorakulmion muotoisen jäljen. Tarkoituksena heillä on tehdä mahdollisimman säännöllinen kappale, jossa joka sivulla oleva kuoppa on yhtä syvä ja yhtä kaukana kappaleen särmistä. Tätä tehtävää ensimmäisen kerran suoritettaessa ensimmäisen ryhmän jäsenille annetaan avuksi molemmat koordinaatistot, origo- ja työkalukoordinaatistot. Tehtävän pitäisi olla varsin yksinkertainen, kun koordinaattiakselit kulkevat suoraan sivujen keskipisteiden kautta ja työkalun koordinaatistosta käyttäjä näkee, missä kohta työkalu on puoliksi tasoon upotettuna.

Ensimmäinen tehtävä on varsin helppo ohjaukseltaan. Toisessa tehtävässä tilannetta vaikeuttaa se, että työkalua täytyy liikuttaa, eikä enää vain painaa kerran. Käyttäjällä on tässäkin tehtävässä apunaan koordinaatistot. Tällä tehtävällä pääsääntöisesti testataan, miten käyttäjä on oppinut käyttämään 3D-ohjaimia ja hahmottamaan ohjelmaa.

Kolmannessa tehtävässä käyttäjän pitää tehdä mahdollisimman säännöllinen noppa. Apuna hänellä on koordinaatistojen lisäksi hahmotustaso. Tasoja voisi käyttää asettamaan nopan pisteet samoihin kohtiin eri puolilla noppaa. Koordinaatistot taas auttavat muun muassa syvyyksien hahmottamisessa.

Näiden ensimmäisten tehtävien jälkeen tehdään samat tehtävät uudelleen. Nyt käyttäjille ei anneta työkaluja käyttöön vaan heidän pitää pystyä tekemään

yhtä säännöllisiä kappaleita kuin aiemminkin silmämääräisesti. Toisella ryhmällä tehtävät 1-3 ja 4-6 ovat toisin päin. He siis aloittavat testin niin, ettei heillä ole apuna hahmotustyökaluja.

Viimeiset tehtävät ovat yleisluontoisempia, eikä niillä voida sinänsä mitata yhtä ja ainutta asiaa. Molemmat ryhmät tekevät tästä eteenpäin samat tehtävät. Testihenkilöille ei anneta ennen testiä mitään muuta kuin kuva lopputilanteesta. Näin ollen he saavat päättää, lähtevätkö he pursottamaan vain kaivertamaan loppukuvan kaltaista kappaletta. Käyttäjät saavat itse valita, käyttävätkö he hahmotustyökaluja vai eivät.

Viimeinen tehtävä, testitehtävä kymmenen, on tarkoitettu testaamaan sitä, mitä ohjelmalla jo pystyy tekemään. Käyttäjä saa veistää, mitä hän haluaa. Aikaa, alkutilannetta tai työkaluja ei ole mitenkään määrätty ennalta. Viimeisten tehtävien suurin mielenkiinto on suurimmaksi osaksi siinä, minkä työkalun käyttäjä valitsee apuvälineeksi, jos hän kokee tarvitsevansa hahmottamiseen apua ja valitsee ko hän pursottamisen vai kaiverruksen eri tilanteissa.

4.4. Yhteenveto

Testini perustuvat olettamukseen, että käyttäjän on helpompi veistää ohjelmalla, kun hänellä on apuna hahmotustyökalut. Hahmotustyökaluihin lasken tässä koordinaatitot, hahmotustason ja -laatikon. Käytettävyydestin jälkeen tiedän paremmin, miten hahmotustyökalut toimivat tai auttavatko ne käyttäjää hahmottamaan paremmin 3D-maailmaa ja tehtävää, jota hänen pitäisi tehdä. Mitattavien tulosten lisäksi suurin mielenkiinto kohdistuu käyttäjien mielipiteisiin, joita he testin aikana puhuvat ääneen ja joita kerään testin jälkeen haastattelussa ylös.

5. Tulokset

Seuraavassa esittelen testeistä saatuja tuloksia. Prototyypin ollessa kehitysvaiheessa oli mitattavia tuloksia tärkeämpää saada mielipiteitä käyttäjiltä ja nähdä tuotoksia, joita he jo pystyivät keskeneräiselläkin ohjelmalla tekemään. Lähes kaikille testihenkilöille 3D-veistäminen oli kaikin puolin täysin uusi asia. Henkilöille, jotka olivat aiemmin käyttäneet mallinnusohjelmia, ohjelman käyttö oli alusta asti helpohkon oloista. Muutkin käyttäjät oppivat nopeasti ohjelman käytön, mutta tehtävien suorittamisissa on selvästi nähtävissä se, kuinka hahmottamisen oppimista tapahtui koko ajan. Pääsääntöisesti kaikki testihenkilöt tulkitsivat testin loputtua 3D-maailmaa paljon paremmin kuin ensimmäisten tehtävien aikana.

5.1. Testin varsinaiset tulokset

Kuten edellä on kerrottu, testihenkilöt oppivat huomattavan hyvin testin aikana käyttämään veisto-ohjelmaa. Hahmottaminen parani testin myötä kaikilla, joille kolmiulotteinen sovellus oli uusi tai vieras asia. Kokeneet käyttäjät osasivat heti testin alusta alkaen ajatella, miten kappale pyörii ja miten maailman tila muuttuu sitä käännellessä. Loput testihenkilöistä oppivat testin aikana. Kaikki henkilöt käyttivät kahta kättä erittäin luontevasti ilman minkäänlaisia ongelmia. Etukäteen pelkäsimme huonon ohjauksen antavan väärän kuvan ohjelman käytöstä, mutta ainoastaan yhdellä testihenkilöllä oli aluksi ongelmia työkalun ohjaamisen kanssa.

Käyttäjät totesivat lopuksi usein, että heille täysin uudenlainen sovellus ja käyttötapa johtivat siihen, että he eivät niinkään keskittyneet hahmotustyökalujen täysipainoiseen käyttämiseen, vaan he joutuivat enemmän keskittymään ohjelman opetteluun. Prototyypin ei siis vielä ole siinä vaiheessa, että sen käyttö olisi täysin itsestään selvää.

Ensimmäisessä ryhmässä, jossa käyttäjille aluksi annettiin hahmotustyökalut käyttöön ja sitten otettiin pois, ei ollut suuria eroja kappaleen säännöllisyydessä ja siinä, miten he ensimmäiset kuusi tehtävää suorittivat. Ilmeisestikin he ensimmäisten kolmen tehtävän aikana kiinnittivät yleisesti huomiota ohjelman käyttöön, eivätkä siksi juuri huomioineet työkaluja. Toinen testiryhmä taas paransi tuloksiaan osion jälkimmäisissä tehtävissä huomattavasti. Alussa heillä ei ollut käytössään työkaluja, mutta oppimisen ja työkalujen käyttöönoton jälkeen jälki oli selvästi parempaa kuin ensimmäisten kolmen tehtävän aikana.

Ensimmäisessä tehtäväsarjassa ryhmien välillä ensimmäisten kolmen tehtävän osalta tuloksissa ei ollut kovastikaan eroa. Tämä selittyy sillä, että tässä vaiheessa käyttäjät hieman totuttelivat ohjelman käyttöä. Toinen ryhmä suoritti kuitenkin tehtävät neljä, viisi ja kuusi selvästi tarkemmin ja paremmin kuin ensimmäinen ryhmä. Tällöin heillä oli hahmotustyökalut käytössään.

Yleisesti tärkein hahmotusväline oli työkaluun kiinnitettävä koordinaatisto, jonka avulla käyttäjät pystyivät havainnoimaan esimerkiksi sitä, kuinka syvällä kappaleessa työkalu kulloinkin oli. Toisaalta myös origokeskeinen koordinaatisto oli monelle erittäin tärkeä apu, kun he halusivat asettaa kappaleen kohtisuorasti itseään kohden.

Yleisesti ottaen tasosta ei ollut juuri apua. Sen huono ohjaus ja vajavainen toiminto selittänevät suurimman osan siitä, miksi testattavat eivät innostuneet sen käytöstä. Toisaalta heillä oli hyvin vähän kokemusta tämän kaltaisesta ohjelmasta. Kuten jotkut testihenkilöistä sanoivat, he ”eivät oikeastaan tienneet

mahdollisuuksista”, joita ohjelmalla voi tehdä. Toiset testihenkilöistä taas käyttivät erittäin luovasti tasoa suorittaessaan viimeisiä tehtäviä.

Hahmotusvälineistä hahmotuslaatikko oli käytössä järjestäen kaikilla viimeisiä tehtäviä tehtäessä. Tämä johtui yleensä siitä, että tehtäviä lähdettiin ennemmin pursottamaan kuin kaivertamaan, jolloin piirtoalueen rajat oli syytä tietää. Laatikkoa käytettäessä liiallinen perspektiivi teki hahmottamisesta joskus hankalaa.

Viimeisissä tehtävissä, joissa loppukuva oli annettu, pääosin jokainen testiin osallistunut valitsi työtavakseen pursottamisen. Kuvassa 3 on esitetty erilaisia tuoleja, joita käyttäjät ovat prototyypillä pursottaen tehneet. Esimerkiksi tehtävää kahdeksan suorittaessa, jossa piti tehdä malja, maljan veistämisen aloitti lähes 40 % testihenkilöistä kaivertamalla, mutta päätyivät lopulta pursottamiseen. Käyttäjiltä saatujen kommenttien perusteella osasyynä kaiverruksen muuttuminen pursottamiseen oli keskeneräisten kaiverrustyökalujen liian rajoittava muoto.

Kaikin puolin testit onnistuivat hyvin. Ainoastaan muutama tehtävä jäi suorittamatta. Syynä näihin keskeytyksiin oli testitilanteeseen varatun ajan loppuminen kesken. Käyttäjät pitivät hahmotusvälineitä tarpeellisina, mutta vielä keskeneräisinä.



Kuva 3: Testihenkilöiden luomia tuoleja

5.2. Parannusehdotuksia

Saimme testin aikana ja sen jälkeen käydyissä keskusteluissa käyttäjiltä paljon parannusehdotuksia. Osa ehdotuksista oli sellaisia, jotka ovat olleet ohjelman suunnittelijan ja testaajankin mielessä. Tässä on nyt kuitenkin listattu muutamia hyviä pikkuehdotuksia.

Monet kaipasivat jonkinlaista tukea sille, koska työkalu osuu kappaleeseen. Se, että työkalu muuttaa hieman ulkonäköään ei riitä, vaan käyttäjät kaipasivat muunkinlaista palautetta. Kuten Hinckley *et al.* [1994] totesivat, olisi hyvä olla

muidenkin aistien kuin vain näköaistin antamaa palautetta. Käyttäjät ehdottivat, että ohjelmaan lisättäisiin tuntopalaute, joka kertoo aina, missä kappaleen kulloisetkin rajat ovat. Näin olisi paljon helpompi kulkea pintaa pitkin työkalulla.

Heidän mielestään työkalua pitäisi muutenkin viedä tasoa pitkin, eli lukita se kappaleen pinnalle jotenkin. Näin ollen kappaletta käännettäessä työkalu pysyisi pinnalla riippumatta siitä, miten päin kappale on. Veistettäessä kappaleeseen ei kuitenkaan aina voi määrittää tasoa, sillä miten voidaan määritellä omenaa veistettäessä suora "taso", jonka suhteen työkalu kulkisi. Hiiren ollessa 2D-ohjain se ei ole paras vaihtoehto kolmiulotteista sovellusta käytettäessä, sillä hiirellä ei pysty kulkemaan luontevasti välisuuntia. Pinnalla kulkemista on lähes mahdoton toteuttaa luontevasti ilman tuntopalautetta ja oikeanlaista 3D-ohjainta.

Tällä hetkellä työkalu ja kappale ovat yhteydessä toisiinsa siten, että kun käyttäjä kääntää maailmaa, työkalu kääntyy mukana. Käyttäjät kaipasivat toimintoa, jossa työkalu ja maailma eivät liikkuisi samassa suhteessa. He haluaisivat erottaa työkalun ja maailman ohjauksen toisistaan. Toisaalta joskus tuntuu järkevältä pitää päällä nykyinenkin toiminto työkalun ja maailman yhteydestä. Ohjelmaan voisikin tehdä edellä mainitunlaisen toiminnon nykyisen lisäksi.

Käyttäjät eivät liiemmästi käyttäneet hahmotustasoa apuna. Tason vähäinen käyttö johtui varmastikin sen vielä vajavaisesta toiminnasta. Käyttäjät olisivat kaivanneet mahdollisuutta lisätä tasoa rajattomasti ja että tason olisi voinut lukita jollekin kohtaa kappaleessa. He olisivat toivoneet, että taso ei ainoastaan olisi ollut viitteellinen apu, vaan että tasoon olisi voinut asettaa työkalun esimerkiksi kiinni. Tason toivottiin reagoivan jotenkin, kun työkalu osuisi siihen. Taso voisi esimerkiksi vaihtaa väriä.

Koordinaatistoista käyttäjät kaipasivat mahdollisuutta lisätä niitä oman mielensä mukaan muuallekin kuin vain origoon tai edes niin, että he olisivat voineet kääntää alkuperäistä koordinaatistoa. Jotkut testihenkilöistä kaipasivat koordinaatistoihin koordinaattien piirtämistä. Tämä on käynyt suunnittelijan ja testaajankin mielessä, mutta matemaattisten koordinaattipisteiden laskeminen sotii lievästi veisto-ohjelmien perusideaa vastaan. Pisteitä ei siksi ole toistaiseksi koordinaatistoihin liitetty. Piirtoalueen ollessa pieni eräs testihenkilö ehdotti, että työkalun ja sen koordinaattiakseleiden väri voisi muuttua, kun alueelta poistuttaisiin.

Työkalun koordinaatisto on luotu sen takia, että sen avulla voitaisiin hahmottaa työkalun kulkua. Eräs testattava ehdotti kuitenkin, että työkalun

ääriviivat voisivat näkyä koko ajan, jolloin työkalun hahmottaminen olisi vielä helpompaa.

Toinen graafikkotestattavista ehdotti myös, että veisto-ohjelmalla tehty kappale voitaisiin tallentaa sellaisessa muodossa, että kappaleen voi ladata esimerkiksi 3D Studio MAXiin. Näin ensimmäisen työvaiheen voisi tehdä testiohjelmalla ja esimerkiksi värityksen jollakin toisella ohjelmalla.



Kuva 4: Testeissä tehtyjä kappaleita

5.3. Käyttäjien ajatuksia

Yleisesti testihenkilöt olivat innoissaan ideasta, miten veisto-ohjelmalla tehdään 3D-kappaleita. He käyttivät varsin luovastikin ohjelmaa ja saivat testin aikana tehtyä monia erilaisia kappaleita nalleista, maljoihin ja lohikäärmeen päihin (Kuvat 4 ja 5). Luonnollisesti testattavia häiritsi ohjelman keskeneräisyys. Monet totesivat testin jälkeen kuitenkin, että ohjelma on parannusten jälkeen varmasti miellyttävä käyttää.

Käyttäjien mielestä ohjaaminen oli nopeasti opittavissa, eikä häirinnyt käyttöä. Kokeneilla käyttäjillä ei ollut minkäänlaisia hahmotusongelmia ja aloittelevatkin käyttäjät oppivat nopeasti ymmärtämään suuntia, syvyyksiä ja kappaleiden suhteita. Testihenkilöt, jotka olivat aiemmin käyttäneet 3D-mallinnusohjelmia, olivat innoissaan ja olivat täysin sitä mieltä, että ohjelmalla tarkkuuden ja työkalujen lisääntyessä voi tehdä hyvää 3D-grafiikkaa. Ryhmän graafikot innostuivat ajatuksesta, että veisto-ohjelmaa voisi käyttää esimerkiksi vielä luonnollisemmin datahanskalla ja Phantomilla.



Kuva 5: Ohjelmalla tehty karhunpää eri suunnista

5.4. Yhteenveto

Testit kertoivat hyvin paljon prototyypin nykyisestä tilasta. Tärkein anti koko testillä oli se, että näimme todellisia käyttäjiä käyttämässä ohjelmaa. Testi ei osoittanut tiedostettujen ongelmien lisäksi juuri muita epäkohtia, mutta käyttäjiltä saadut ajatukset jatkokehityksestä antoivat paljon lisäarvoa testille. Prototyyppiä kehitetään eteenpäin ottaen huomioon tässä testissä esiin tulleet ajatukset.

6. Lopuksi

Tässä tutkimuksessa on tutkittu Tampereen yliopiston tietojenkäsittelytieteiden laitoksella tehtyä ja kehityksen alla olevaa 3D-veisto-ohjelmaprototyyppiä. Kyseinen sovellus perustuu kaksikäisyyteen. Ohjelmassa on noudatettu suorakäyttöisyyden perusteita. Käyttäjän ei tarvitse hakea toimintoja valikoista, vaan kaikki tieto on esitetty selvästi ja näkyvästi. Kolmiulotteisuudesta johtuen ohjelman ymmärtäminen ei ole välttämättä aivan yksinkertaista aloittelevalle käyttäjälle. Kolmiulotteisuus on abstraktissa ympäristössä aluksi vaikeaa. Kokeneelle 3D-mallintajalle 2D-tasolla tapahtuvan kappaleen hahmottamisessa ei kuitenkaan ole minkäänlaisia ongelmia.

Testeissä käytettävä prototyyppi on vielä keskeneräinen ohjelma, mutta sillä pystyy hyvin tutkimaan, miten käyttäjät hahmottavat kolmiulotteisuutta. Ohjelmaan on lisätty hahmottamista helpottavia työkaluja, joita tässä testissä on tutkittu. Oletamus siitä, että työnjälki on parempi, kun käyttäjällä on apunaan hahmotustyökaluja, piti täysin paikkansa. Työkalut ovat osin kuitenkin vielä vajavaisia, eikä aivan täysin kaikkea niistä saatavaa hyötyä ole vielä saatu irti.

Testit antoivat valtavasti tietoa prototyypin jatkokehitykseen, mitä voi pitää testien ehdottomasti tärkeimpänä antina. Tulevaisuudessa prototyyppiin voitaisiin ajatella lisättävän tuntopalaute ja mahdollisesti uusia ohjaustapoja. Kaikista ihanteellisista tapoista toteuttaa 3D-veisto-ohjelma olisi sellainen, joka kuvaisi mahdollisimman realistisesti oikeata veistämistä. Vain tulevaisuus voi näyttää, millainen tuo tapa tulee olemaan.

Lähteet

- [3Dconnexion, 2003] 3Dconnexion Logitech company, SpaceMouse, <http://www.3dconnexion.com/spacemouseplus.htm>, 2003, Tarkistettu 13.12.2003.
- [Amento *et al.*, 1996] Brian Amento, Patrick Brooks, Hope Harley, and Mike McGee, Fitt's Law, 1996, <http://ei.cs.vt.edu/~cs5724/g1/expfitts.html>, Tarkistettu 13.12.2003.
- [Buxton and Myers, 1986] William Buxton and Brad A. Myers, A study in two-handed input. *Human Factors in Computer Systems, CHI '86 Conference Proceedings*, ACM Press, 1986, 321-326.
- [Carrol *et al.*, 1988] John M. Carrol, Robert L. Mack, and Wendy A. Kellogg, Interface metaphors and user interface design. In Helander, M. (Ed.), *Handbook of Human-Computer Interaction*, Elsevier Science Publishers, 1988, 283-307.
- [Fitts, 1954] P. M. Fitts, The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47, 1954, 381-391.
- [Galyean and Hughes, 1991] Tinsley A. Galyean and John F. Hughes, Sculpting: an interactive volumetric modeling technique. *Computer Graphics*, 25 (4), 1991, 267-274.
- [Guiard, 1987] Asymmetric division of labor in human skilled bimanual action: the kinematic chain model. *Journal of Motor behaviour*, 19 (7), 486-517.
- [Haber and Hershenson, 1973] R. N. Haber and M. Hershenson, *The psychology of visual perception*. Holt, Rinehart and Winston, New York, 1973.
- [Hinckley *et al.*, 1994] Ken Hinckley, Randy Pausch, Goble, and Kassell, A survey of design issues in spatial input. *Proceedings of the 7th annual ACM symposium on User Interface Software and Technology*. ACM Press, 1994, 213-222.
- [Leganchuk *et al.*] Andrea Leganchuk, Shumin Zhai, and William Buxton, Manual and cognitive benefits of two-handed input: an experimental study, *ACM Transactions on Computer-Human Interaction*, 15 (4), 1998, 326-359.

- [MacKenzie and Buxton, 1992] I. Scott MacKenzie and William Buxton, extending Fitts' law to two-dimensional tasks. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, 1992, 219-226.
- [Nielsen and Molich, 1990] Jacob Nielsen and Rolf Molich, Heuristic evaluation of user interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Empowering People*, 1990, 249-256.
- [Nielsen and Olsen, 1987] Gregory M. Nielsen and Dan R. Olsen Jr., Direct manipulation techniques for 3D objects using 2D locator devices. *Proceedings of the 1986 Workshop on Interactive 3D Graphics*, ACM Press, 1987, 175-182.
- [Parent, 1977] Richard E. Parent, A system for sculpting 3-D data. *Computer Graphics*, 11 (2), 1977, 138-147.
- [Parviainen, 2004] Jyrki Parviainen, Suorakäyttöisyys kolmiulotteisessa veisto-ohjelmassa, Raportti B-2004-x, Tietojenkäsittelytieteiden laitos, Tampereen yliopisto, 2004.
- [Sensable, 2003a] Sensable Technologies, Phantom haptic interface, http://www.sensable.com/products/phantom_ghost/papers.asp, 2003, Tarkistettu 13.12.2003.
- [Sensable, 2003b] Sensable Technologies, FreeForm Modeling Plus system, <http://www.sensable.com/products/3ddesign/freeform/index.asp>, 2003, Tarkistettu 13.12.2003.
- [Shneiderman, 1982] Ben Shneiderman, The future of interactive systems and the emergence of direct manipulation. *Behaviour and Information Technology* 1, 1982, 237-256.

Relaatiotietokantojen takaisinmallintaminen

Ari Seppi

Tiivistelmä

Tässä tutkimuksessa olen kartoittanut tietokantojen takaisinmallintamisesta kertovia viime aikaisia ja vähän vanhempia artikkeleita. Otin mukaan sekä yleisellä tasolla liikkuvia artikkeleita että sellaisia, joissa esitetään käytäntöjä ja menetelmiä. Havaitsin, että menetelmissä on osittain selviä yhtäläisyyksiä, esimerkiksi oliomallin hyväksikäyttö laajemmissa mallinnuskäytännöissä.

Avainsanat ja -sanonnat: Relatiotietokannat, takaisinmallintaminen, tietokantakaaviot.

CR-luokat: H 2.1, H 2.4

1. Johdanto

Takaisinmallinnus (reverse engineering) eli valmiin ohjelmatuotteen toimintatapojen selvittäminen, on erittäin tärkeää ohjelmistokehityksessä. Usein ohjelmistoja tuotettaessa ei aloiteta tyhjästä, vaan pohjalla on entinen järjestelmä.

Entistä järjestelmää muokataan muuttuneita tarpeita vastaavaksi niin kauan, kun se on kannattavaa; muokattu järjestelmä ei yleensä kuitenkaan ole ohjelmistoteknisesti niin hyvä kuin aikaisempi. Ajan myötä muokkaukset näin kuluttavat järjestelmän muokauspotentiaalia ja lopulta muokkausten tekeminen vanhaan ohjelmaan käy hankalammaksi ja sitä myötä myös liian kalliiksi. Tarvitaan siis kokonaan uusi ohjelmisto, ja sen kehittämiseen takaisinmallinnustekniikoita.

Takaisinmallinnusta tarvitaan, koska entiseen järjestelmään on saattanut sitoutua suuri määrä yritykselle tärkeää hiljaista tietoa. Esimerkiksi ohjelma saattaa toimia yrityksen sisäisten, tarkemmin määrittelemättömien, käytäntöjen mukaan. Tällainen tieto on tietenkin saatava mahdollisimman täydellisesti talteen, ja käyttöön uudessa järjestelmässä, koska sen hukkaaminen aiheuttaisi suuria tappioita.

Entisen järjestelmän dokumentointi saattaa olla puutteellinen tai puuttua kokonaan. Tämä voi johtua järjestelmää rakentaneiden ohjelmoijien ja suunnittelijoiden ajanpuutteesta tai kiireellisistä muutoksista järjestelmässä. Jos muutoksia on tehty ohjelmointivaiheen alkamisen jälkeen on erityisen suuri mahdollisuus, että muutokset on jätetty dokumentoimatta.

Erityisesti vanhempia järjestelmiä tehtäessä on voitu käyttää epäselviä optimointeja. Aikana, jona koneteho ja muisti olivat erittäin arvokkaita, on koodin selvyyttä uhrattu tehon vuoksi.

Takaisinmallinnusvaiheessa yritetään selvittää, mitä ohjelmat oikein tekevät. Tämä on hankalaa, jos sitä joudutaan puutteellisen dokumentoinnin vuoksi tekemään suureksi osin koodin pohjalta. Hankaluudet korostuvat, jos ohjelma on suuri ja ohjelmakoodia paljon.

Tietokannat toimivat tietovarastona, joten niiden kohdalla tarkoituksen selvittäminen ei ole yhtä hankalaa kuin ohjelmissa. Johtuen tietovarastoluonteesta tietokannat ovat yleensä ohjelmia loogisemmin rakennettuja. Ohjelmointikielet ovat ilmaisuvoimaisia ja liikkuvat niin yksinkertaisella tasolla, että koodin ymmärtäminen on toisinaan vaikeaa.

Yleensä saatavilla olevasta tietokannankannan kuvauksesta pääsee selville yleisrakenteesta, ja tietomassoja taas voi analysoida tehokkaasti koneellisesti. Kaikki ei kuitenkaan ole suoraviivaista: tietoalkioiden tarkka merkitys voi helposti jäädä epäselväksi.

Työssäni tulen keskittymään relaatiotietokantoihin, koska ne ovat yleisimmin käytetty tietokantatyyppejä. Relaatiotietokantojen takaisinmallinnuksesta on kirjoitettu kattavasti artikkeleja, erityisesti belgialaiset Henrard ja Hainaut ovat osallistuneet moniin artikkeleihin. Huomion erityisesti oliomallinnukseen liittyvät menetelmät, joita löytyy Premerlanin ja Blahan sekä Ramalathanin ja Hodgesin kirjoituksista. Tarkoitukseni on siis kartoittaa takaisinmallinnuksen yleistä teoriaa ja sen menetelmiä.

Tulen tässä ensimmäisessä luvussa ensin antamaan yleistietoa takaisinmallinnuksen ongelma-alueesta, minkä jälkeen listaan kolmannessa luvussa käyttökelpoisia tiedonlähteitä. Neljännessä luvussa käyn läpi tietokannan takaisinmallinnuksen vaiheita ja viidennessä luvussa suunnittelun yleisiä käytäntöjä siltä osin kuin se on takaisinmallintamisen kannalta olennaista. Kuudennessa luvussa esittelen pienempiin yksityiskohtiin keskittyviä takaisinmallinnusratkaisuja, ja seitsemännessä kaksi laajempien takaisinmallinnuskäytäntöjen lajittelua. Lopuksi on vuorossa muutaman tällaisen laajemman käytännön esittely.

2. Takaisinmallinnuksen sudenkuopat

2.1. Optimointeja ja rajoitteita

Vaikka ohjelmakoodista löytyvien rakenteiden kaltaisia optimointeja tietokannasta yhtä hankalina löydykään, tiedon tulkintaa vaikeuttavia

rakenteita on voitu sisällyttää tietokantaankin. Kannassa voi esimerkiksi olla tietoalkio, jonka sisällön perusteella pitää tulkita jonkin toisen tietoalkion sisältöä (olemassaoloriippuvuus). Tällainen tapaus voisi olla, jos yksi alkio on henkilön palkka, mutta palkkaan lasketaan ohjelmakoodissa lisäbonuksia työtehtävän mukaan. Dokumentointi onkin erityisen tärkeää tietokannan kanssa, koska tehtävä ohjelma rakentuu tietokannan päälle.

Kaikki tietokantajärjestelmät eivät myöskään näytä suoraan erilaisia rajoitteita. Rajoite on nimensä mukaisesti tietokannan sisältöä jollain tavalla rajoittava ehto; tarkkaa virallista määrittelyä asiasta ei ole, mutta esimerkiksi Elmasrin ja Navathen [Elmasri and Navathe., 2000] mukaan kaikkein yksinkertaisimmat rajoitteet määrittelevät yksittäisen tietokantakentän tyyppin (esimerkiksi numero- tai kirjainmerkkiaineistoa) sekä mahdollisesti pituuden. Osa läpikäydyistä artikkeleista jättää ne huomiotta; ehkä siitä syystä, että ne saadaan yleensä aina helposti selvitettyä, eivätkä ne sen vuoksi ole kiinnostavia. Tieto siitä, saako kenttä saada tyhjän arvon (null), saattaa olla hankalampi selvitettävä, vaikka sekin koskee vain yhtä kenttää.

Monimutkaisemmat rajoitteet koskevat yleensä useampia kenttiä ja ovat entistä hankalammin selvitettävissä. Uudemmat relaatiotietokannat tukevat avain(ainutlaatuisuus)- ja pääavainrajoitteita, sekä avainten suhteita toisiinsa, vierasavainsuhteita. Kovin monipuolisia työkaluja suhteiden kuvaamiseen eivät uudemmakaan SQL-pohjaiset relaatiotietokannat tarjoa.

Uudemmat relaatiotietokantajärjestelmät osaavat yleensä kertoa ainutlaatuiset kenttäjoukot, vierasavaimet ja taulujen sisäiset avaimet. Erityisesti taulujen välisiä suhteita ei relaatiomallissa edes voi kuvata kunnolla. Vierasavaimet kertovat jonkinlaisesta suhteesta, mutta nekään eivät tarkenna, millaisesta suhteesta on kyse. Rajoitteita voi päätellä analysoimalla ohjelmallisesti tietokannan sisältöä ja kaaviota sekä hankkimalla tietoja tietokannan kohdealueesta.

2.2. Häiriö ja piilotieto

Takaisinmallinnusmenetelmien tuloksia voidaan arvioida niiden sisältämän häiriön (noise) ja piilotiedon (silence) avulla. Näistä häiriö on yksinkertaisempi tapaus.

Häiriötä syntyy Henrardin ja Hainautin [Henrard and Hainaut, 2001] mukaan, kun menetelmä löytää tietokannasta jotain, mitä siellä ei ole. Häiriön haittavaikutuksena olemattomien asioiden poissulkemisessa kuluu aikaa. Toisaalta piilotieto jää nimensä mukaisesti takaisinmallinnusmenetelmän ulottumattomiin. Tässä tapauksessa haitan laskeminen on hankalampaa.

Jos piiloon jäävä tieto selviää myöhemmin, voidaan sen haitta yksinkertaiseksi laskea työmääräksi, joka vaaditaan tiedon liittämiseksi

kehitettyyn tietomalliin. Tämä työmäärä jo sinällään on tietysti usein suurempi kuin häiriön aiheuttama, koska jonkin olemassaolemattomuuden tarkistaminen on monesti helpompaa kuin alkaa myöhemmin muokkaamaan valmista mallia.

Toisaalta piilotieto ei ehkä selviä ollenkaan, jolloin se aiheuttaa myöhemmin ongelmia järjestelmän toimiessa epätarkasti tai jättäessä jotain huomiotta. Tällaiset ohjelmavirheet voivat aiheuttaa pitkällä aikavälillä tietojärjestelmää käyttävälle taholle suuriakin tappioita.

2.3. Inhimillisen päätöksenteon tarve

Tietokannan takaisinmallintaminen voi olla huomattavan suuri työ. Näin on erityisesti, jos mallinnetaan satoja tauluja sisältäviä tietokantoja. Automatisointi on siis tarpeen. Ongelmana on, että jokainen tietokanta on erilainen, mitään kattavia yleispäteviä säännöstöjä on hankalaa luoda ja vielä vaikeampaa on kehittää ohjelmaa, joka pystyisi itsenäisesti mallintamaan oikein.

Automatisointi ei kuitenkaan ole yksiselitteisen mahdotonta. On prosesseja, joita voidaan automatisoida täysin. Henrard, Hainaut, Hick, Roland ja Englebert [Henrard *et al.*, 1999] esittävät tällaisesta tilanteesta esimerkiksi vierasavainsuhteiden tunnistamisen tiettyjen sääntöjen perusteella (vierasavaimen toinen puoli on pääavain ja toisella puolella on samannimiset ja -tyyppiset kentät). Osittain voidaan automatisoida esimerkiksi kenttäyhdisteiden havaitseminen. Kolmanneksi voidaan rakentaa työkaluja, jotka osaavat automaattisesti luoda mallintajan avuksi erilaisia raportteja tietokannasta.

3. Takaisinmallinnuksen tiedonlähteet

Tietokannan merkityksiä voidaan kannan kaavion lisäksi selvittää tietokannan sisällön ja sitä käyttävän ohjelmakoodin perusteella. Näistä kaaviontulkitseminen on kaikkein nopein ja varmin tapa, ja tietokannan sisällönkin käyttäminen on koodin tulkitsemistä helpompaa. Henrard, Hainaut, Hick, Roland ja Englebert [Henrard *et al.*, 1999] tukevat tätä näkemystä neljällä perustelulla:

Ensinnäkin välimatka käsitteellisen mallin ja ohjelmiston käytännön toteutuksen välillä on tietorakenteissa pienempi kuin toiminnallisissa osuuksissa. Toiseksi tietorakenteet ovat yleisesti ottaen sovelluksen vakain, hitaimmin muuttuva, osa. Kolmanneksi tietorakenteet ovat toiminnoista erillään jopa hyvin vanhoissa sovelluksissa. Neljänneksi ohjelmakoodin tulkitseminen on paljon helpompaa tietorakenteiden selvittämisen jälkeen.

Kaikkea tietoa ei kuitenkaan kaaviosta löydy. Koska relaatiomalli ei tarjoa suurta ilmaisuvoimaa esimerkiksi eri olioiden (taulujen) suhteista, tarvitaan

avuksi ohjelmakoodia. Tieto siitä, miten tauluja tulisi käyttää, ja miten ne toimivat yhteen löytyy usein upotettuna ohjelman käyttämiin SQL-lauseisiin.

Tämä koodin tulkitseminen on hidasta, mutta usein ainoa tapa selvittää rakenteiden merkityksiä, kun dokumentointi on kehnoa tai puutteellista. Varsinaisten upotettujen SQL-lauseiden analysoimisen lisäksi voidaan tutkia sitä, mitä ohjelmakoodi tekee tietokannasta haetuille lauseille, ja kuinka kantaan tallennettavaa tietoa käsitellään. Tällainen tulkinta on kuitenkin hankala ja aikaavievää, erityisesti jos koodi ei ole rakennettu hyvien suunnitteluperiaatteiden mukaisesti.

Muiksi takaisinmallinnuksen tiedonlähteiksi Henrard, Hainaut, Hick, Roland ja Englebert [Henrard *et al.*, 1999] listaavat ohjelman tulosteet, tietokantajärjestelmän ominaisuudet, olemassaolevan dokumentaation (tämä on tietenkin kaikkein paras tiedonlähde, jos se on ajan tasalla), käyttäjien ja ohjelmistonkehittäjien haastattelut sekä mahdolliset tiedot käyttöympäristöstä. Tässä kartoituksessa keskityn tietokantalähtöisiin tiedonlähteisiin, eli tietokannan kuvaukseen ja sisältöön sekä jonkin verran myös tietokantaa käyttävän ohjelman ohjelmakoodiin.

4. Takaisinmallinnuksen vaiheet

Jean Henrard, Jean-Marc Hick, Philippe Thiran ja Jean-Luc Hainaut [Henrard *et al.*, 2002] huomauttavat, että yleensä harkitaan kahta erilaista järjestelmänmuuntostrategiaa ja sitä myötä myös kahta erilaista takaisinmallinnusstrategiaa. Joko muutetaan koko järjestelmä kerralla tai pieniä askeleita ottaen.

Jens H. Jahnke, Wilhelm Schäfer ja Albert Zündorf [Jahnke *et al.*, 1997] jakavat tietokannan takaisinmallinnuksen kolmeen osan: tietokantakaavion takaisinmallinnus, tietokannan sisällön takaisinmallinnus ja tietokantaa käyttävien ohjelmien takaisinmallinnus. Henrard, Hick, Thiran ja Hainaut [Henrard *et al.*, 2002] tekevät samanlaisen jaon. Olen keskittynyt ensimmäisestä kertoviin artikkeleihin.

Ensimmäisen vaiheen Henrard, Hainaut, Hick, Roland ja Englebert [Henrard *et al.*, 1999] jakoivat kahteen osaan: tietorakenteen selvittäminen ja tietorakenteen käsitteellistäminen. Myöhemmin Henrard ja Hainaut [Henrard and Hainaut, 2001] ottivat mukaan valmisteluvaiheen edeltämään kahta varsinaiseen mallintamiseen kuuluvaa vaihetta. Valmisteluvaiheessa kerätään ja arvioidaan tarvittavia tietolähteitä ja tutustutetaan mallintaja kohdealueeseen haastattelujen ja muiden esitelmien avulla. Jälkimmäiset osat keskittyvät kahden erilaisen kaavion kehittämiseen.

Tietorakennetta selvittäessä kootaan looginen kaavio, johon kuuluvat kaikenlaiset rakenteet ja rajoitteet. Henrard, Hainaut, Hick, Roland ja Englebert [Henrard *et al.*, 1999] jakavat tämän vaiheen neljään alivaiheeseen. Ensimmäiseksi analysoidaan tiedonkuvauskieli (Data Description Language). Jäsentämällä kuvaus saadaan aikaan raaka, fyysinen kaavio. Toisessa vaiheessa yhdistetään fyysiset kaaviot, jos niitä on ensimmäisessä vaiheessa tehty useampi kuin yksi.

Kaavion jalostusvaiheessa etsitään piiloon tai kokonaan toteuttamatta jääneitä rakenteita. Ensiksi kehitetään hypoteeseja tietokannan piilorajoitteista, sitten käytetään muita tiedonlähteitä varmistamaan hypoteesirajoitteiden oikeellisuus. Hypoteesi joko muokataan, hylätään tai hyväksytään, minkä jälkeen se liitetään kaavioon.

Neljänneksi poistetaan kaaviosta kaikki tekniset optimoinnit, kuten indeksit ja rypäsrakenteet. Tämän jälkeen tietorakenteen selvittäminen päättyy, eikä tuloskaavio ole enää tietokantajärjestelmän kanssa yhteensopiva kahdesta syystä [Henrard *et al.*, 1999]: ensinnäkin kaavio saattaa sisältää useamman kuin yhden tietokannan ja nämä tietokannat voivat olla useammasta erilaisesta järjestelmästä. Toiseksi mukaan on otettu rakenteita, joita tietokanta järjestelmä ei luultavasti tue.

Kolmas päävaihe on siis tietorakenteen käsitteellistäminen. Tämä tarkoittaa ei-käsitteellisten rakenteiden muokkaamista tai poistamista, toistamisen vähentämistä ja tietokantajärjestelmälle ominaisten sekä muiden teknisten optimointien tulkitsemista. Lopulta myös tietokantajärjestelmän rakenteet korvataan valitulla käsitteellisellä mallilla.

5. Tietokantojen suunnittelun yleisemmät käytännöt

Halutessa selvittää tietokannan merkityksiä, eräs hyvä lähtökohta on aloittaa takaisinmallintaminen tietokannan alkuperäisestä mallinnuksesta. On siis pohdittava, miten tietokanta luotiin, ja millaisia kohteita yleensä mallinnetaan tietyiksi tietokannan rakenteiksi.

William J. Premerlani ja Michael R. Blaha [Premerlani and Blaha, 1994] listaavat takaisinmallinnuksen syylajeiksi paradigman vaihtamisen (esimerkiksi siirtyminen relaatiotietokannasta oliopohjaiseen tietokantaan), paradigman esiintymän vaihtamisen (esimerkiksi siirtyminen relaatiotietokantajärjestelmästä toiseen) ja yleisen halun selvittää tietokannan tietojen merkityksiä.

He huomauttavat myös, että käytännössä tietokantojen suunnittelijat monesti rikkovat hyvän suunnittelun periaatteita. Näin ollen tarvitaan myös

sellaisia takaisinmallinnustekniikoita, jotka eivät vaadi mallinnettavan tietokannan olevan esimerkiksi kolmannessa normaalimuodossa.

Usein luokat on tietokantaan mallinnettu tauluiksi ja luokkien attribuutit tietokannan sarakkeiksi [Premerlani and Blaha, 1994]. Luokkia voidaan myös yhdistää ja jakaa useammaksi tauluksi. Toisaalta myös yksi-yhteen -suhteet voidaan toisinaan mallintaa tauluksi. Attribuutittomat luokat taas voidaan jättää kokonaan mallintamatta.

Luokkien yleistyksien ja erikoistuksien mallintamiseen on käytössä useita eri tapoja [Premerlani and Blaha, 1994]. Selkein tapa on luoda oma taulu yliluokalle ja sen kaikille aliluokille, missä yliluokalla on sarake, joka määrittää, mihin aliluokaan kukin rivi kuuluu. Toinen tapa on luoda yksi taulu sisällyttämällä eri aliluokkien attribuutit yliluokkaan. Kolmanneksi voidaan luoda aliluokista omat taulut, joissa on kaikissa toistettuna yliluokan attribuutit.

Luokkien välisiä suhteita voidaan relaatiomallissa kuvata vierasavaimilla [Premerlani and Blaha, 1994]. Jos vierasavain viittaa yliluokkaan, sitä on voitu työntää alaspäin aliluokkaan. Näin on voitu saada aikaan samaan tauluun useita vierasavaimia tai useaan tauluun viittaava vierasavain. Jälkimmäiset usein muunnetaan tauluiksi. Suunnittelija voi muuttaa tauluiksi myös ei-n*n-suhteita, jos esimerkiksi suhteella itsellään on attribuutteja.

Premerlani ja Blaha listaavat myös muita tietokantojen takaisinmallinnuskokemuksista peräisin olevia periaatteita. Ensinnäkin he ovat huomanneet, että vaikka isoissa tietokannoissa taulujen iso määrä hankaloittaa kaikkien niiden läpikäymistä, saa suuruus myös suunnittelijat käyttämään samanlaisia rakenteita. Toisaalta se, että kahden kentän pituus eroaa toisistaan, ei tarkoita, että ne tarkoittaisivat eri asiaa (estäen vierasavainsuhteet). Pituus voi olla merkki siitä, että toisessa taulussa kentän perään on lisätty jotain vakiotäytettä.

Erityisen hankalaa on tekijöiden mukaan huomata tietokantaan itseensä piilotettu metadata. Tällaista on esimerkiksi sarakkeiden nimet, jonkin taulun rivien arvoina. Samoin hankalaa on huomata, jos yksi kenttä sisältää useampaa tietoa (komposiitti).

6. Algoritmillisia ratkaisuja ja keinoja riippuvuuksien löytämiseksi

Tietokantojen kenttien – sekä samassa taulussa olevien että eri taulujen kenttien – välillä voi olla erilaisia riippuvuuksia. Näitä ovat vierasavainsuhteet, toistokentät ja olemassaoloriippuvuudet. Toistokenttä tarkoittaa kenttää, jonka tieto voidaan johtaa muista tietokentistä. Olemassaoloriippuvuus puolestaan,

että kentän arvo riippuu jonkin toisen kentän arvosta. Ensimmäisessä alaluvussa käsittelemme Flachin ja Savnikin koneoppimisnäkökulmaa, toisessa taas Limin ja Harrisonin kehittämää algoritmia riippuvuuksien löytämiseksi. Kolmanneksi on vuorossa Soutoun SQL-kyselyitä käyttävä menetelmä. Lopuksi neljäs luku esittelee Henrardin ja Hainautin listaamia menetelmiä lähdekoodin käyttämiseksi.

6.1. Koneoppimisen näkökulma

Peter A. Flach ja Iztok Sarnik [Flach and Sarnik, 1999] ovat tarkastelleet erilaisia rajoitustenetsintäalgoritmeja logiikkaohjelmointikielten ja koneoppimisen näkökulmasta. Lähestymistavassa logiikkaohjelmoinnin sanastolla ilmaistuna sääntöjä (riippuvuudet) etsitään faktoista (tietokantatauluista). Flach ja Sarnik kokeilivat kaikkiaan kolmea induktioperustaista (tietokannan sisällöstä sääntöjä päättelevää) menetelmää.

Ensimmäinen kokeiltu algoritmi oli yksinkertainen, yleisestä-erityiseen (top-down)-pohjainen: käydään kaikki attribuuttiparit läpi tietokannan sisältöä tutkien, ja tarkistetaan löytyykö riippuvuutta. Ellei löydy, riippumatonta osuutta kasvatetaan.

Toisessa menetelmässä, jota Flach ja Sarnik kutsuvat kaksisuuntaiseksi induktioalgoritmiksi, on käytetään sekä yleisestä-erityiseen- että erityisestä-yleiseen (bottom-up) -lähestymistapoja. Ensin sovelletaan yleisestä-erityiseen-menetelmää keräämään lista niistä riippuvuuksista, jotka aineisto todistaa vääräksi. Nämä riippuvuudet muodostavat negatiivisen katteen (negative cover). Sen jälkeen käytetään ensimmäisen menetelmän yleisestä-erityiseen-tapaa riippuvuuksien etsimiseksi, mutta tällä kertaa tutkitaan vain niitä riippuvuuksia, joita negatiivisesta katteesta ei löydy.

Kolmannessa algoritmi on puhdas erityisestä-yleiseen-menetelmä. Siinä luodaan negatiivinen kate, kuten edellä. Tässä tapauksessa se vain käydään läpi kerran tutkien vähiten yleisiä rikottuja riippuvuuksia.

Eryyisesti ensimmäisen menetelmän vaatima prosessointi kasvaa tietenkin suuresti attribuuttien määrän kasvaessa. Flach ja Sarnik totesivatkin menetelmän liian hitaaksi. Toinen algoritmi on jonkin verran parempi, mutta kolmas on selkeästi tehokkain. Flach ja Sarnik eivät kuitenkaan huomioineet Limin ja Harrisonin pari vuotta aikaisempaa tutkimusta, jossa he olivat kehittäneet optimointija yleisestä-erityiseen-mallille.

6.2. Lim-Harrison -algoritmi

Wie Ming Lim ja John Harrison [Lim and Harrison, 1997] ovat jalostaneet olemassaolevia algoritmeja saadakseen aikaan tehokkaan algoritmin

toiminnallisten riippuvuuksien löytämiseksi tietokannasta. Lisäksi he ovat kehittäneet tehokasta tapaa löytää virheet tietokannan sisällöstä.

Aiemmat tällaiset algoritmit ovat yrittäneet luoda pienen määrän hypoteeseja riippuvuuksista ja tarkistaneet niitä sitten tietokantaa vastaan. Näiden menetelmien ongelmana on tehottomuus suurissa tietokannoissa. Menetelmät myös olettavat tietokannan tiedon olevan virheetöntä, mikä on tosielämässä harvinainen tilanne mittaus virheiden tiedon vääränlaisen koodauksen vuoksi.

Tekijät jakavat algoritmit kahteen ryhmään yleisestä-erityiseen- ja erityisestä-yleiseen-algoritmeihin. Ensimmäisessä etsitään ensin mahdollisia riippuvuuksia ja sitten tarkistetaan niiden paikkansapitäminen tietokannan sisällön perusteella.

Ekspontiaalisesti kasvavaa (Flachin ja Savnikin mukaan yleisestä-erityiseen-menetelmän tehottomaksi tekevää) sarakeryhdistelmien määrää kaventamaan on kehitetty muutamia yksinkertaistuksia. Jos pienemmän avaimen on jo huomattu aiheuttavan riippuvuuden, ei tarvitse tarkistaa suurempia avaimia, jotka sisältäisivät kaikki pienemmän avaimen attribuutit. Mahdollisen attribuuttiyhdistelmän kokoa voidaan myös pienentää tai hankkia lisätietoa tietokannan kohdealueesta. Hypoteesien vahvistaminen voidaan tehdä pareissa, mutta tehokkaampi tarkistus saadaan jos taulu ensin järjestetään riippumattomien muuttujien mukaan.

Erytisestä-yleiseen-menetelmät puolestaan tutkivat Limin ja Harrisonin määrittelemän mukaan pelkkää tietokannan sisältöä ottaen kohteeksi kaksi sarakeryhmää kerralla ja tutkien aiheuttavatko tietyt arvot ryhmässä A tiettyjä arvoja ryhmässä B. Tällainen on tietenkin laskennallisesti erittäin raskasta.

6.3. Moniasteisten suhteiden etsiminen

Christian Soutou [Soutou, 1996] on kehittänyt menetelmän moniasteisten suhteiden löytämiseksi käytössä olevasta tietokannasta. Menetelmän olettaa lähtövaatimuksenaan, että avainattribuuttien nimissä ei esiinny homonymiaa, muttei vaadi rajoitteita etukäteen tiedetyksi.

Soutoun menetelmä käyttää aineistona otosta tietokannan sisällöstä ja uuttaa niistä havaitsemilleen suhteille asteet. Tämän menetelmän ongelmana on, että tietokannan suunnittelija on voinut aikoa sallia N:M-suhteet, mutta senhetkinen tietokannan sisältö noudattaa vain N:1 tai 1:1 -suhdetta.

Menetelmän ensimmäinen vaihe on noutaa kannasta tietokantataulujen kuvaukset. Toisessa vaiheessa suoritetaan varsinainen suhteasteiden tunnistaminen ja kolmannessa vaiheessa annetaan tietokantakuvausten perusteella nimet suhteeseen osallistuville kentille.

Näistä toinen vaihe on takaisinmallinnusnäkökulmasta kiinnostavin. Monista muista menetelmistä poiketen Soutoun menetelmä käyttää tietokantaan suoritettavia SQL-alikyselyitä.

Käytännössä menetelmä käyttää kahdentyyppisiä alikyselyitä: 'on olemassa' -kyselyitä ja 'ei ole olemassa' -kyselyitä. On olemassa -kyselyt tarkistavat onko taulussa olemassa sellaista riviä, jossa tietyn vierasavaimen liittyvän kentän arvon vaihtelisi; ei ole olemassa -kyselyt toimivat tietenkin päinvastoin. On olemassa -kyselyitä tarvitaan kutakin taulua kohtaan niin monta kuin taulussa on pääavaimen kuulumattomia vierasavainattributteja.

Ei ole olemassa -kyselyissä puolestaan tarkistetaan ettei ole olemassa kahta riviä, joissa pääavainattribuutit olisivat samoja. Tämän vuoksi niitä tarvitaan saman verran kuin on pääavainattributteja.

Soutou on havainnut myös ongelmia menetelmässään. Ensinnäkin on olemassa -alikeselyt ja ei ole olemassa -alikeselyt voivat olla ristiriidassa esiintyessään samassa hakulauseessa. Tällöin mitään tuloksia ei tietenkään voida saada, mikä voi johtaa piilotietoon. Tältä voidaan Soutoun mukaan välttyä huomioimalla alikyselyiden asteet ja huolehtimalla siten, ettei samaan attribuuttiin liittyen ole alikyselyitä yhdessä kyselyssä.

Toiseksi on mahdollista, että samassa kyselyssä on kaksi samaa asiaa kokeilevaa on olemassa - ja ei ole olemassa -alikeselyä. Ongelman ratkaisuksi Soutou ehdottaa kyselypuiden muodostamista.

6.4. Menetelmiä ohjelmakoodin tulkitsemiseen

Hennard ja Hainaut [Hennard and Hainaut, 2001] ovat keränneet menetelmiä ohjelmakoodin tulkitsemiseksi, koska heidän mukaansa rajoitukset useimmiten määrittellään juuri ohjelmakoodissa. Nämä menetelmät tutkivat, esiintyykö kahden tai useamman kentän välisiä tarkistuksia.

Ensimmäinen menetelmä on muuttujariippuvuuskaavio (variable dependency graph), joka on helposti laskettava versio tietovirtakaaviosta. Kaaviossa jokainen muuttuja esitetään solmuna, nuoli solmujen välille piirretään suorasta yhteydestä (sijoitus, vertailu tms.) Jos muuttujien välillä on translatiivinen suhde, ne voivat esittää samoja olioita, jakaa samat arvot tai jotain muuta.

Hennard ja Hainaut havaitsivat, että muuttujariippuvuuskaaviota tulkitsemalla, voidaan päätyä kolmen eri lajin piilotietoon ja kahden eri lajin häiriöön. Ensinnäkin, jos otetaan huomioon vain sijoitussuhteet, saattaa riippuvuuksia jäädä huomaamatta. Tästä ongelmasta selviää, ottamalla huomioon monipuolisesti erilaisia vuorovaikutuksia muuttujien välillä (esimerkiksi vertailut). Toinen mahdollinen piilotieto johtuu siitä, että ohjelmointikielen rakenne on liian epäselvä, että tulkintaa voitaisi tehdä.

Kolmas piilotiedon syntymisvaihtoehto on ohjausrakenteiden huomiottajättäminen. Ohjelmakoodissa voi määriteltynä, että muuttujan a ollessa arvoltaan jokin tietty, muuttujaa b muutetaan. Tällainen riippuvuus jää huomaamatta mentäessä läpi pelkkiä yksittäisiä lauseita.

Häiriötä puolestaan voi syntyä, jos tiettyä muuttujaa käytetään useammassa erillisessä prosessissa vertailemaan useampia erillisiä tietokantamuuttujia [Henrard and Hainaut, 2001]. Tällöin muuttujakaavio voi alkaa seuraamaan myös tiettyyn muuttujaan vaikuttamattomia muuttujia. Toiseksi johonkin tietokannasta aluksi poimittuun muuttujaan, on voitu myöhemmin sijoittaa jokin täysin toinen arvo. Tämän arvon sijoittaminen johonkin toiseen kenttään ei luo riippuvuutta näiden kenttien välille.

Ohjelman siivutuksessa, ohjelmakoodista poimitaan ohjelmasta O kaikki ne kohdat, jotka voivat tietyssä kohdassa vaikuttaa tietyn muuttujan arvoon. Henrard ja Hainaut huomauttavat, että Horwitz et al. kehittivät tästä eteenpäin järjestelmäriippuvuuskaavion, joka ottaa huomioon myös toimenpidesarjat. Henrard ja Hainaut kehittävät kaaviota vielä eteenpäin lisäämällä siihen satunnaisen ohjausvirran. Kaaviossa eri koodirivit ja sijoitukset ovat solmuja, joiden välillä kulkee siis ohjaus- ja vaikutusnuolia.

Ohjelman siivuttaminen lasketaan kulkemalla kaaviossa kahdessa vaiheessa [Henrard and Hainaut, 2001]. Ensin tutkitaan, mitkä ohjelman O sisäiset solmut tavoittavat solmun s , toisessa vaiheessa tutkitaan, mitkä ohjelman O kutsumat toiminnot ja ohjelmaa O kutsuvien toimintojen kutsuvat toiminnot tavoittavat solmun s .

Ohjelman siivuttamista voidaan käyttää kolmella eri kyselytavalla [Henrard and Hainaut, 2001]. Yksinkertaisessa ohjelmasiivutuksessa tutkitaan edeltääkö jonkin kentän tietokantaan kirjoittamista jonkin toisen kentän lukeminen. Tällöinhän kyseessä voisi olla riippuvuus (tarkistetaan sopiiko luettavan kentän arvo kirjoitettavaan). Näin ei kuitenkaan aina ole, jos kentän arvot esimerkiksi syystä tai toisesta pitää lukea ennen kirjoittamista, mutta mitään vertailua ei suoriteta. Menetelmä voi siis aiheuttaa huomattavan paljon häiriötä.

Tietovirtaohjelmasiivutus (dataflow program slicing) keskittyy kaaviossa riippuvuutta kuvaaviin nuoliin, ja jättää ohjausvirtaa kuvaavat nuolet huomiotta [Henrard and Hainaut, 2001]. Tämän vuoksi tekniikka ei kerro kaikkia riippuvuuksia (jättää piilotietoa), muttei myöskään tuota häiriötä.

Tietovirtaohjelmasiivutetekniikkaa voi parantaa seuraamalla muuttujia [Henrard and Hainaut, 2001]. Kun viipaleessa on luku- ja kirjoitusohjeita, voidaan tutkia muuttujan matkaa ohjelmassa ja analysoida siihen liittyviä ohjelmakoodirivejä.

7. Takaisinmallinnusmenetelmien luetteloinnit

Koko takaisinmallinnusprosessin kulkua määritteleviä käytäntöjä on luokiteltu eri tavoin. Aliluvuissa käsittelen kaksi jakoperustetta.

7.1. Pedro-de-Jesuksen ja Sousan jakoperusteet

Pedro-de-Jesus ja Sousa [Pedro-de-Jesus and Sousa, 1999] jakavat eri menetelmät ryhmiin viiden tekijän perusteella. Nämä tekijät ovat vaaditun tiedon muoto, vaaditusta tiedosta tehtävät oletukset, menetelmän tuottama tieto, menetelmän käytännöt ja mitä, uutta menetelmät ovat tuoneet.

Tuo jako toimii mielestäni hyvin mallinnusmenetelmiä opiskelevan tarpeisiin. Toisaalta olisi hyödyllistä jaotella menetelmiä myös keskittyen tarkemmin sopivaa menetelmää etsivän tarpeisiin. Tästä näkökulmasta ei oikeastaan ole olennaista, mitä uutta menetelmässä on ollut, tai edes menetelmän käytännöt, koska menetelmän käytännöntoteutus tuskin muodostaa esteitä, joita saanti- ja ulostulotietojen ja oletusten määrääminen ei jo olisi kattanut.

Pedro-de-Jesus ja Sousa määrittelevät myös kahdeksan ominaisuutta lajittelemaan takaisinmallinnustekniikoita. Näistä kolme ensimmäistä kertovat menetelmän vaatimista esitiedoista ja syötelajeista. *Semanttinen tieto* määrittelee, kuinka paljon mallintaja tietää järjestelmästä. Tällä ominaisuudella on kolme mahdollista arvoa. Joko mallintaja tuntee attribuuttien merkityksen, tai ainakin pystyy ratkomaan ristiriitaisuudet (homo- ja synonyymeistä johtuvat) tai sitten mallintaja ei tunne järjestelmää ollenkaan. Toinen ominaisuus liittyy *tietokannan sisältöön*: käyttääkö menetelmä sitä vai ei. Kolmas ominaisuus kertoo *ohjelmakoodin hyödyntämisestä*: koodia, joko käytetään tai ei, ja jos koodia käytetään menetelmä voi olettaa, että siinä voi olla virheitä, tai että koodi on virheetöntä.

Neljä seuraavaa määrittävät, minkälaisia oletuksia tietokannan rakenteesta tehdään. *Avainriippuvuusominaisuus* kertoo, olettaako menetelmä, että taulujen avaimet ja erityisesti pääavain ovat tiedossa.

Vierasavainriippuvaisuusominaisuus taas kertoo tiedetäänkö vierasavainsuhteita. *Ei-avainperustaiset toiminnalliset riippuvaisuudet tai kolmas normaalimuoto* -ominaisuus tarkoittaa, että tietokannan tarvitsee olla kolmannessa normaalimuodossa tai avaimiin perustumattomat riippuvuudet ovat tiedossa. Toisaalta *ei-avainpohjaiset sisällymisriippuvuudet* -ominaisuus kertoo oletetaanko vierasavaimiin liittymättömät sisällymisriippuvuudet tiedetyiksi. Tämä siis tarkoittaa esimerkiksi yli- ja aliluokan suhdetta. Viimeisenä ominaisuutena määritellään *käyttäjän vuorovaikutuksen määrä*, eli siis millaisissa tilanteissa käyttäjän vuorovaikutusta tarvitaan.

Luonnollisesti tarkkaa tietoa attribuuttien merkityksistä vaativa järjestelmä ei tarvitse niin paljon muuta tietoa mallinnuskohteesta kuin vähemmän tietoa vaativa. Toisaalta tiukat vaatimukset riippuvuusominaisuuksissa heijastuvat vähempänä käyttäjävuorovaikutuksen tarpeena.

7.2. Henrard-Hick-Thiran-Hainaut -jako

Jean Henrard, Jean-Marc Hick, Philippe Thiran ja Jean-Luc Hainaut [Henrard *et al.*, 2002] kuvailevat kuutta erilaista tapaa suorittaa takaisinmallinnus. Näiden tapojen avulla voidaan myös jakaa takaisinmallinnusmenetelmät kuuteen ryhmään.

Jako ryhmiin tapahtuu kahden ulottuvuuden perusteella. Tietokantaulottuvuus määrittää, miten tarkasti tietoa käsitellään. Mahdollisuutena on jäädä fyysiselle tasolle, jolloin entiset rakenteet mallinnetaan mahdollisimman läheisiksi uuden järjestelmän rakenteiksi. Tällöin ei siis yritetä millään lailla ymmärtää tiedon merkityksiä. Toinen vaihtoehto on pyrkiä käsitteelliselle tasolle, eli merkityksien ymmärtämiseen.

Työkaluja avustamaan prosessissa löytyy molemmille tasoille sijoittuville menetelmille. Fyysisellä tasolla tarvitaan vain yksinkertaisia työkaluja, kuten määrittelykielen tulkkia, yksinkertaista kaaviomuunninta ja määrittelykieligeneraattoria. Käsitteellisellä tasolla tarvitaan monipuolisempia työkaluja, jotka osaavat tulkita luettuja rakenteita.

Toinen ulottuvuus on ohjelman taso. Käyn tasot lyhyesti läpi, vaikka pääasiassa keskitynkin tietokantatason mallinnusmenetelmiin. Ohjelmassa muutosprosessi voidaan suorittaa välikerrosten, hakulauseiden tai logiikan tasolla. Yksinkertaisimmassa välikerrosstrategiassa luodaan ohjelma, joka toimii kuten entinen tietokanta, mutta muuntaa vanhanmalliset tietokantaoperaatiot uusiksi. Tällöin ohjelma siis toimii kuten ennenkin.

Hakulausetasolla muokataan ohjelman käyttämiä hakulauseita vastaamaan uutta tietokantaa. Logiikan tasolla taas muutetaan koko ohjelmaa, niin että se saa täyden hyödyn järjestelmämuutoksista. Sijoittamalla kunkin menetelmän yhdelle sijalle ulottuvuusakseleilla saadaan siis yhteensä kuusi menetelmäjä.

8. Erilaiset tietokantojen takaisinmallinnusmenetelmät

Lopuksi esittelen muutamia käytäntöjä mallintamisprosessin suorittamiseksi. Ensimmäisenä esittelen tavan jäsentää prosessia graafisesti ja kaksi jälkimmäistä ovat tapoja muuntaa relaatorakenteet olioiksi.

8.1. Yleiset sumean päättelyn verkot takaisinmallinnuksen apuna

Tietokantojen takaisinmallintaminen perustuu eri lähteistä hankittaviin tietoihin. Koska nämä oletukset eivät ole varmoja, tarvitaan jokin keino niiden

käsittelyyn ja vertailuun. Jens H. Jahnke, Wilhelm Schäfer ja Albert Zündorf [Jahnke *et al.*, 1997] kehittivät tällaisen menetelmän pohjautuen yleisiin sumean päättelyn verkkoihin (generic fuzzy reasoning nets). Menetelmänsä kohdealueen he rajaavat oman jaottelunsa mukaisesti ajateltuna takaisinmallinnusprosessin ensimmäiseen vaiheeseen eli tietokantakaavion takaisinmallinnukseen.

Erityisenä perusteena sumean päättelyn verkkojen puolesta kirjoittajat käyttävät takaisinmallintamisessa esiin tulevia ristiriitaisia oletuksia, jotka seuraavat huonosta suunnittelusta ja itse ohjelman ristiriitaisuudesta. Käytettäessä verkkoesitysmuotoa ja oletuksien arvoista saadaan mallintajalta kyselemällä tehokkaasti rakennettua järkevin mahdollinen kaaviokokonaisuus.

Kaikkia kannasta irtisaatavia oletuksia ei ole mahdollista ottaa mukaan heti, koska suuri vaihtoehtomäärä johtaisi liian suureen analysoitavaan verkkoon. Sen sijaan aloitetaan pienemmällä määrällä oletuksia ja lisätään niitä mukaan mallinnusprosessin edetessä.

Kirjoittajat kutsuvat oletuksia aksiomiksi (axiom) ja ryhmittelevät ne vahvoihin, heikkoihin ja muokkautuviin aksiomiin. Vahvat aksiomat ovat perusoletuksia, joita ei voida mallinnusprosessin edetessä todistaa vääriksi. Heikot aksiomat ovat mallintajan mukaan tuomia oletuksia. Ne voivat muuttua prosessin aikana ja niitä voidaan myös hylätä. Muokkautuvat aksiomat ovat oletuksia joita ei hyväksytä vahvoiksi, koska vahvojen aksiomien liian suuri määrä vaikeuttaisi mallinnusta. Myös nämä aksiomat muokkaantuvat mallinnusprosessin aikana.

Käsitellessä tällaista epätäydellistä ja muuttuvaa tietoa, tarvitaan vaikutusmekanismi, joka mahdollistaa ei-monotonisen (ihmisen päättelyä matkimaan pyrkivän) päättelyn. Kirjoittajat käyttävät tähän tarkoitukseen Petri-verkkoja niiden rinnakkaisuuden vuoksi. Heidän käyttämänsä algoritmi on seuraavanlainen:

Ensiksi sumea Petri-verkko v_1 luodaan sumean päättelyn verkon v_2 avulla. Jokaista verkon v_2 propositiota kohden luodaan verkkoon v_1 kaksi paikkaa, toinen proposition totuutta ja toinen epätotuutta varten. Ensin luodaan aluksi oletetut heikot ja vahvat aksiomat. Tämän jälkeen tutkitaan onko uusin propositioiden myötä jonkin implikaation esiehdot täyttyneet. Myönteisessä tapauksessa luodaan implikaation seurauspropositiot. Tätä jatketaan, kunnes uusia propositioita ei enää synny.

Toisessa vaiheessa luodaan siirtymiä positiivisten ja negatiivisten paikkojen välille. Implikaation lisääminen johtaa ainakin kahden siirtymän luomiseen sillä $a \Rightarrow b$ johtaa siihen, että myös $\neg a \Rightarrow \neg b$. Yleisesti ottaen siirtymien määrä on sama kuin propositioiden edeltävässä laajennetussa implikaatiossa.

Kolmanneksi vahvoihin ja muokkautuviin aksioomiin johtavat siirtymät poistetaan, koska niiden oletuksia ei saa arvioinnin aikana muuttaa. Lopuksi on arviointivaihe.

8.2. Takaisinmallinnus olioiksi: Premerlani ja Blaha

Tietokantakaavion takaisinmallinnuksen eri vaiheita ovat muun muassa William J. Premerlani ja Michael R. Blaha [Premerlani and Blaha, 1994].

Kirjoittajien mukaan oliopohjaiset mallit ovat luonnollinen työkalu takaisinmallinnustyöhön. Tämä on tosiaan varsin selvä asia: oliot pystyvät kuvaamaan sen, mitä relaatiomallikin ja lisäksi oliomallit tarjoavat monipuolisempia merkintöjä.

Ensimmäinen askel on taulujen muuttaminen luokiksi ja sarakkeiden attribuuteiksi; toiseksi etsitään avainhokkaita. Kolmanneksi ovat vuorossa vierasavaimet. Moderneista tietokantajärjestelmistä vierasavaimet saa suoraan, vanhemmista järjestelmistä avaimet joutuu mallintaja etsimään.

Vierasavainten etsintä tapahtuu ratkomalla ensin homonyymit (attribuutit, joiden nimi on sama, mutta jotka viittaavat eri asioihin) ja synonyymit (attribuutit, joiden nimi on eri, mutta jotka viittaavat samaan asiaan).

Samanlaiset attribuuttinimet, tietotyypit ja tietoaalueet voivat olla merkki vierasavainsuhteesta. Yleistysten vuoksi tässä vaiheessa on vielä kyse vierasavainryhmistä, eikä yksittäisistä avaimista.

Vierasavainten etsimisen jälkeen Premerlani ja Blaha jatkavat jalostamalla alkuperäisiä luokkia. Horisontaalisesti pilkotut luokat, joilla on sama tietomalli pitää yhdistää. Näitä on jaettu useammiksi tauluksi ehkä tietokannan jakamiseksi useammalle palvelimelle tai muiden optimointien vuoksi. Kannattaa myös etsiä rajoitteita, jotka esitetään tauluina. Näistä vihjaa se, ettei tauluun ole yhdistetty vierasavainryhmiä.

Viidentenä vaihteenä on yleistyksien huomaaminen. Kannattaa keskittyä isoihin vierasavainryhmiin ja myös toisen taulun pääavaimen kokonaan sisältävä vierasavain voi olla merkki yleistyksestä. Jos myös aliluokkien attribuutit on nostettu ylliluokkaan, on seurauksena attribuuttiryhmiä, joissa joko kaikissa on jokin arvo, tai sitten kaikki ovat tyhjiä null-arvoja.

Tämän jälkeen voidaan keskittyä olioiden välisiin suhteisiin. Kahdesta vierasavaimesta koostuva luokka on selvä merkki suhteesta. Suhteen voi myös muodostaa olioiden välille, kun avainhokas sisältää vierasavaimen ja ei-vierasavainattribuutteja. Loput suhteet esitetään vierasavaimina. Premerlanin ja Blahan mukaan suhteiden minimi- ja maksimiastelukuja voi olla vaikeaa selvittää pelkän datan avulla, Soutouhan oli huomasi saman asian myöhemmin kirjoittamassaan artikkelissa [Soutou, 1996]. Lopuksi osa luoduista suhteista kannattaa yrittää muuttaa keräytymiksi (aggregation).

Viimeisenä vaiheena on optimointien poistamiseksi suoritettavat muutokset. Luokkia voi ensinnäkin muuttaa linkkiluokiksi (suhteeksi, jonka linkit voivat osallistua suhteisiin muiden luokkien välillä). Kevyet 1:1-suhteet voidaan mallintaa myös attribuuteiksi, ja N-asteen suhteita pitäisi yrittää muuttaa pienemmiksi mikäli mahdollista.

Ongelmana Premerlanin ja Blahan mukaan on, että takaisinmallinnusvaiheessa on hankalaa löytää minimi- ja maksimiasteita. Tässä voisikin olla hyvä tilaisuus käyttää Soutoun aikaisemmin käsiteltyä menetelmää hyväksi. Toisaalta myös transitiivisia suhteita on selkeyden vuoksi hyvä järjestellä. Esimerkiksi taulujen A ja B sekä B ja C välillä voi olla suhde, joka voi olla selkeämpää järjestää suhteeksi luokkien A ja B sekä A ja C välille.

Siistintävaiheessa on myös huomioitava, että tietokannat voivat kuvata samaa suhdetta osoittimilla suhteen molemmissa päissä. Onkin varottava, ettei luoda samaa suhdetta kaavioon useampaan kertaan. Luokkahierarkian selkeyttämiseksi on ehkä luotava uusia luokkia, joita tietokannassa ei mainita mitenkään.

Transitiivisuus pitää huomata myös yleistyksien kautta kulkevana. Ilman semanttista tulkintaa on vaikeaa tunnistaa mikä on varsinainen suhde kahden erillisen luokan välillä, ja mikä on yleistys-erikoistus-suhde. Jos tietokannassa luokalla A vierasavain suhde luokan B kaikkiin aliluokkiin voidaan suhde siirtää koskemaan luokkaa B.

8.3. Takaisinmallinnus olioiksi: Ramanahan ja Hodges

Myös Ramanathan ja Hodges [Ramanathan and Hodges, 1997] kehittivät menetelmän oliorakenteiden löytämiseksi relaatiotietokannasta kolmisen vuotta Premerlania ja Blahaa myöhemmin. Menetelmä vaatii, että mallinnettava relaatiotietokanta on vähintään toisessa normaalimuodossa (se ei siis vaadi kolmatta normaalimuotoa, kuten useat muut). Lisäksi menetelmä vaatii esitietona pää- ja vierasavaimet. Myös homonyymi- ja synonyymiominaisuudet ovat tiukat: samaan arvoalueeseen viittaavat attribuutit oletetaan samoiksi ja homonyymejä ei oleteta olevan.

Ramanathan ja Hodges aloittavat määrittelemällä kiinnostaviksi rakenteiksi luokat ja suhteet. Ensinnäkin on löydettävä ne tietokantataulut, jotka viittaavat olioluokkaan. Toiseksi pitää havaita niiden väliset suhteet; näitä suhteita on kolmea tyyppiä yhteys (association), periytyminen (inheritance) ja keräytymä (aggregation).

Menetelmä aloittaa tutkimalla tiedossa olevia riippuvuuksia ja muuntaen siten kaavion kolmanteen normaalimuotoon. Tällä tavalla muodostetut tietokantataulut mallinnetaan luokiksi, paitsi ne joiden pääavain muodostuu

kokonaan vierasavaimista (ne mallinnetaan suhteiksi). Vierasavainten perusteella luodaan suhteita luokkien välille.

Periytymissuhteiden mallintamisessa Ramanathan ja Hodges ottavat huomioon samat tapaukset, kuin Premerlani ja Blaha: Yliluokka ja aliluokat voivat olla eri tauluja, joissa molemmissa käytetään samaa pääavainta. toisaalta aliluokkien attribuutit on voitu kaikki tuoda yliluokkaan tai yliluokan attribuutit sisällyttää aliluokkiin.

Keräytymien havaitsemiseen Ramanathan ja Hodges ovat löytäneet kaksi tapausta. Ensimmäinen merkki keräytymästä on tapaus, jossa relaation R1 pääavaimessa on enemmän kuin yksi attribuutti eivätkä kaikki noista attribuuteista liity vierasavaimen. Relatio R1 voi olla keräytymäsuhteessa relaatioon R2, jos R2 viittaa eniten relaation R1 pääavaimen vierasavaimiin.

Toisessa tapauksessa käytetään hyväksi relaatiotietokantojen kyvyttömyyttä säilöä lukutaulukoita. Tätä rajoitusta kierretään yleisesti käyttämällä binääritietotyyppiä, jonka tietokantaa käyttävä sovellus muuntaa hakiessaan taulukoksi ja tallentaessaan binäärimuotoiseksi. Koska ohjelmallisesti binääritiedon laadun tulkitseminen ei onnistu on binääritietoa kohdattaessa kysyttävä kentän merkitystä käyttäjältä.

Ramanathanin ja Hodgesin tutkimus tuo siis uusina piirteinä Premerlanin ja Blahan menetelmään verrattuna ohjeita keräytymien havaitsemiseen sekä siihen liittyen huomion, että binäärimuotoinen tieto on hyödyllistä huomioida erityistapauksena. Mitään tekniikkaa binääritietoa varten Ramanathan ja Hodges eivät kuitenkaan kehittäneet (se olisikin ollut erittäin hankalaa), vaan merkityksiä pitää kysellä mallintajalta.

9. Lopuksi

Olen tässä työssä kartoittanut aiempaa tietokantojen takaisinmallintamista koskevaa kirjallisuutta. Mukaan ottamani artikkelit ovat yhtä mieltä monista perusasioista, ja eroja löytyy lähinnä kunkin artikkelin uusista keksinnöistä ja yksityiskohdissa.

Takaisinmallinnuksen kaksi suurta ongelmaa ovat eri kenttien merkitysten ja kenttien välisten suhteiden löytäminen. Menetelmät ja algoritmit vaativat erilaisia ennako-oletuksia ja käyttäjävuorovaikutuskohtia ja antavat erilaisia tuloksia, ja tuottavat lisäksi eri määriä häiriötä ja jättävät huomaamatta piilotietoa. Juuri tämän vuoksi useat eri menetelmät ovat tarpeen.

Erilaiset ratkaisut takaisinmallinnusongelmiin voidaan yleisesti ottaen jakaa kahteen ryhmään. Yhtäältä löytyy erilaisia algoritmeja ja muita yksittäiseen ongelmaosaan keskittyviä menetelmiä ja toisaalta koko prosessin kulkuun päätyneitä mallinnuskäytäntöjä. Näitä pienempiä ratkaisuita voi yhdistää

isompiin kehyksiin ja myös isompia kehysratkaisuita voi jossain tilanteissa yhdistellä. Esimerkiksi Jahnken graafisen esitystavan [Jahnke *et al.*, 1997] sopii käyttöön mallinnettaessa oliokäytäntöjen mukaisesti.

10. Viiteluettelo

- [Elmasri and Navathe., 2000] R. Elmasri and S. Navathe, *Fundamentals of Database Systems 3 ed.*, Addison-Wesley, 2000.
- [Flach and Savnik, 1999] Peter A. Flach and Iztok Savnik, Database dependency discovery: A machine learning approach, *AI Communications*, **12**, (3), 1999, 139-160.
- [Henrard *et al.*, 1999] Jean Henrard, Jean-Luc Hainaut, Jean-Marc Hick, Didier Roland and Vincent Englebert, Data Structure Extraction in Database Reverse Engineering, In: Peter P. Chen, David W. Embley, Jacques Kouloumdjian, Stephen W. Liddle and John F. Roddick (eds.), *Advances in Conceptual Modeling: ER '99 Workshops on Evolution and Change in Data Management, Reverse Engineering in Information Systems, and the World Wide Web and Conceptual Modeling, Paris, France, November 15-18, 1999, Proceedings*, Springer, 1999, 149-160.
- [Henrard and Hainaut, 2001] Jean Henrard and Jean-Luc Hainaut, Data dependency elicitation in database reverse engineering, *5th European Conference on Software Maintenance and Reengineering (CSMR 2001)*, IEEE Computer Society, 2001, 11-19.
- [Henrard *et al.*, 2002] Jean Henrard, Jean-Marc Hick, Philippe Thiran and Jean-Luc Hainaut, Strategies for Data Reengineering, *Ninth Working Conference on Reverse Engineering (WCRE'02)*, IEEE Computer Society Press, 2002, 211-220.
- [Jahnke *et al.*, 1997] Jens Jahnke, Wilhelm Schaefer and Albert Zuendorf, Generic Fuzzy Reasoning Nets as a Basis for Reverse Engineering Relational Database Applications, *Sixth~European Software Engineering Conference (ESEC '97)*, Springer, 1997, 193-210.
- [Lim and Harrison, 1997] Wie Ming Lim and John Harrison, Discovery of Constraints from Data for Information System Reverse Engineering, *Australian Software Engineering Conference, ASWEC '97*, IEEE Computer Society Press, 1997, 39-48.
- [Pedro-de-Jesus and Sousa, 1999] Maria de Lurdes Pedro-de-Jesus and Pedro Manuel Antunes Sousa, Selection of Reverse Engineering Methods for Relational Databases, *Third European Conference on Software Maintenance and Reengineering 1999*, IEEE Computer Society, 1999, 194-197.

- [Premerlani and Blaha, 1994] William J. Premerlani and Michael R. Blaha, An approach for reverse engineering of relational databases, *Communications of the ACM*, **37** (5),1994, 42-49.
- [Ramanathan and Hodges, 1997] Shekar Ramanathan and Julia E. Hodges, Extraction of object-oriented structures from existing relational databases, *SIGMOD Record*, **26** (1), 1997, 59-64.
- [Soutou, 1996] Christian Soutou, Extracting N-ary Relationships Through Database Reverse Engineering, *International Conference on Conceptual Modeling 1996*, Springer, 1996, 392-405.

Katsaus mallipohjaiseen käyttöliittymäkehitykseen

Perttu Sliden

Tiivistelmä.

Käyttöliittymän kuvaaminen abstraktin käyttöliittymän määrittelykielen avulla on hyvin haastavaa, jos mallinnettava sovellus koostuu laajoista määristä ohjelmistokirjastoja, ohjelmistokomponentteja ja näiden välisiä yhteyksiä. Lisäksi ongelmaksi voi muodostua näiden komponenttien väliset yhteydet, jotka eivät välttämättä ole yhteensopivia käytettävän kuvauskielen kanssa. Kuvaamisessa ja mallintamisessa pystytään käyttämään arkkitehtuurisesti abstrakteihin XML-skeemoihin ja erilaisiin XML-dialekteihin perustuvia kuvauskieliä. Määrittelykielen avulla voidaan selvittää muun muassa kohdealustan abstraktien graafisten alkioden väliset liittynät ja lisäksi toteuttaa käyttöliittymämalli, jonka avulla alustalle pystytään mallintamaan rajattuja käyttöliittymäkuvauksia. Uusien ominaisuuksien ja käyttöliittymätiedon lisääminen malliin onnistuu ilman, että järjestelmän ydinrakennetta tarvitsee muuttaa. Mallilla toteutetuilla kuvauksilla pystytään osoittamaan ja toteuttamaan omat itsenäiset ohjelmistokomponentit käytettävän järjestelmän osaksi.

Avainsanat ja -sanonnat: sovellusarkkitehtuuri, määrittelykieli, XML, käyttöliittymämalli.

CR-luokat: D2.2

1. Johdanto

Kaikki käyttöliittymiin liittyvät suunnittelu- ja mallintamismenetelmät vaativat kompromisseja. Erityisesti näitä ilmenee sovelluskehityksessä liittyen käyttöliittymästandardeihin, monialustaisuuteen sekä siirrettävyyteen sovelluksien välillä. Käyttöliittymät ovat tärkeä osa sovellusohjelmia, mitkä sallivat käyttäjänsä vuorovaikutuksen tietokoneen kanssa ja tehtävien suorittamisen tarkoituksenmukaisen käyttöliittymämallin kautta. Tämän tutkielman tarkoituksena on koota yhteen mallipohjaiseen käyttöliittymäkehitykseen liittyvää tutkimustietoa ja esittää käyttöliittymämallin käsitteellinen pohja. Lisäksi luodaan katsaus abstrakteihin käyttöliittymämalleihin ja tutkimuskenttään liittyviin peruskäsitteisiin. Kukin malli ja mallipohjainen suunnittelumetodiikka käsittelevät itsessään tärkeää, olioperustaisessa ohjelmistokehityksessä ehkäpä ongelmallisinta osuutta, konkreettisen käyttöliittymän kuvausta.

Käyttöliittymämalli (engl. UIM - User Interface Model) on deklaratiivinen määrittely käyttöliittymästä, sen ulkoasusta, yhteyksistä erilaisten käyttöliittymäelementtien kesken ja mallin kommunikoinnista sovelluksen functionaliteetin kanssa [Müller, A., *et al.*, 2001]. Malli voidaan määrittellä jollain tätä tarkoitusta varten kehitetyllä abstraktilla liittymänkuvauskielellä korkeammalla abstraktiotasolla. Tällaisella kuvauskielellä rakennettujen käyttöliittymäkuvausten sisältö tulkitaan malliin liittyvässä sovelluslogiikassa, johon osallistuvat esimerkiksi XML-teknologiaan pohjautuvat työkalut, kuten XML-parserit, XML-jäsentimet ja XML-dokumentin tiedon sitomiseen erikoistuneet mekanismit. Yksinkertaistettuna käyttöliittymämallipohjaiseen työkaluun perustuvat mekanismit muuntavat mallin määrittelemät osat ja korkean tason kuvaukset, jonka jälkeen automaattinen sovelluksen käyttöliittymän tuottaminen mallin määrittelemille osille ja halutulle kohdealustalle voi tapahtua.

Abstrakteja käyttöliittymien määrittelyyn, kuvaamiseen ja tuottamiseen liittyviä metakieliä sekä määrittelykieliä on useita ja tunnetuimpia niistä ovat XSLT [W3C, 1999], XUL [XUL, 2003], UIML [Ali M. F., *et al.*, 2001] ja XIML [Puerta, A., Eisenstein, J., 2001]. Erityisesti XUL on saavuttanut laajan suosion sekä kehittäjien että käyttäjien keskuudessa. Suosio johtuu osittain Mozilla-projektin tuotoksena syntyneestä Web-selaimesta, missä itse XUL-moottori on sisäänrakennettuna. Projektissa kehitetystä XUL-mallintamiskielestä ja XUL-moottorista on johdettu useita muunnelmia, jotka muuntavat syötetyn XML-dokumentin rakenteen mukaisen käyttöliittymäkuvauksen. Korhonen [2002] on muun muassa kuvaillut tutkielmassaan UIML-määrityksen mukaisen UIML-moottorin toteutuksen ja sen soveltuvuuden Windows-pohjaisten sovellusten kehitystyöhön.

Yksi suurimmista ongelmista sovellusten kehittämisessä on, miten varmistetaan käyttöliittymän tai jonkin toiminnallisen osan siirrettävyys erilaisille laite- ja ohjelmistoalustoille [Vanderdonck, J., *et al.*, 2001]; [Silva, P.P., 2000]. Käyttöliittymästandardien moninaisuus on yksi este siirrettävyydelle. Standardit määrittelevät tarkasti sovelluksen ulkoasun (*presentaatiomalli*) ja kommunikointimekanismit (*dialogimalli*, *käyttäjämalli*) eri sovelluksen tasoilla, joilla sovellus käy vuoropuhelua tietokoneen ja käyttäjän välillä. Mallin on esimerkiksi pystyttävä erottamaan toisistaan eri käyttöliittymästandardien mukaiset käyttöliittymäprimitiivit. Itse malli ohjaa puolestaan sovelluksen tehtävien (engl. task model) ja komentojen (engl. task-dialogue model) toimintaa ja se liittyy oleellisesti sovelluskehittäjällä käytössä olevan ohjelmistotyökalun toimintamekanismiin. Mallipohjaisella työkalulla pyritään mallintamaan aina käyttöliittymän tuottamiseen asti suoritettavaa tehtävää. Määrityksen saatuaan työkalu suorittaa päättelyn sovelluksen ulkoasusta tai

toimintalogiikasta usein puoliautomaattisesti, jolloin vähäisempi määrä parametrisointeja joudutaan määrittelemään. Tällainen mekanismi ohjaa sovelluskehittäjän toteuttamistyötä ja sallii erilaisten käyttöliittymämallien kokeilemisen sekä tuottamisen ajonaikaisesti [Silva, P.P, 2000];[Stirewalt, R. E., Rugaber, S., 1998].

2. Käyttöliittymämallit

Käyttöliittymämalli yhdistettynä tätä tarkoitusta tukevaan sovellustyökaluun luo pohjan dynaamiselle suunnitteluajalle kehitys-ympäristölle, millä voidaan tuottaa monialustaisia käyttöliittymäkuvauksia. UIM-¹⁸ ja MB-UIDE-järjestelmät¹⁹ pyrkivät tarjoamaan ympäristön, missä käyttöliittymien kehitys ja toteutus on helpompaa, systemaattisempaa ja osin ammattimaisempaa, mitä perinteisillä käyttöliittymien kehitystyökaluilla on totuttu tekemään [Schlungbaum, E., 1996];[Silva, P.P, 2000]. Jotta tämä olisi mahdollista, niin käyttöliittymät täytyy kuvata usein deklaratiiivisilla malleilla. Yleensä mallipohjaiset järjestelmät ovat selkeästi tehtäväorientoituneita, missä käyttäjän tekemiä toimenpiteitä pyritään pitämään pohjana muiden mallien kehityksessä. Yhteisiä piirteitä korkean tason käyttöliittymämallien hyödyntämiselle on:

- Abstrahointi
 - Käyttöliittymämallit on kuvattu korkealla abstraktiotasolla esimerkiksi UIML-, XIML- tai XUL-pohjaisilla XML-dialekteilla.
- Systemaattisuus
 - Käyttöliittymämallit ohjaavat kehittäjää käyttämään systemaattisempia suunnittelu- ja toteutus-menetelmiä käyttöliittymille.
- Uudelleenkäytettävyys
 - Käyttöliittymämallit tarjoavat käyttöliittymienkuvausten uudelleenkäytettävyyden ja differoimisen alustasidonnaisesti esimerkiksi mobiili-, digi-tv- ja Web-ympäristöön sovitettut kuvaukset.
- Automatisointi
 - Käyttöliittymämallit tarjoavat pohjan käyttäjän tehtävien automatisoinnille, mitkä liittyvät suunnittelutyöhön, osin toteutukseen, riippuen käyttäkö käytettävä sovellustyökalu esimerkiksi automaattista sovellusgenerointia.

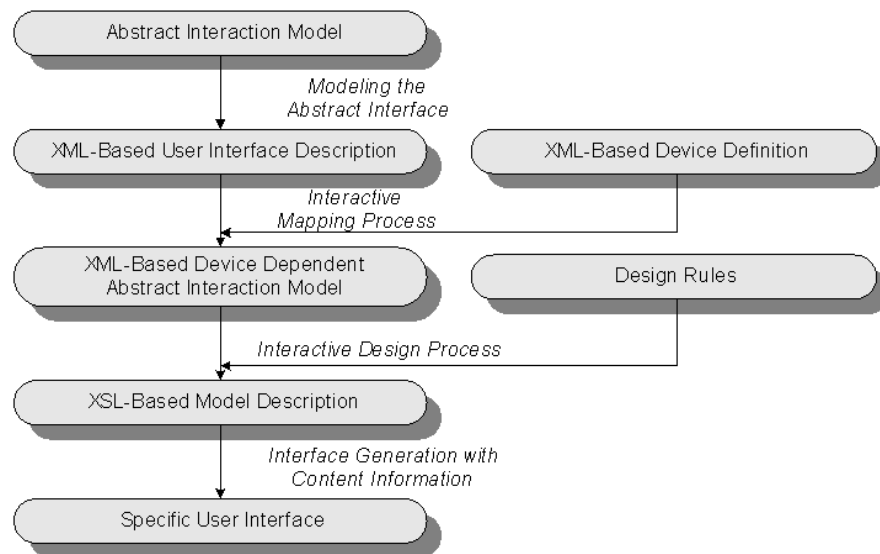
Kuvassa 1 on esitetty eräs XML-pohjainen käyttöliittymällin luonti- ja mallintamisprosessi, mitä mukailten tutkielman esitys etenee. Mikään esitetty

¹⁸ User Interface Model

¹⁹ Model-Based User Interface Development Environment

käyttöliittymämalli tai abstrakti käyttöliittymänkuvauskieli ei sinänsä esitä uutta tai käytännössä kokeilematonta käyttöliittymämalleihin liittyvää metodiikkaa. Tutkimuskenttänä malleihin liittyvät menetelmät ovat tuttuja jo Myersin [1995] esittelemän luokittelusta asti, kuten UIDE, MASTERMIND ja MECANO-tutkimuksissa on aikaisemmin esitetty. Viime aikoina uutena suuntauksena on käyttöliittymämallien kuvaamisessa tullut käyttää niin sanotun plastisuuden (engl. plasticity) käsitettä. Plastisuus esiteltiin Thevenin [1999] paperissa ensimmäisen kerran. Thevenin väitöskirjassa on kuvattuna plastisuuden teoreettinen sovelluskehys, siihen liittyvä taksonomialuokittelu ja itse notaatiomalli [Thevenin, D., 2001]. Siinä luotiin pohja itse käsitteelle ja kehysmallille, jonka tarkoituksena oli tuoda uutta lisäarvoa urauurtaviin ihmisen ja tietokoneen välisen vuorovaikutuksen -tutkimusalan suunnittelumetodiikoille, malleille ja metodologisille ohjeistuksille, kuten ADEPT- ja MAD-tutkimusprojekteille.

Ensimmäiset mallipohjaiset tekniikat, kuten UIDE [Silva, P.P, 2000] ja MasterMind [Schlungbaum, E., 1996] pyrkivät tekemään kaiken, eivätkä fokusoineet yhden asian tai sovelluskohteen mallintamiseen hyvin. Lopputuloksena oli usein hyvin rajoittuneita ja kompleksisia mallinnus- ja sääntöjoukkoihin perustuvia mekanismeja.



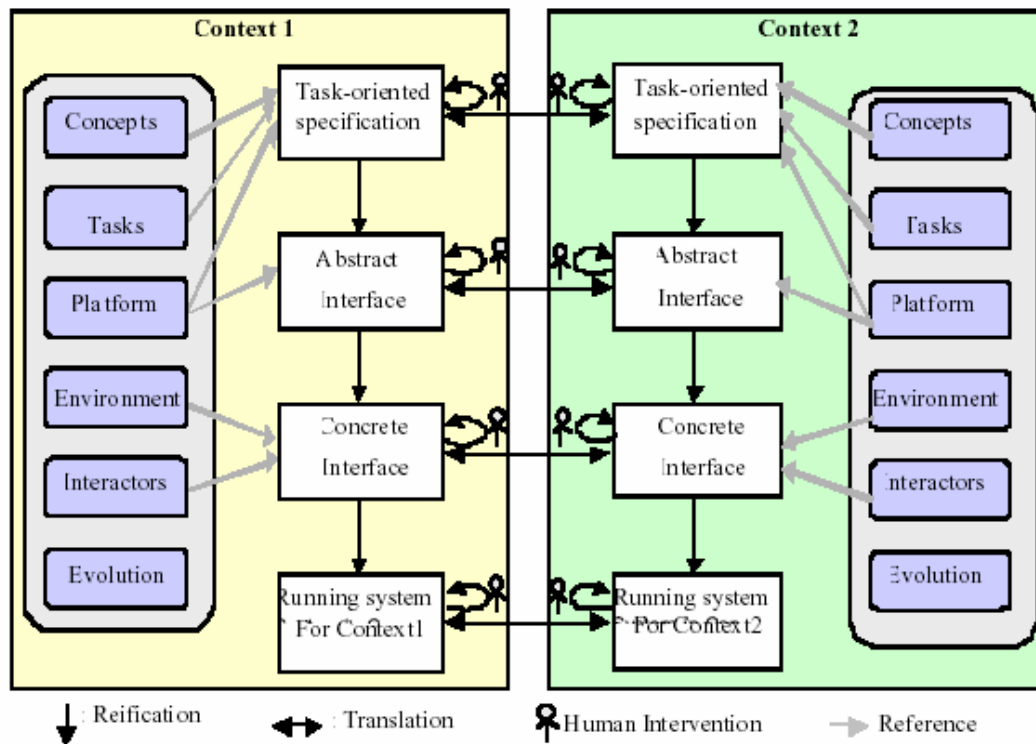
Kuva 1. Käyttöliittymämallin luontiprosessi [Müller, A., et al., 2001].

Müllerin [2001] esittämässä lähestymistavassa suunniteltujen mallien ja kuvauksien uudelleenkäyttö on mahdollistettu käyttämällä XML-pohjaisia teknologioita esitysmallien ja muunnoksien luomisessa. Malli eroaa Thevenin [1999] esittelemästä kehiksestä siinä, että XML-tekniikat on otettu perusteknologiaksi (joskaan Thevenin malli ei ota kantaa itse toteutukseen),

joiden ympärille ja hyväksikäyttämiseen käyttöliittymien kuvaukset perustuvat. Kuvassa 2 tämä lähestymistapa käyttöliittymämallin luontiprosessiksi on esitelty. Malli on varsin looginen, osin abstrakti ja tulevaisuutta ajatellen oikea arkkitehtuurinen ratkaisu, mikä mahdollistaa monialustaisten sovellusten kehitystyön.

2.1. Plastisuus käyttöliittymämalleissa

Plastisuus (engl. plasticity) käyttöliittymäsuunnittelussa voidaan määritellä helposti mukautuvaksi käyttöliittymämalliksi, jos mallia pystytään sovittamaan erilaisille konkreettisille käyttöliittymätyypeille. Tämä lähestymistapa ottaa erityisesti huomioon käyttöliittymämallin visuaalisen muodostamisen, missä fyysinen (esimerkiksi graafiset alkiot ja käyttöliittymäprimitiivit) ja looginen rakenne ovat tarkkaan eriteltyinä.



Kuva 2. Plastisten järjestelmien kehitysprosessi [Calvary G., et al., 2000].

Käsitteet plastisissa järjestelmissä on esitetty korkealla tasolla, joista muodostetaan käsitteet, tehtävät, kuvaukset ja aina lopulta kontekstisidonnaiset liittymät sovelluksen tai järjestelmän mallintamiseksi. Kontekstissa esitetyt käyttöliittymät kuvataan korkealla tasolla (esimerkiksi UIML, XUL tai XIML-pohjaisilla kielillä) ja tuotetut mallit, liittymät, ohjaavat suunnittelijaa käyttämään systemaattisimpia suunnittelu- ja toteutusmenetelmiä

käyttöliittymien toteutukselle. Kontekstissa kehitetyt kuvaukset ja mallit tarjoavat uudelleenkäytettävyyden ja liittymien differeroimisen alustasidonnaisesti. Mallit tarjoavat pohjan tehtävien automatisoinnille, jotka liittyvät suunnittelutyöhön (osin toteutukseen, riippuen käyttäkö kehitettävä työkalu tai järjestelmä esimerkiksi puoliautomaattista tai täysin automatisoitua sovellusgenerointia). [Calvary G., *et al.*, 2002]; [Calvary G., *et al.*, 2002]; [Thevenin, D., Coutaz, J., 1999].

3. Mallipohjaisen käyttöliittymätiedon sitominen

UIMS-järjestelmät (engl. User Interface Management System) [Silva, P.P, 2000] edustivat dialogia mallipohjaisen käyttöliittymän kanssa käyttäen formaalia kieltä kuvauskielenä. Termi on vanha nimitys käyttöliittymien hallintaan keskittyneille ohjelmistoille. Sitä mukaa, kun elementtien ja attribuuttien määrä kasvavat käyttöliittymämallin sanastossa, niin yhä vaikeampaa tällaisen sovelluksen ylläpitäminen on. Jatkuvasti päivittyvä tietomäärä esimerkiksi käyttöliittymien rakentamiseen tarkoitetulla lomake-editorilla tapahtuva työskentely sidotaan eli automatisoidaan tietorakenteiden struktuurien viittaushallinta suoraan (rakennetaan osaksi) sovellukseen on yhtä vanha kuin kuvauskielen käsite.

Käyttöliittymän määrittelyyn liittyvän ohjelmakoodin kuvaaminen, mikä koostuu fyysisistä widgeteistä, kuten painonapeista, ryhmälaatikoista ja muista elementeistä on usein järkevämpää määritellä XML-muodossa. Jokaisella käyttöliittymäalkiolla ja elementillä on omat erityisominaisuutensa, kuten tieto siitä, missä komponentti sijaitsee ja mikä komponentin nimi on. Nämä määrittelyt eivät sinällään tuo tai lisää toiminnallisuutta itse sovelluksen logiikkaan. Ne vain määrittelevät, miltä jokin elementti näyttää. XML-formaatti on hyvä tapa kerätä kaikki tämä tieto tiedostomuotoon, jolloin sovelluksen presentaatio- tai toimintalogiikkaa on helpompi käsitellä mallipohjaisesta työkalusta.

Bourret [2003] erottelee Web-sivustollaan XML-tiedon sidonnan ajonaikaisiin ja suunnittelunaikaisiin työkaluihin. Tiedon sidonnassa voidaan tehdä selkeä ero ajonaikaiseen ja suunnittelunaikaiseen sidontojen toimintotiloihin. Esimerkkinä tällaisesta XML-sovellustyökalusta on Altovan edistysellinen xmlspy. Sovellusten ja käyttöliittymäratkaisujen kirjoittaminen, mitkä hyödyntävät ja prosessoivat XML-dokumentteja käyttäen puupohjaista DOM-liittymää tai vastaavaa sovellusrajapintaa on matalan tason tehtävä, toistettavaa ja usein virhealtista. Helpompi tapa käsitellä XML-dokumentin rakenteita on käyttää visuaalista työkalua, jolla XML-elementtien ja puurakenteiden hahmotus on huomattavasti ymmärrettävämpää.

Ajonaikaiset sidontatyökalut tekevät parhaansa sitoakseen XML-dokumentin sen rakenteen ja rakennekuvauksen (engl. reflection) perusteella, kuten automaattisesti tapahtuvan itsehavainnoinnin avulla. Suunnittelun aikaiset työkalut nojaavat puolestaan johonkin formaaliin skeemamalliin, minkä perusteella halutut sidonnat tapahtuvat. XML-skeemakielen käyttäminen määrittelykielenä tarjoaa hienovaraisemman tavan määrittellä uusia rakenteita ja rakenteisia tietoja, joiden perusteella tyypittää esimerkiksi uusia käyttöliittymätietoon sidonnoisia tietotyyppejä. Ohjelmoija määrittelee tarvittavat operaatiot järjestelmälle tai sovellusohjelmalle jollain käyttöliittymämalliin liittyvällä määrittelykielellä (mallipohjainen suunnittelu). Käyttöliittymämallin tietoon liittyy muun muassa seuraavia peruskäsitteitä, joita Myers [2003] esittelee käyttöliittymien kehitystyön luentosarjassaan:

- Malliin liittyvä käyttöliittymätieto on korkean tason kuvaus käyttöliittymämallista.
- Käyttöliittymätieto on yleensä deklaratiiivinen esitys osien ja suhteiden välisestä luetteloinnista, kuten rajoitteiden, asemointien ja semantiikan määrittelyistä.
- Mallipohjainen sovellusohjelma luo automaattisesti käyttöliittymätiedon suoran manipuloinnin kautta.
- Käyttöliittymämalli käyttää matalan tason työkalupakkia fyysisille käyttöliittymäwidgeteille alustasidonnaisesti.
- Käyttöliittymämalli tarjoaa usein käyttäjälleen mallipohjat ja sovelluskoodirungot käyttöliittymäprototyyppien suunnitteluun.

3.1. XML tiedon sidonta

XML-sidonnan kautta luodut komponentit tuotetaan (engl. generate) XML-skeemasta. Komponentti edustaa skeemaa luokkiensa ja ominaisuuksiensa kautta. Tämä on täysin riippuvainen alustasta, missä itse tiedon sidonta suoritetaan. Käytännössä jokainen XML-elementti edustaa luokkarakennetta ja jokainen attribuutti on vastaavasti luokassa oleva muuttuja. Dokumentin rakenne luetaan komponenttiin sisään ja sitä voidaan käsitellä käytettävällä ohjelmointikielellä. Tällaisia ovat esimerkiksi Java, .NET-kielet, Delphi ja Python. Sidonnassa syntyneet komponentit pitävät huolen tiedon oikeellisuudesta, joten tuotetut XML-dokumentit ovat varmasti yhteensopivia käytettävän skeeman kanssa. Sidonta mahdollistaa monialustaisten sovellusten kehitystyön ja XML-dokumentit on metakielenä käytettävissä jokaisella kohdealustalla. Nykyään on olemassa joukko työkaluja ja sovelluskehityspaketteja, jotka automatisoivat nämä tiedon sitomiseen liittyvät

toimenpiteet käyttäjän puolesta, kuten Java-pohjainen Castor [Sosnoski, D.M, 2003a, 2003b].

W3C-konsortio on määrittelemässä standardia sovellusrajapintaa, minkä perusteella lähitulevaisuuden ohjelmistotyökalut voivat pyytää skeemaprosessorilta (engl. processor) kehittyneitä analysointiin- ja toteutukseen liittyviä metodiikoita. Prosessori voi kohdistaa työkalussa esimerkiksi seuraavia kysymyksiä: "Minkä toiminnon tietyssä kohdassa XML-dokumenttia voin tehdä?", "Voinko lisätä uuden XML-elementin tähän kohtaan?", "Voinko päivittää tämän XML-noodin tällä arvolla?". DOM tason 3 abstraktit skeemat ja lataus ja tallennus spesifikaatio [W3C, 2003a] määrittelee tärkeän käsitteen - abstraktit skeemat, jotka ovat tarpeeksi yleisiä kattaakseen W3C:ssä määritellyt XML-skeemat ja XML-dokumenttien tyyppimäärittelyt. Se käsittää mahdollisesti muitakin skeemakieliä. Tyyppimäärittelyt kuitenkin DTD:tä hyödyntämällä ovat jo vanhentunutta teknologiaa. Kunnes tämä spesifiointityö saatetaan loppuun, ja on laajalta toteutettuna, niin määrittelyön mukainen sovellusrajapinta mahdollistaa minkä tahansa skeemaprosessorin kytkemisen mihin tahansa XML-pohjaiseen sovellustyökaluun. XML-sidontaan liittyy muun muassa seuraavia peruskäsitteitä ja sääntöjä:

- Tiedon sidonnan on tapahduttava yhden tai useampien XML-elementtien kanssa.
- Tiedon sidonta tapahtuu ohjelmallisesti oliorakenteiden käsittelyn kautta, joilla hallinnoidaan XML-dokumentin rakenteiden käsittelyä.
- Tiedon sidonta vaatii useita ohjelmallisia keinoja XML-dokumenttien rakenteiden käsittelyyn.
- Keinoja ovat muun muassa luominen, lukeminen, päivitys, poistaminen, validointi, prosessointi, muuntaminen ja muistin hallinta.

3.2. XML-sidonnan käsittely

XML-tiedon sidonnan (engl. data binding) avulla XML-dokumenttien käsittely tapahtuu sidottujen olioiden kautta. Sidonta tapahtuu XML-dokumentin rakenteita havainnoiden. Olioiden esittely tapahtuu käyttämällä jotain ohjelmointikieltä, kuten Javaa tai Delphiä. Sidonta tarjoaa näin ollen korkean tason kerroksen dokumentin rakenteisiin, joita käsitellään ohjelmakoodin kautta. Menetelmää hyödyntämällä kehittäjän ei välttämättä tarvitse olla tietoinen XML-skeemoihin ja XML-parsereihin tai jäsentimiin liittyvistä monimutkaisuuksista. XML-dokumentissa esiintyviin XML-elementteihin voidaan sidonnan kautta liittää esimerkiksi tapahtumaohjattuja kontrollirakenteita. DOM-parserin tehtävänä on luoda XML-dokumentin rakenteesta hierarkkinen puurakenne, mikä edelleen jakaantuu XML-entiteetteihin. [Sosnoski, D.M, 2003a];[McLaughlin, 2002].

Puurakenteessa navigoiminen ja muutokset on helpompi toteuttaa, kuin esimerkiksi käyttämällä SAX-pohjaista tapahtumaohjattua jäsentelyä. Ongelmana puun käsittelyssä on, että XML-dokumentin rakenne täytyy kokonaisuudessaan ladata muistiin. Ratkaisuja tietyn DOM-osarakenteen tai puurakenteen osittaiseen käsittelyyn on olemassa, kuten W3C:ssä meneillään oleva W3C DOM Load and Save - määrittely [W3C, 2003a]. DOM-tason rakenteita voidaan suorittaa käytettävässä sovellustyökalussa ja ajonaikainen dynaaminen käyttöliittymämallin muokkaaminen on mahdollista. Käyttöliittymämalli on sovelluskohtainen, joka voi olla esimerkiksi XForms-lomakkeen muokkaustila. Mallien kautta syntyneitä rakenteita voidaan tuottaa automaattisesti, jolloin kieliriippumattomien tynkäluokkien tai mahdollisesti täysin automatisoitujen luokkarakenteiden sekä tapahtumaohjatun koodin että presentaatiotason käyttäytyminen toteutettuina ovat mahdollisia.

Toinen lähestymistapa on rakentaa sovellus XML-skeemoissa esitellyistä tiedoista, yhdistellen osa-skeemoja. Tämä on kuitenkin monimutkaisempi tapa, kuin tiedon yhdistäminen presentaatiomalliin ja kontrollirakenteisiin. Tämä menetelmä mahdollistaa käyttäjän XML-dokumentin muokkaamisen tähän tarkoitusta varten sovitetulla sovellustyökalulla. Tällaisessa tavassa yhdistyvät yleensä suunnittelumallien käyttö, erityisesti yleisesti käytössä olevat MVC-suunnittelumallit (engl. Model View Controller), mitkä erottelevat sovelluslogiikan ja sovelluksen ulkoasun eli presentaation.

4. Yhteenveto

Osiassa 1 esiteltiin käyttöliittymämallin käsitteellinen pohja ja luotiin katsausta tutkimusalalla vallitseviin käsitteeseen sekä malleihin. Osiossa 2 käsiteltiin käyttöliittymämalleihin liittyvää käsitteistöä ja esiteltiin kaksi lähestymistapaa näiden rakentamiseksi. Osiossa 3 käsiteltiin käyttöliittymätietoon liittyvän tiedon, metatiedon sitomista XML-rakenteiksi ja niiden käsittelyä. Lisäksi esiteltiin lyhyesti XML-sidonnan periaate ja siihen läheisesti liittyvät peruskäsitteet.

Microsoft on esitellyt tulevan käyttöjärjestelmänsä beetaversioon ja siihen mahdollisesti kuuluvat uudet teknologiset erityispiirteet. Beeta-asteella oleva käyttöjärjestelmä nimeltään Longhorn sisältää uuden presentaatio- ja käyttöliittymätason koodinimeltään Avalon. XML-dialektina tällä kerroksella toimii XAML-merkkikieli. Se on erehdyttävästi hyvin samankaltainen deklaratiivinen ja laajennettava kieli, kuten XUL tai XIML, joilla luodaan käyttöliittymäkuvauksia. XAML:in taustalla voidaan käyttää tapahtumaohjatun koodin hallintaan (niin sanottu tukikoodi) esimerkiksi C#-kielellä luotuja sovellusrakenteita [Fortes, F., 2003]. Käyttöliittymän luonti onnistuu sujuvasti,

koska itse sisältö (toisin sanoen käyttöliittymämalli) erotellaan selkeästi pois ohjelmointityöstä.

Graafiseen suunnitteluun orientoitunut kehittäjä voisi suunnitella käyttöliittymän ja ohjata luodun XAML-dokumentin sovelluskehittäjälle, joka edelleen luo tarvittavan sovelluslogiikan annetulle käyttöliittymämallille. Vaikka kehitystyö käyttöjärjestelmän suhteen on beetavaiheessa, niin nyt jo on havaittavissa varsin käänteentekeviä ominaisuuksia. Longhornin sovelluskehityspaketti tarjoaa useita mielenkiintoisia uusia tapoja jäsentää käyttöliittymän widgettejä, kuten uusia jäsennyspohjia monimutkaisten käyttöliittymärakenteiden jäsentelyyn.

Viiteluettelo

- [Ali M. F., *et al.*, 2001] M. F. Ali, M. A. Pérez-Quinones, M. Abrams, A Multi-Step Process for Generating Multi-Platform User Interfaces using UIML, 2001.
- [Calvary G., *et al.*, 2000] G. Calvary, J. Coutaz, D. Thevenin, Embedding Plasticity in the Development Process of Interactive Systems, *HUC2000 Workshop on Resource Sensitive Mobile HCI*, CLIPS-IMAG, Institut d' Administration et de Gestion.
- [Calvary G., *et al.*, 2002] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, N. Souchon, L. Bouillon, J. Vanderdonckt, Plasticity of User Interfaces: A Revised Reference Framework, CLIPS-IMAG, Institut d' Administration et de Gestion.
- [Fortes, F., 2003] F. Fortes, XAML vs XUL, 28.10.2003 weblog, <http://blogs.gotdotnet.com/ffortes/PermaLink.aspx/b7708c79-cbb9-4b2e-a22d-f8a7b1416b7b>, 2003.
- [Bourret, 2003] R. Bourret., XML Data Binding Resources, <http://www.rpbourret.com/xml/XMLDataBinding.htm>.
- [Korhonen, J., 2002] J. Korhonen, UIML-määrittelykielen avulla esitetyn käyttöliittymän muodostaminen, *Pro gradu*, Tietotekniikan laitos, Jyväskylän yliopisto, 2002.
- [McLaughlin, 2002] B. McLaughlin, Why Data Binding Matters, <http://www.onjava.com/pub/a/onjava/2002/05/15/databind.html>.
- [Müller, A., *et al.*, 2001] A. Müller, P. Forbrig, C. Cap, Model-Based User Interface Design Using Markup Concepts, University of Rostock, Dept. of Computer Science.
- [Myers, B.A., 2003] B.A. Myers, Advanced User Interface Software, <http://www-2.cs.cmu.edu/~bam/uicourse/830spring03/schedule.html>.

- [Puerta, A., Eisenstein, J., 2001] A. Puerta, A. Eisenstein, XIML: A Universal Language for User Interfaces, RedWhale Software, 2001.
- [Ruini, H., 2000] Ruini. H., Englanti - Suomi XML-sanasto, Helsingin yliopisto, <http://www.cs.helsinki.fi/u/ruini/structure/xml/sanasto.html>.
- [Schneider, K.A., Cordy, J.R., 2002] K.A. Schneider, J.R. Cordy, Abstract User Interfaces: a Model and Notation to Support Plasticity in Interactive Systems, *Interactive Systems: Design, Specification and Verification*, Springer Verlag *Lecture Notes in Computer Science* 2220, Berlin, p. 28 - 48.
- [Schlungbaum, E., 1996] E. Schlungbaum, Model-based User Interface Software Tools, Current state of declarative models, Graphics, visualization & Usability Center, Georgia Institute of Technology, GIT-GVU-96-30, 1996.
- [Silva, P.P, 2000] P.P. Silva, User Interface Declarative Models and Development Environments: A Survey, Dept. of Computer Science, University of Manchester.
- [Sosnoski, D.M, 20003a] D.M. Sosnoski, Data binding, Part 1: Code generation approaches - JAXB and more, <http://www-106.ibm.com/developerworks/xml/library/x-databdopt/>.
- [Sosnoski, D.M, 20003b] D.M. Sosnoski, Data binding, Part 2: Performance, <http://www-106.ibm.com/developerworks/xml/library/x-databdopt2/>.
- [Stirewalt, R. E., Rugaber, S., 1998] R. E., Stirewalt, S., Rugaber, Automating UI Generation by Model Composition, *Proceedings of the 13th IEEE International Conference on Automated Software Engineering, ASE98*, 1998.
- [Thevenin, D., Coutaz, J., 1999] D. Thevenin, J. Coutaz, Plasticity of User Interfaces: Framework and Research Agenda, Human-Computer Interaction, *INTERACT'99*.
- [Thevenin, D., 2001] D. Thevenin, Adaptation in Human Computer Interaction: the case of Plasticity, *PhD thesis*, Joseph Fourier University, Grenoble, 2001, p. 234.
- [Vanderdonckt, J., et al., 2001] J. Vanderdonckt, L. Bouillon, N. Souchon, Flexible Reverse Engineering of Web Pages with VAQUISTA, *Working Conference on Reverse Engineering*, p. 241 - 248, 2001.
- [Vlist, E., 2002] E. Van Der Vlist, *XML Schema*, O'Reilly, 2002.
- [W3C, 1999] W3C, XSL Transformations (XSLT) Version 1.0, <http://www.w3.org/TR/xslt>.
- [W3C, 2003a] W3C, DOM Load and Save, <http://www.w3.org/TR/DOM-Level-3-LS/>.
- [W3C, 2003b] W3C, Document Object Model (DOM) Level 3 Validation Specification, Version 1.0, <http://www.w3.org/TR/DOM-Level-3-Val/>, W3C Candidate Recommendation.

[W3C, 2003c] W3C, Document Object Model (DOM) Level 3 Abstract Schemas Specification, Version 1.0, <http://www.w3.org/TR/DOM-Level-3-AS/>, W3C Note.

[XUL, 2003] Open XUL Alliance, <http://xul.sourceforge.net/>.

Kuntien verkkosivujen sisällön kartoitus

Minna Sundström

Tiivistelmä.

Tutkimus kuvaa kahdentoista kunnan verkkosivujen sisältöä ja sähköisiä palveluita asiakkaan näkökulmasta. Kuntien verkkosivujen sisällön kartoitus tapahtui 15.9. – 15.12.2003 välisenä aikana. Tutkimuksessa asiakkaat käsittävät sekä kunnan asukkaat, vierailijat että kunnan yritykset.

Tutkimuksessa pyritään vastaamaan neljään kysymykseen: Mistä palveluista kunta tiedottaa sivuillaan? Kuinka helposti kohderyhmän tarpeet on huomioitu? Mitä sähköisiä palveluita on tarjolla? Mitä vuorovaikutusmahdollisuuksia kunta tarjoaa sähköisesti?

Tutkimus tehtiin listamenetelmänä ja pisteyttämällä tiedon löydettävyys ja esitystavan laatu. Lähteenä käytettiin kuntien verkkosivuja. Kuntien palvelut osoittautuivat laadultaan tiedottaviksi. Perustehtävistä tiedotus onnistui parhaiten sosiaali- ja sivistystoimessa. Vuorovaikutuksen mahdollisuus jäi kuntien sivuilla suhteellisen vähäiseksi.

Sähköisiä asiakirjoja on tarjolla, mutta sähköisiä palveluja on vielä varsinkin asukasmääriltään pienemmillä kunnilla vähän tai ei lainkaan. Kuntien seudullinen yhteistyö näkyy myös verkossa seutuportaalien ja aineistoyhteistyön merkeissä.

Avainsanat ja -sanonnat: internet, kunta, palveluportti, portaali, sähköinen asiointi, sähköinen palvelu, verkkosivu, verkkosivusto, verkkoasiointi, vuorovaikutus

CR-luokat: H 3.1., H 3.5., H 5.4

1. Johdanto

Viime vuosituhaten lopussa sisäasiainministeriö aloitti lukuisia hankkeita edistämään julkishallinnon sähköisten palveluiden luomista ja kehittämistä.

Tutkimuksessani pyrin kartoittamaan kahdentoista kunnan verkkosivujen sisältöjä, sähköisiä palveluita, seudullista yhteistyötä verkossa sekä vuorovaikutusmahdollisuuksia kuntalaisten näkökulmasta.

Tutkimus suoritettiin käyttämällä listamenetelmää ja pisteytystä tietojen esitystavasta. Sisältöön tutustuminen suoritettiin 15.9. – 15.12.2003 välisenä aikana.

Tutkimuksessa pyritään vastaamaan neljään kysymykseen: Mistä palveluista kunta tiedottaa sivuillaan? Kuinka helposti kohderyhmän tarpeet on huomioitu? Mitä sähköisiä palveluita on tarjolla? Mitä vuorovaikutusmahdollisuuksia kunta tarjoaa sähköisesti?

Tutkimus tehtiin listamenetelmänä ja pisteyttämällä tiedon löydettävyys ja esitystavan laatu. Lähteenä käytettiin kuntien verkkosivuja. Kuntien palvelut osoittautuivat laadultaan tiedottaviksi. Perustehtävistä tiedotus onnistui parhaiten sosiaali- ja sivistystoimessa. Vuorovaikutuksen mahdollisuus jäi kuntien sivuilla suhteellisen vähäiseksi.

Sähköisiä asiakirjoja on tarjolla, mutta sähköisiä palveluja on vielä varsinkin asukasmääriltään pienemmillä kunnilla vähän tai ei lainkaan.

Kuntien seudullinen yhteistyö näkyy myös verkossa seutuportaaleina ja aineistoyhteistyönä. Ongelmana oli tarkastella asukasmäärältään isojen kuntien ja pienten kuntien sivustoja samalla asteikolla. Ongelma ratkaistiin tarkastelemalla kuntien peruspalveluiden esitystapaa.

2. Tutkimuksen lähtökohdat

Kuntien verkkosivujen sisällön kartoitus on erityisen ajankohtainen, sillä valtakunnallisellakin tasolla pyritään kohottamaan kuntien verkkosivujen profiilia ja saamaan kuntien palveluja myös sähköiseen muotoon. Julkishallinnolla on jatkuvat muutospaineet saada jäykkä organisaatorakenne toimimaan nopeasti muuttuvassa sähköisessä maailmassa sekä vastata asiakkaitensa eli kuntalaisten vaatimuksiin. [Julkinen verkkoasioinnin kehittämishanke, 1/2000].

Yksityinen sektori on ottanut laajasti käyttöönsä Internetin tuomat palveluntarjontamahdollisuudet omien verkkosivujen kautta, hyödyntänyt sivustojen ajankohtaisuutta sekä tietojen jatkuvaa uusiutumisen- ja päivittämismahdollisuutta. Tässä asiassa julkinen sektori on selkeästi jäänyt yksityisen sektorin kehityksestä jälkeen. Suomi mainostaa itseään tietoyhteiskuntana, mutta vasta vuosituhaten vaihtuessa myös kunnat ja valtionhallinto on lähtenyt selvittämään omien palvelujen ja tietojen siirtämistä Internetiin. 90-luvun lopussa perustettiin työryhmiä kartoittamaan sen hetkinen kuntien verkkosivujen tilanne.

Sisäasianministeriön asettamissa työryhmissä ja hankkeissa pohdittiin, mitä sähköisiä palveluja verkkosivuille voi laittaa ja miten näitä palveluja tulisi jatkokehittää. Työryhmät selvittivät myös kuntien verkkosivujen samankaltaistamisen tarvetta. Sisäasiainministeriö ei ole ainoa taho, joka asettaa kunnille velvoitteita, vaan myös EU määrää omia velvoitteitaan

jäsenmaittensa sähköisille palveluille. [Julkisen verkkoasioinnin kehittämishanke, 3/2002].

Tulevaisuudessa julkiselle hallinnolle asetetaan lisää paineita, sillä käyttäjät eivät keskity vain tietokoneen ja tietoverkkojen käyttäjiin, vaan myös muut käyttötavat on otettava huomioon. Matkapuhelimen, digitaalitelevision ja lisäinnovaatiot tuovat omat haasteensa yhteensopivuutensa puolesta. Tulevaisuudessa on varauduttava tekemään sivustot niin että ne eivät ole käyttövälineriippuvaisia. [Julkisen verkkoasioinnin kehittämishanke, 2001].

2.1. Aikaisempi tutkimus

Sisäasiainministeriö aloitti vuonna 1999 Julkisen verkkoasioinnin kehittämis- eli JUNA -hankkeen, jonka yhtenä tarkoituksena oli selvittää, kohtaavatko kuntien verkkopalvelut sähköisen asiakkaan. Tästä hankkeesta syntyi lukuisia tutkimuksia ja se sai lisähankkeita sekä kumppanuushankkeita. Hankkeista on syntynyt lukuisia julkaisuja julkishallinnon visioista ja strategioista koskien sähköisiä palveluja ja sähköistä viestintää. Hankkeen työryhmä tutki Suomen kuntien verkkosivujen palveluja asiakkaan näkökulmasta vuonna 2001. Verkkosivut kehittyvät ja muuttuvat jatkuvasti ja tämän vuoksi tutkimukset vanhenevat nopeasti [Ruusula, 2001].

JUNA -hankkeen avainhankkeena ollut Asiakaspäätteet ja kansalaisten verkkoasiointi tutki julkisten asiakaspäätteiden käyttöastetta ja verkkoasiointia edistäviä tai estäviä seikkoja. Hanke teki tutkimusta myös siitä mitä tietoa kansalaiset hakevat ja millä sivuilla he käyvät. Tutkimuksen mukaan julkisten palveluiden sivuista eniten haettiin tietoja kunnallisista palveluista. [Saarijärvi M., 2001].

2.2. Tutkimusmenetelmät

Tutkimusotteeni on laadullinen. Tutkimuksessani tarkastelen kuntien verkkosivujen sisältöä ja sähköisiä palveluja. Sisällön ja sähköisten palvelujen arvioimisessa käytin mukailen Matti Ruusulan tiedottavan palvelun listamenetelmää.

Listamenetelmässä on listattu tärkeimmät asiat, jotka kuntien sivuilla pitäisi esiintyä ja niihin liittyvät sisältövaatimukset. Lista koostuu seuraavista kohdista: sivistystoimi, sosiaali- ja terveystoimi, tekninen toimi, elinkeino ja yleishallinto. Jokaisessa kohdassa tarkastellaan erikseen näiden alakohtia ja sisältövaatimuksia, jotka vaihtelevat joko tiedon puuttumisesta täydelliseen palvelun tai tiedotteen kuvaukseen. Pisteitä ei saanut lainkaan, jos tieto puuttui tai sitä ei löytynyt sieltä, missä se loogisesti sijaitisi muiden kuntien sivuilla. Yhden pisteen sai, jos palvelusta oli kerrottu edes yhteystiedot ja kaksi pistettä, jos palvelusta oli kerrottu muutamalla lauseella. [Ruusula, 2001].

Tutkimuksessa maksimipistemäärä oli neljä ja sen sai, jos palvelu oli kuvattu yksityiskohtaisesti ja sen lisäksi oli kerrottu vielä yhteystiedot. Asukasluvultaan pienemmillä kunnilla on usein vähemmän palveluja kuin suuremmilla, tästä johtuen sivujen arvioinnissa keskitytään ainoastaan tiedon laatuun.

2.3. Tutkimusaineisto

Tutkimusaineistona on kahdentoista kunnan Internetissä julkaistavat verkkosivustot. Tutkimuksen kunnat ovat Hattula, Hauho, Hausjärvi, Hyvinkää, Hämeenlinna, Janakkala, Kalvola, Lammi, Loppi, Renko, Riihimäki ja Tuulos. Kuntien osoitteet on saatu käyttämällä osoitetapaa www.kunta.fi. Kunnista kolme on kaupunkia, mutta luettavuuden vuoksi käytän kaikista nimitystä kunta. Tutkimusaineisto valittiin sillä perusteella, että edellä mainitut kunnat liittyvät Tampereen yliopiston täydennyskoulutuskeskuksen Hajautettu työ Kanta-Hämeessä -projektiin. Aineiston tarkastelu tapahtui 15.9. - 15.12.2003 välisenä aikana.

Taulukosta 1. selviää kuntien verkkosivuston osoitteet, kunnan väestöluku sekä pinta-ala. Väestöluku ja pinta-ala on saatu kuntien verkkosivuilta ja tiedot puuttuvat, jos niitä ei sivuilta löytynyt.

Kunta	verkkosivun osoite	Väestöluku	Pinta-ala km²
Hattula	www.hattula.fi	9165	454,76
Hauho	www.hauho.fi	4000	ei tietoa
Hausjärvi	www.hausjarvi.fi	8171	362,83
Hyvinkää	www.hyvinkaa.fi	42 997	336,66
Hämeenlinna	www.hameenlinna.fi	46 734	ei tietoa
Janakkala	www.janakkala.fi	15 483	586
Kalvola	www.kalvola.fi	3427	334
Lammi	www.lammi.fi	5614	610,8
Loppi	www.loppi.fi	7578	655,8
Renko	www.renko.fi	2328	290
Riihimäki	www.riihimaki.fi	26341	126
Tuulos	www.tuulos.fi	1 531	171

Taulukko 1. Kuntien verkkosivuilla olevat tiedot 31.12.2002

3. Tutkimustulokset

3.1. Sivistystoimi

Opetus	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
		2	4	6
Kirjastot	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
				12
Liikunta	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
	1		6	5
Nuorisotoimi	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
	1		3	8

Taulukko 2. Sivistystoimi

Kuuden kunnan Opetus-osio piti sisällään koulujen (ala- ja yläkoulut, lukiot, toisen asteen koulut) esittelyn, yhteystiedot, opetustoimen tavoitteet, työ- ja loma-ajat, viikoittaisen ruokalistan sekä ainakin rehtoreiden yhteystiedot. Useimmilla kuntien kouluista oli omat kotisivut. Kaksi kuntaa esitteli ainoastaan koulun yhteystiedot ja neljä yhteystietojen lisäksi muutaman lauseen kuvaten koulutoimea kunnassa.

Kirjastot olivat kaikissa kunnissa eniten tietoa ja palveluita sisältävät sivut. Kaikissa kunnissa annettiin tieto kirjastojen aukioloajoista, yhteystiedot pääkirjastosta ja toimipaikoista sekä henkilöstön tiedot. Aineistotietokannat olivat siirretty joko Vanajaverkon Web-Origoon, Lopen Web-Origoon tai RATAMO-kirjastojen tietokantoihin.

Liikunta- ja nuorisotoimi oli useissa kunnissa yhdistetty esitettäväksi samalla sivulla, tällöin otsikkona sivustoilla oli vapaa-aika. Yksi kunta ei tarjonnut tietoa kummastakaan palvelusta. Muilla kunnilla oli esiteltyinä liikuntapalvelut (laskettelukeskus, uimahalli, uimarannat), aikataulut, tilat, yhteyshenkilöt ja maksut. Kahdeksan kuntaa kertoi tarkasti nuorisotoimen

alueen tiedot. Nuorisotilat ja kerhotoiminta yhteystietoineen oli selkeästi kuvattu. Kolme kuntaa kertoi vain vastuuhenkilön tiedot ja kuvasi muutamalla lauseella nuorisotoimen toimintaa.

3.2. Sosiaali- ja terveystieteiden palvelut

Päivähoito	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
			1	11
Kotipalvelu	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
		1	1	10
Vammais- palvelut	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
			3	9
Päihdehuolto	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
	3		3	5
Terveys- keskukset	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
			2	10

Taulukko 3. Sosiaali- ja terveystoimi

Sosiaali- ja terveystieteiden palvelut oli kaikilla kunnilla kattavasti esitelty. Ainoa puute oli päihdehuollon sivut, jotka puuttuivat kolmelta kunnalta. Päivähoito oli esitelty kaikilla yksityiskohtaisesti kuvaillen päiväkotia, hakemusmenettelyä, maksuja ja tavoitteita. Kahdeksan kuntaa tarjosi sähköisen päivähoitohakemuksen ja kolmella kunnalla hakemuksen pystyi myös lähettämään sähköisesti.

Kotipalvelu liitetään usein vanhustenhuoltoon ja näin ollen se löytyy kuntien sivuilta usein vanhustenpalvelut -osion alta. Kymmenellä kunnalla oli tiedot kotipalvelun sisällöstä, yhteyshenkilöistä ja maksuista. Vain kaksi kuntaa tarjosi suppeammat tiedot palvelusta.

Vammaispalveluissa kerrottiin kuljetuspalveluista, tulkkipalveluista sekä apuvälineiden hankinnasta. Yhdeksältä kunnalta löytyi edellä mainitut tiedot ja kolme kuntaa antoi kuvauksen sekä yhteystiedot vammaispalveluvastaavasta.

Päihdehuollosta kolmen kunnan sivuilta ei löytynyt tietoa laisinkaan. Toiset kolme kuntaa kertoivat päihdehuollosta vain muutamalla sanalla tai olivat yhdistäneet sivut nuorisotoimeen. Vain viisi kuntaa kahdestatoista kuvasi raittiussivut yksityiskohtaisesti. Päihdehuolto löytyi useimmilta kohdasta elämän kriisitilanteet tai sitten otsikon raittiutta.

Terveyskeskuksista kolmen (Hattula, Hauho, Kalvola) kunnan kohdalla sivu siirtyi Hämeenlinnan seudun kansanterveystyön kuntayhtymän sivuille. Kahden (Lammi, Tuulos) kunnan kohdalla sivu siirtyi Lammi-Tuulos kansanterveystyön kuntayhtymän sivuille. Lopen kunnan terveyskeskusten sivu siirtyi Riihimäen terveydenhoitopalveluihin. Loput kunnat olivat esitelleet palvelunsa omilla sivuillaan, joko tarkan selvityksen kanssa tai vain yhteystiedot mainiten.

3.3. Tekninen toimi

Vesihuolto	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
	1	1	2	8
Jätehuolto	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
	2	1	1	8
Rakennus-tarkastus	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
		4	2	6
Tontit	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
			3	9

Taulukko 4. Tekninen toimi

Kunnallistekniikka on esitelty lähes kaikkien kuntien sivuilla. Vesihuollosta on parhaiten kerrottu viiden kunnan sivuilla, joista jokaisessa on mahdollista lähettää vesimittarilukema tai seurata veden kulutusta sähköisesti. Kahdeksan kunnan sivuilla kerrotaan, mikä taho vastaa vedenhuollosta ja tämän yhteystiedot. Yksi kunta ei ilmoittanut lainkaan tietoja kunnan vesihuollosta.

Jätehuollosta ei kahdella kunnalla ollut mitään tietoa sivuillaan ja kaksi kuntaa esitti vain yhteystiedot ja muutaman lauseen siitä mitä kunnan

jätehuolto käsittää. Kahdeksan kuntaa selvitti tarkasti kuka hoitaa kunnan jätehuollon ja mistä löytyy ongelmajätteiden kierrätyspisteet.

Rakennustarkastus käsitti parhaimmillaan rakennuslupahakemuksen, joka löytyi sähköisenä kolmelta kunnalta. Rakennustarkastuksesta neljä kunta antoi ainoastaan yhteystiedot rakennustarkastajasta ja edellisten tietojen lisäksi kaksi kuntaa selvitti muutamalla lauseella mitä rakennustarkastus heidän kunnissa tarkoittaa. Yksi suurimmista kunnista oli sisällyttänyt laajan rakentajan oppaan verkkosivuilleen.

Tonttitarjonta oli jokaisen kunnan sivuilla esitelty hyvin. Yhdeksällä kunnalla oli tarkat tiedot sen hetkisestä tonttitarjonnasta, neliöhinnoista ja yhteystiedot. Yhdellä kunnalla oli edellisten tietojen lisäksi sähköinen tonttivaraushakemus.

3.4. Kunnan hallinto

Hallinnollinen työ	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
	1	1	7	4

Kunnan hallitus	nimet	ei tietoa	esityslistat	ei tietoa	pöytäkirjat	ei tietoa
	12		8	4	12	
Kunnan valtuusto	nimet	ei tietoa	esityslistat	ei tietoa	pöytäkirjat	ei tietoa
	12		8	3	12	
Lauta-kunnat	nimet	ei tietoa	esityslistat	ei tietoa	pöytäkirjat	ei tietoa
	10	1	5	5	12	

Taulukko 5. Kunnan hallinto

Kunnan hallinnollista työtä kuvaava sivu löytyy 11 kunnalta. Ainoastaan yksi kunta ei kerro hallinnon työstä mitään, ei edes kunnanjohtajan tietoja. Tosin tämäkin kunta mainitsee kunnanjohtajan nimen yhteydenotto-lomakkeessa. Vain neljä kuntaa kuvaa yksityiskohtaisesti kunnan hallinnon toimintaa ja sen organisaatorakennetta. Kunnan työntekijöiden tiedot löytyivät jokaisen kunnan sivuilta, Hausjärvellä oli jopa palautelomake, jolla sai yhteyden kunnan tiettyyn henkilöön.

Kunnanhallituksen ja -valtuuston jäsenten nimet ja puoluekanta löytyvät kaikilta kunnilta. Ainoastaan yhdeltä kunnalta puuttui jäsenten osoitetiedot, näiden tiedusteluun annettiin kuitenkin yhteystiedot. Hallituksen ja valtuuston esityslistoille oli varattu tiedonantopaikka kahdeksan kunnan sivuilta, mutta hallituksen osalta se puuttui neljästä kunnasta ja valtuuston kohdalta kolmesta. Kaikilta kunnilta löytyi sekä kunnanhallituksen että kunnanvaltuuston

pöytäkirjat kuluvalta vuodelta. Neljältä kunnalta löytyi sekä hallituksen että valtuuston pöytäkirjat vuodesta 1999 lähtien. Useimpien kuntien pöytäkirjoihin oli linkki heti etusivulta.

Lautakuntien kokousten pöytäkirjojen näkyvyys oli myös kaikilla kunnilla hyvä, tosin yksi kunta ei näyttänyt jäsenten tietoja lainkaan.

3.5. Elinkeino

Elinkeinoasiat	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
	1	1	4	6
Matkailu	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
	0	0	2	10
Yritykset	Ei tietoa	Yhteystiedot	Yhteystiedot ja pari lausetta	Yhteystiedot ja yksityiskohtainen esittely
		1	4	7

Taulukko 6. Elinkeino

Elinkeinotoiminnan esittely oli useimpien kuntien kohdalla liitetty joko yritykset -yläotsikon alle tai sitten löytyi yhteystiedot elinkeinoasiamiehestä. Kuudella paikkakunnalla oli tarkat tiedot elinkeinorakenteesta, suurimmista työllistäjistä ja yhteystiedot.

Kaikilta kunnilta löytyi tietoa matkailusta. Sivut käsittivät ainakin nähtävyydet ja majoitustoiminnan. Nähtävyyksien osalta kaikilla oli kuvaus kohteesta ja aukioloajat ja vuosittaiset perinnetapahtumat oli kuvattu tarkasti. Niiden kuntien, jotka ovat panostaneet matkailuun, matkailusivut löytyivät erikseen, eikä liitettynä kulttuuriosioon tai tapahtumakalenteriin. Isoimmilla kunnilla löytyi erikseen matkailusivusto tai palvelupiste, jossa kuvattiin erittäin tarkasti nähtävyydet, majoitustoiminta, ostosmahdollisuudet.

Yritystietojen esittelyä eivät kaikki kunnat nähneet tarpeellisena esittää. Asukasluvultaan pienemmät kunnat esittivät toimialahakemiston, jossa kerrottiin yrityksen yhteystiedot. Isommat kunnat siirsivät sivun erilaisille kehittämisskeskuksien ja yrityshautomoiden ylläpitämille sivuille.

4. Tulosten analysointi

Suurin osa kuntien verkkosivujen sisältämistä palveluista on tiedottavaa ja näin ollen ne kuuluvat ensimmäisessä kehitysvaiheessa oleviin verkkosivuihin. Kunnan verkkosivut toimivat tällöin tiedottavana verkkopalveluna, jolloin ne pääosassa ovat yksisuuntaista viestintää. Toisen vaiheen verkkosivustot ovat enemmän kaksisuuntaisia, jolloin asiointi asiakas- viranomaisen akselilla toimii jo laajemmassa mittakaavassa. [Sisäasiainministeriön verkkolomakeryhmä, 2000].

Tutkimuksessani suurin osa kunnista kuului vielä ensimmäisen kehitysvaiheen verkkosivuihin. Ilahduttavaa oli, että jopa neljä kuntaa ylsi toiseen kehitysvaiheeseen tarjoten todellisia vuorovaikutusmahdollisuuksia.

Julkisen hallinnon tietohallinnon neuvottelukunta asettaa omia kriteeristöjään kuntien verkkosivujen rakentamiselle, esitystavoille, sisällön julkaisulle ja tekniikalle. Tämän tutkimuksen osalta kiinnostavinta näiden kriteereiden osalta oli palautteenantomahdollisuus, joka neuvottelukunnan mukaan tulisi olla joka kunnalla ja valmiin lomakkeen muodossa, ei pelkällä sähköpostiyhteydellä varustettuna. Sähköpostia ei jokaisella kansalaisella vielä ole, joten sen puuttuminen ei saa olla palautteenannon esteenä. [Julkisen hallinnon tietohallinnon neuvottelukunta, 2000].

Tutkimuksessani ainoastaan yksi kunta antoi ainoastaan sähköpostiosoitteen palautteenantoa varten, joten tässä osin sääntöjä noudatettiin. Yllättävää oli, että kyseessä oli suurempi kunta, jolla sivuston taso muuten oli korkea.

Palveluista parhaimmat yhteispisteet sai kirjasto ja perhepäivähoito. Molemmissa palveluissa tarjottiin sähköisiä asiakirjoja, joita pystyi myös lähettämään sähköisesti. Sähköisiä asiakirjoja oli muutenkin tarjolla osalla kunnista kymmenittäin, mutta todellinen verkkoasiointi jäi puuttumaan, sillä lomakkeiden lähetyksmahdollisuus oli puutteellista. Muutama kunta tarjosi lastenhoitopaikkahakemusta sähköisesti, jolloin asiakas pystyi täyttämään lomakkeen koneellaan ja lähettämään lomakkeen samalla kertaa. Kirjastot ovat usein yhdistäneet tietokantansa seudullisesti, mikä mahdollistaa sen, että palveluun pystytään panostamaan. Ainoastaan yhdellä kunnalla oli oma tietokantansa, mutta sekin tuotettu kaupallisesti.

Teknisen toiminnan tiedottaminen oli kauttaaltaan puutteellista. Tämä johtuu ehkä siitä, ettei sitä nähdä kuntalaisten arjessa niinkään ensisijaisena tiedon tarpeena. Sähköisiä lomakkeita oli tarjolla myös tämän alueen alla, rakennuslupahakemuksia tarjottiin sähköisesti, samoin vesimittari-lukeman ilmoitusta ja sen kulutuksenseurantaa.

Matkailuun panostavat kunnan olivat sijoittaneet matkailu -otsikon heti etusivulle ja esitelleet laajasti nähtävyyksiä ja majoitustoimintaa. Matkailu sekoitettiin usein kulttuuriin ja se vaikeutti pisteytystä. Varsinkin pienten kuntien kulttuuritarjonta on vähäistä omien teatterisalien ja taidenäyttelyiden puuttuessa.

Asukasluvultaan pienemmät kunnat esittelivät kunnan alueella toimivia yrityksiä hyvin. Toimiala-hakemiston alta löytyi tiedot yrityksistä. Suurimmat kunnat tyytyivät esittelemään vain suurimmat työnantajat. Tämä saattaa johtua siitä, että isoimmilla kunnilla yrityshakemisto saattaisi muistuttaa tietokantaa, jolloin sen ylläpitäminen, alati muuttuvien tietojen suhteen, olisi työlästä.

4.1. Erityispiirteitä

Kuntien sivujen yleisilme on tiedottava. Kovinkaan erilaisia vuorovaikutuskeinoja ei ole sivuilla käytetty. Perinteisimpiä vuorovaikutustapoja verkkosivuilla ovat palautelomake tai keskustelualue. Kaikilta kunnilta löytyi palautelomake tai palautteeseen kehottava teksti ja sähköpostiosoite. Keskustelupalsta oli käytössä ainoastaan yhdessä kunnassa ja toisessa sen kokeilu oli lopetettu 25.2.2003.

Kielitarjonta oli myös vähäistä, kahdeksan kuntaa kahdestatoista tarjosi kielivaihtoehtona vain suomea. Neljällä oli englanninkielinen tiedotussivu ja yhdellä tiedotussivut olivat myös saksaksi, ranskaksi, ruotsiksi ja venäjäksi. Suomi on kuitenkin kaksikielinen maa, joten kielitarjonnan supistuminen vain suomeen ja Englantiin ihmetytti. Matkailun edistämisen kannalta englannin kielellä tarjottavat majoitus- ja tapahtumapalvelut olisi enemmän kuin toivottavaa. Yleisestikin sivuilta olisi toivonut löytävän perustiedot ja palvelut suomeksi, ruotsiksi ja englanniksi.

Tapahtumakalenteri löytyi kymmeneltä kunnalta. Osa kunnista oli liittänyt omat tapahtumatiedoitteet paikallisten palveluporttien tapahtumatarjontaan. Tapahtumakalenterin helppokäyttöisyys sekä löydettävyys olisi käyttäjien ja matkailun edistämisen kannalta äärimmäisen tärkeää, varsinkin pienten kuntien kohdalta, jossa suuret tapahtumat toteutuvat vuosi vuodelta samaan aikaan ja samassa paikassa, tällöin tieto näistä olisi äärimmäisen toivottavaa.

4.2. Sähköinen asiointi

Sähköisen asioinnin tarjonnassa oli isommilla kunnilla valikoimaa. Lomakkeita löytyi kymmenittäin ja palautusmahdollisuus sähköisesti oli myös joissain lomakkeissa mahdollista. Kolmella tutkittavista kunnista ei ollut tarjota minkäänlaista sähköistä lomaketta kuntalaisille.

Sähköisen asioinnin vähäinen mahdollisuus johtuu siitä, että laki sähköisestä asioinnista hallinnossa on tuore. Se on tullut voimaan vuoden 2000

alusta ja laki on vasta mahdollistanut sähköisen asioinnin kehittämisen ja laillistanut sähköiset lomakkeet. Myös EU asettaa sähköiselle asioinnille sekä paineita että velvoitteita. EUn asettaman työryhmän, eEuropen, tavoitteena on mahdollistaa julkinen asiointi jokaiselle jäsenmaansa kansalaiselle. EUn määräämä direktiivi sähköisestä allekirjoituksesta tuli voimaan 13.1.2003. [Sisäasiainministeriön verkkolomakeryhmä, 2000].

Toinen sähköisen asioinnin edistämistä vaikeuttava asia on ollut asioijan tunnistaminen ja tietoturvasta huolehtiminen. [Sisäasiainministeriön verkkolomakeryhmä, 2000]. Perinteinen lomake tunnistetaan allekirjoituksella, mutta sähköinen asiointi vaatii oman tunnistemuodon. Henkilötunnistekortin käyttöasteen nouseminen ja laitepäätteiden hinnan halpeneminen voi tuoda lisää sähköisiä palveluita, tällöin viranomainen tunnistaa asioijan kortin tietojen perusteella. Kunnat ovat esittäneet laajaa kiinnostusta henkilön sähköistä tunnistamista (HST) kohtaan tulevaisuudessa ja palveluita suunnitellaan laajasti. Verkkodemokratia, mahdollisuus äänestää ja vaikuttaa verkossa, on vasta ideointi-asteella. [Sisäasiainministeriö, Julkisen verkkoasioinnin kehittämishanke 1/2001].

4.3. Seutuportaalien vaikutus

Seudullisesti yhteistyötä tekevät kunnat tekevät yhteistyötä myös verkossa. Verkossa näkyvää seudullista yhteistyötä kutsutaan portaaliksi eli palveluportiksi. Suurin osa kuntien palveluista on poikkihallinnollista, jolloin sama palvelu joudutaan toteuttamaan jokaisessa kunnassa. Kuntien yhteiset palvelut esitetään myös verkossa yhteisesti, eikä jokainen kunta esitä samaa asiaa omilla sivuillaan.

Kunnat hyötyvät seudullisista palveluporteista yhteisten tietokantojen ja sähköisten palvelujen kautta. Jokaisen kunnan ei tarvitse tehdä ja ylläpitää itse tietokantaansa, vaan usein se on ostettu kaupallinen palvelu, jota kunta voi käyttää liittämällä sen omiin sivuihinsa. Asiakasta ei näin ollen tarvitse juoksuttaa luukulta toiselle myöskään verkossa, vaan asiakas saa parhaimman mahdollisen palvelun samasta osoitteesta eli oman kuntansa sivuilta. [Sisäasiainministeriön verkkopalveluhanke (HAAVE), 2001].

Kaikki tutkimani kunnat ovat joko Aina -palveluportin tai Hyrinetin jäseniä. Aina -palveluportin kuuluvat kunnat Hattula, Hauho, Hämeenlinna, Janakkala, Kalvola, Lammi, Tuulos. Hyrinet -palveluporttiin kuuluvat kunnat Hausjärvi, Hyvinkää, Riihimäki ja Loppi. Näkyvimpiä seudullisia palveluita ovat kirjasto ja terveydenhoitopalvelut, joissa jokainen kunta teki yhteistyötä vähintään yhden kunnan kanssa.

5. Yhteenveto

Kuntien verkkosivustojen suurimpana ongelma on yhtenäisyys sekä itse sisällön esitystavassa, sisällön tärkeyden määrittelyssä että sähköisten asiakirjojen tarjonnassa. Kuntien verkkosivujen tuottamisessa suurimpana ongelmana on taloudellisten resurssien puute ja henkilöstön osaamattomuus. Tämä näkyy sekavana tarjontana sivustoilla. Vertailua ei helpottanut yhteisten nimikkeiden puuttuminen ja esitystavan epäselkeys. Kuntien täytyisi pystyä tiivistämään seudullista yhteistyötä entistäkin tehokkaammin. Varsinkin asukasluvultaan pienet kunnat hyötyvät tämänkaltaisesta yhteistyöstä, samalla palvelujen tasoa ja määrää pystyttäisiin kohottamaan, jolloin myös yhtenäisyys säilyisi ja tiedon löytyminen sivuilta helpottuisi. Suurimpien kuntien palveluiden taso on vähintäänkin hyvää tasoa, mutta palvelujen määrää ei saisi esittää sivujen käytettävyyden kustannuksella. Usein palveluiden löytäminen suurten kuntien sivuilta oli vaikeaa. Vaikka tutkimukseni ei varsinaisesti käsitellyt sivujen käytettävyyttä, tulee käytettävyys esiin väistämättä sivuja selatessa. Kuntien tulevaisuuden tavoitteena on tiedon helppo löydettävyys, eikä se ainakaan tämän tutkimuksen perusteella tuntunut toteutuvan suurten kuntien sivuilla [Julkisen verkkoasioinnin kehittämishanke, 7/2000].

6. Loppupäätelmiä

Tutkimus on hyvin pintaa raapaiseva ja syvempi analysointi puuttuu, johtuen tutkimukselle asetetuista ajallisista ja määrällisistä rajoitteista. Tutkimukseen olisi hyvä liittää sekä käytettävyyden arviointia että tutkia käyttäjäastetta, jotta kunnille tulisi tietoa, kuka heidän sivujaan käyttää ja mitä palveluja asiakkaat todella sivuilta käyttävät. Tutkimuksessa olisi hyvä haastatella myös kuntien verkkosivuista vastaavia henkilöitä ja tuoda heidän näkemystään verkkosivujen rakentamisesta, tekemisestä sekä sisällön tuottamisesta. Tutkimukseen sisällytettäviä asioita olisi myös erilaisten tahojen luomien kriteeristöjen mukaiset arvioinnit, ja lisäksi tulisi selvittää täyttävätkö sivustot näiden asettamat viitekehukset.

Tutkimukseni kuvaa lähinnä perustiedottamisen tasoa, eikä tässä tutkimuksessa paneuduttu erityispalveluihin eikä tehty vertailua erityispalveluiden toimivuudesta. Tämän tyyppinen tutkimus olisi hyvä liittää tutkimukseeni.

Viiteluettelo

- [Julkisen verkkoasioinnin kehittämishanke, 1/2000] Julkisen verkkoasioinnin kehittämishanke. Kohti tietoyhteiskuntaa - julkisten verkkopalvelujen kehittämisen lähtökohdat. Sisäasiainministeriön julkaisusarja 1/2000.
- [Julkisen verkkoasioinnin kehittämishanke, 7/2000] Julkisen verkkoasioinnin kehittämishanke. Ensiaskleet - verkkopalveluiden kehittäminen ja tulevaisuus julkisessa hallinnossa keväällä 2000. Sisäasiainministeriön julkaisusarja 7/2000.
- [Julkisen verkkoasioinnin kehittämishanke, 2001] Julkisen verkkoasioinnin kehittämishanke. Julkisten verkkopalveluiden kehittäminen ja XML. Julkisen verkkoasioinnin kehittämishankkeen julkaisuja 5/2001. Oy Edita Ab, Helsinki 2002.
- [Julkisen verkkoasioinnin kehittämishanke, 3/2002] Julkisen verkkoasioinnin kehittämishanke. Juna vislas jo pois! JUNA-hankkeen matkaraportti 1.9.1999 - 31.12.2001. Sisäasiainministeriön julkaisusarja, 3/2002.
- [Ruusula M., 2001] Matti Ruusula. Palveleeko verkko kuntalaisia? Suomen kuntien verkkosivustojen laadullinen vertailu 2001. Julkisen verkkoasioinnin kehittämishanke (JUNA), Oy Edita AB, Helsinki 2001.
- [Saarijärvi M., 2001] Saarijärvi Marjukka. Asiakaspäätteillä julkisiin verkkopalveluihin. Asiakaspäätteet ja kansalaisten verkkoasiointi - hankkeen loppuraportti. Sisäasiainministeriö, Julkisen verkkoasioinnin kehittämishankkeen julkaisuja 6/2001. Oy Edita Ab, Helsinki 2002.
- [Sisäasiainministeriö, Julkisen verkkoasioinnin kehittämishanke 1/2001] Sisäasiainministeriö, Julkisen verkkoasioinnin kehittämishanke Kohti verkkoasiointia ja e-hallintoa. Ohjeita ja neuvoja verkkopalvelujen kehittäjille. Sisäasiainministeriö, Julkisen verkkoasioinnin kehittämishankkeen julkaisuja 1/2001 .
- [Sisäasiainministeriön verkkolomakeryhmä, 2000] Sisäasiainministeriön verkkolomakeryhmä. Kohti uutta verkkolomakepalvelua. Sisäasiainministeriön julkaisusarja 8/2000.
- [Sisäasiainministeriön verkkopalveluhanke (HAAVE), 2001] Sisäasiainministeriön verkkopalveluhanke (HAAVE). Sisäasiainministeriön hallinnonalan verkkoasioinnin kehittämisohjelma 2002–2005. Sisäasiainministeriön julkaisusarja 14/2001.
- [Julkisen hallinnon tietohallinnon neuvottelukunta, 2000] Julkisen hallinnon tietohallinnon neuvottelukunta. JHS 129 Julkishallinnon www-sivuston suunnittelun ohjeet 2003. Sisäasiainministeriö 2000.

Rinnakkaisuuden toteutus Symbian OS käyttöjärjestelmässä

Liisa Supponen

Tiivistelmä.

Tämä dokumentti käsittelee rinnakkaisohjelmoinnin toteutusmahdollisuuksia Symbian OS käyttöjärjestelmäalustalla. Dokumentissa tutustutaan käyttöjärjestelmän säikeiden ja säikeet osittain korvaavien aktiiviolioiden toimintaan sekä esitellään esimerkitapauksia niiden käytöstä ja käyttöön liittyvistä ongelmatilanteista rinnakkaisohjelmien yleisten käsitteiden pohjalta.

Avainsanat ja -sanonnat: Rinnakkaisohjelmointi, Symbian OS, säie, aktiiviolio
CR-luokat: A.2

1. Johdanto

Nykyajan tietoyhteiskunnassa matkapuhelimet nousevat jatkuvasti entistä tärkeämpään asemaan jokapäiväisessä elämässämme. Taskussamme oleva laite ei olekaan enää pelkkä puhelin vaan sillä voi myös pelata, maksaa laskuja ja tehdä asioita, jotka olemme yleensä mieltäneet mahdollisiksi vain kookkailla pöytäkoneilla. Matkapuhelimesta on muodostunut pieni tietokone, jolle kehitetyt ohjelmistot lähenevät vaatimuksiltaan muista ohjelmointiympäristöistä tuttuja ohjelmistoja.

Ohjelmistovaatimusten kasvaessa rinnakkaisohjelmoinnista on samanaikaista suoritusta tukevana ohjelmointimenetelmänä kasvanut merkittävä osa nykyaikaista ohjelmistokehitystä. Jotta nykyaikaisten ohjelmistojen vaatimukset matkapuhelimilla täytyisivät, on rinnakkaisohjelmointi täytynyt ottaa huomioon myös matkapuhelinten käyttöjärjestelmiä kehitettäessä.

Symbian käyttöjärjestelmä on erityisesti mobiilien päätelaitteiden tarpeisiin suunniteltu käyttöjärjestelmä, joka tukee nykyaikaista ohjelmointia eri muodoissaan. Rinnakkaisohjelmoinnin tarve on otettu laitteessa myös huomioon. Rinnakkaisohjelmoinnin toteuttaminen Symbian OS käyttöjärjestelmää käyttävillä laitteilla on kuitenkin vaatinut erityisiä, muista ohjelmistoalustoista poikkeavia ratkaisuja laitteiden erityispiirteiden, kuten suorituskykyresurssien vähyyden ja laitteiden jatkuvan käytön vuoksi. Tästä johtuen perinteisten säikeiden rinnalle on kehitetty toinen, rinnakkaisen kaltainen ohjelmointimenetelmä aktiivioliot, joka on säikeitä kevyempi tapa suoriutua rinnakkaisohjelmoinnista.

2. Symbian OS

2.1. Yleistä

Symbian OS käyttöjärjestelmä on suunniteltu ja kehitetty alati kehittyvien matkapuhelinten tarpeet sekä erityispiirteet huomioon ottaen. Vaikkakin matkapuhelimet muistuttavat entistä enemmän ominaisuuksiltaan kannettavia tietokoneita sekä PDA-laitteita niiden ominaispiirteet, kuten suorituskykyresurssien vähyys sekä muista laitteista poikkeava käyttötapa ovat vaatineet oman käyttöjärjestelmän kehittämisen.

Symbian OS on avoimien standardien käyttöjärjestelmä joka on maailman suurimpien matkapuhelinvalmistajien, kuten Nokian sekä Sony Ericssonin, yhteisomistuksessa [Mery, 2003]. Ensimmäinen täysin avoimella Symbian OS käyttöjärjestelmällä varustettu matkapuhelin oli Nokian 9210 ja se julkaistiin vuonna 2001 [Symbian, 2003].

2.2. Erikoispiirteitä

Vaikkakin matkapuhelimet lähestyvät ominaisuuksiltaan kannettavia tietokoneita sekä PDA-laitteita on niille ohjelmoitaessa otettava huomioon matkapuhelimiin liittyviä erityisominaisuuksia, kuten saavutettavuuden vuoksi vaadittava jatkuva päällä olo tai vähäiset muisti- ja virtaresurssit. Rinnakkaisohjelmoinnin toteutuksen ja suunnittelun kannalta merkittävin erityisvaatimus näistä on vähäiset virtaresurssit, joita yleensä raskaana ominaisuutena pidetty rinnakkaisuus ei saa kuluttaa liikaa [Mery, 2003].

2.3. Rinnakkaisohjelmoinnin tarve Symbian OS käyttöjärjestelmässä

Matkapuhelimet toimivat pääasiassa puhelinkoneina, mutta nykyaikaisten kommunikaatiotarpeiden lisääntyessä myös matkapuhelinten tietoliikennevaatimukset kasvavat mikä tuo tarpeen rinnakkaisuudelle esimerkiksi luodessa bluetooth- tai gprs-yhteyksiä soketteja käyttäen. [oma] Lisävaatimuksen rinnakkaisuudelle tuo se käyttömukavuuden mahdollistava vaatimus, että samanaikaisesti tietoliikenneyhteyksien kanssa on myös pystyttävä käyttämään puhelinta muihin toimintoihin. Matkapuhelimen on vastattava käyttäjän pyyntöihin ja näppäimen painalluksiin vaikka olisikin suorittamassa jotain pitkäkestoista toimintoa, kuten avaamassa yhteyttä palvelimeen, lukemassa tiedostoa tai suorittamassa laskutoimitusta [Tasker, 1999].

3. Ohjelman suoritus ja prosessit

3.1. Yleistä

Symbian OS on käyttöjärjestelmä, jonka ydin mahdollistaa moniajon. Käyttöjärjestelmä tukee monisäikeisyyttä, mikä itsessään mahdollistaa rinnakkaisuuden käyttöjärjestelmässä. Käyttöjärjestelmä on kerroksittainen ja sen ylin kerros tarjoaa tuen sovelluskehitykselle. Sovellukset ajetaan virtuaalikoneympäristössä, jokaisen sovelluksen muodostuessa yhden tai useamman säikeen ryhmittymien muodostamista prosesseista [Sanders, 2002]. Kukin prosessi jakaa tietyn muistialueen siihen kuuluvien säikeiden kanssa [Koskimies, 2000]. Prosessit itsessään ovat ytimen olioita joihin niitä käyttävät sovellukset pääsevät käsiksi käsittelyrajapintojen RProcess kautta. Prosessitasoa käsittelevät yleensä ainoastaan järjestelmäsovellukset, tavallisten sovellusten ei tarvitse niitä muuttaa tai käsitellä toimiakseen [Symbian, 2002]. Sovelluksen varsinaiset resurssit ovat Symbian OS käyttöjärjestelmässä prosessin sijaan säikeen hallinnassa, mikä mahdollistaa aktiiviolioiden tehokkaamman käytön. [Tasker, 1999]

3.2. Prosessien hallinta

Prosessia luotaessa pelkkä rakennin luo ainoastaan käsittelyrajapinnan prosessiin, varsinainen ytimen olio luodaan erillisellä metodilla. Luomismetodi, Create, lataa prosessin annetun asennustiedoston perusteella ja luo prosessiin yhden säikeen, joka toimii prosessin pääsäikeenä. Luodun prosessin varsinainen omistaja on se prosessi, josta uuden prosessin luomiskomennot annettiin. Omistusoikeus on kuitenkin siirrettävissä. Käsittelyrajapinta voidaan avata myös jo olemassa oleviin prosesseihin prosessien tietojen perusteella.

Käynnistettäessä prosessia sen pääsäie käynnistetään ensimmäisenä pääsäikeen prioriteetti-arvosta suhteessa muihin prosessin säikeisiin huolimatta. Prosessi voidaan lopettaa suoraan tai virheilmoituksen kera milloin tahansa. Prosessin lopettajalla ei ole merkitystä ja prosessi voi lopettaa halutessaan myös itse itsensä. Kun prosessia ei enää tarvita, prosessi on suljettava Close metodin avulla. Prosessin ollessa järjestelmäolio sitä ei luoda muistiin ja koska sisältää myös vain järjestelmän perustyyppisiä, joten sitä ei tarvitse tuhota ohjelmallisesti. [Symbian, 2002]

3.3. Prosessien prioriteetit

Prosessille voidaan antaa prioriteetti, eli arvo, jonka perusteella prosessi saa suoritusoikeutta käyttöjärjestelmän ytimeltä. Prioriteetti voidaan määrittää missä tahansa vaiheessa prosessin luomista tai käyttöä. Prioriteetille on kahdeksan mahdollista arvoa, joista neljä korkeinta on tarkoitettu ainoastaan

käyttöjärjestelmän omaan käyttöön, kuten laitteen kellonajan ylläpitoon sekä ytimen valvontatarkoituksiin. Erillisten sovellusten on mahdollista käyttää neljää alemman tason prioriteettia, joista löytyy erillinen arvo muun muassa sovelluksen taustalla pyöriville prosesseille. Prosessien prioriteettia on mahdollista muuttaa myös niihin kuuluvien säikeiden kautta. [Symbian, 2002]

4. Säikeet

4.1. Yleistä

Säikeet ovat yksi rinnakkaisohjelmoinnin peruskäsitteistä ja usein rinnakkaisohjelmointi käsitetäänkin pitkälti säikeillä ohjelmointina. Säikeet ovat rinnakkaisia suoritusyksiköitä, jotka Symbian OS käyttöjärjestelmässä ovat prosessien osana olevia ytimen olioita, joita itsessään kutsutaan usein keveiksi prosesseiksi. Säikeiden aikataulutusta on Symbian OS käyttöjärjestelmässä suoritusturvattu eli kaikki säikeet saavat suoritusvuoron kohtuullisen ajan kuluessa. Suoritusvuorossa oleva säie voidaan siis keskeyttää, jotta jokin toinen säie saisi vuorostaan suoritusoikeuden. Säikeiden käsittely tapahtuu prosessien lailla käsittelyrajapinnan RThread kautta. Kunkin säikeen käsittelyrajapinta on oletusarvoisesti sen prosessin hallinnassa, johon säie kuuluu. Symbian OS käyttöjärjestelmän säikeet poikkeavat hyvin vähän säikeistä muilla ohjelmistoalustoilla. [Symbian, 2002].

4.2. Säikeiden hallinta

Säiettä luotaessa rakennin luo ainoastaan käsittelyrajapinnan RThread säikeeseen, varsinainen ytimen olio luodaan erillisellä metodilla. Luomismetodi, Create, luo ytimen säie olion ja lisää sen luomishetkellä käytössä olevaan prosessiin tai prosessiin joka on määritelty luomismetodin parametrina. Oletusarvoisesti luotu säie on luomishetkellä käytössä olevan prosessin omistuksessa, mutta omistusoikeus voidaan siirtää myös säikeelle, josta uuden säikeen luomiskomento annettiin. Luomisvaiheessa säikeelle luodaan keko joka jakaa sitä osaa järjestelmän muistista, jossa myös ohjelmapino sijaitsee. Keon koko määritellään luomismetodin parametrina. Säikeelle on myös mahdollista antaa käsittelyrajapinta jo luotuun kekkoon kuten esimerkiksi sen säikeen kekkoon, joka uutta säiettä on luomassa. Tarvittaessa säikeelle voidaan luomisvaiheessa välittää myös muuta dataa, kuten sovellusmuuttujia, parametrina.

Säikeen suoritus ei ala välittömästi säikeen luomisesta vaan välittömästi luomisen jälkeen säie on keskeytetyssä tilassa, jotta säikeen suoritusparametreja voisi muuttaa vielä ennen säikeen käynnistämistä. Säie on käynnistettävä

erikseen käyttämällä jatka, eli Resume metodia. Säie voidaan asettaa keskeytetty tilaan myös eksplisiittisesti käyttämällä keskeytä, eli Suspend metodia, joka keskeyttää säikeen suorituksen. Suspend ei pääätä säiettä kokonaan vaan pysäyttää sen väliaikaisesti. Keskeytetystä tilasta säikeen voi käynnistää uudelleen jatkamalla sen suoritusta jälleen Resume-metodia käyttäen.

Säikeen voi päättää säikeen ulkopuolelta joko virheilmoituksen kera tai ilman käyttäen joko Kill- tai Panic-metodeita, joista jälkimmäiselle annetaan virheparametri ilmaiseen tapahtunut poikkeus ohjelman kulussa. Säie itse voi päättää suorituksensa käyttämällä staattista järjestelmämetodia Exit. Kun säiettä ei enää tarvita, se on suljettava Close metodilla. Säiettä ei kuitenkaan tarvitse erikseen tuhota ohjelmallisesti, koska se on ytimen olio eikä sitä siten luoda muistiin.

Käsittelyrajapinta säikeeseen on mahdollista avata myös jo olemassa olevaan säikeeseen Open metodin avulla antamalla joko avattavan säikeen nimi, tunniste tai etsimällä sitä TFindThread järjestelmätyypin avulla, joka etsii säikeitä muun muassa niiden nimen osan perusteella. Käsittelyrajapinnan avaamisen jälkeen säiettä voi hallita kokonaan luodun säikeen tapaan. [Symbian, 2002]

4.3. Säikeiden prioriteetit

Jokaisella prosessin säikeellä on prioriteettiarvo, jonka perusteella aikataulutetaan ajettavaksi säie, jolla on korkein prioriteetti ja joka on valmis suoritettavaksi. Prioriteetiltaan samanarvoisten säikeiden aikataulutus tapahtuu jakamalla kullekin säikeelle aikajakso niin, että kaikki säikeet saavat suoritusvuoron jossain vaiheessa.

Säikeiden prioriteetti on oletusarvoisesti prioriteettien keskiarvo. Koska säie täytyy käynnistää erikseen, säikeen prioriteettiarvo on mahdollista muuttaa sen luomisen jälkeen. Säikeille ei voi asettaa tyhjää prioriteettiarvoa. Ytimen itsensä prioriteettiarvo on aina yksittäisiä säikeitä suurempi, jotta keon varaus ja vapauttamistoiminnot toimisivat jatkuvasti. Prioriteettiarvon voi asettaa joko suhteellisena tai absoluuttisena. Käytettäessä suhteellista prioriteettia säikeen todellinen prioriteetti riippuu myös säikeen omistavan prosessin prioriteettiarvosta suhteessa muihin prosesseihin. Absoluuttisen prioriteettiarvon tapauksessa asetettu prioriteetti on myös säikeen todellinen prioriteetti.

Säikeen käsittelyrajapinnan kautta myös prosessin prioriteettiarvon muuttaminen on mahdollista [Symbian, 2002].

4.4. Ongelmatilanteet

Säikeen käyttöön liittyvistä ongelmista Symbian OS käyttöjärjestelmässä merkittävin on säikeiden ja vaatimat käyttöjärjestelmäresurssit. Laitteissa, joissa prosessoritehot ja muistin määrä ovat alhaisia joko sen takia, ettei näitä ole mahdollista asentaa mittavissa määrin tai sen takia, etteivät suuret prosessoritehot söisi patterilla toimivien laitteiden vähäisiä virtaresursseja, onkin syytä välttää säikeiden ylimääräistä käyttöä ja korvata ne vähäisempiä resursseja vaativin keinoin, kuten esimerkiksi aktiiviolioiden avulla. Symbian OS ei varsinaisesti kiellä säikeiden käyttöä, mutta niihin turvautumista suositellaan vain jos muita ratkaisuja ei löydy [Digia, 2003].

Rinnakkaisohjelmoinnin yksi yleisemmistä ongelmista on säikeiden käyttöön liittyvä muistin jako, joka on ongelma myös Symbian OS käyttöjärjestelmässä. Yhden sanan, joka Symbian OS käyttöjärjestelmää käyttävissä laitteissa on suositusten mukaan 32-bittinen, kirjoittaminen kaikilla laitteistoarkkitehtuureilla on aina atominen eli sitä ei voida keskeyttää kesken kirjoituksen. Laajempien alueiden käyttö, lukeminen tai kirjoittaminen ei kuitenkaan ole millään alustalla atomista. Koska yhden prosessin säikeet käyttävät samaa aluetta järjestelmän muistista, on säikeitä käytettäessä varmistuttava siitä, että muistiin tehtävät operaatiot ovat ohjelmallisesti tehty atomisiksi esimerkiksi lukoin niin, että ne suoritetaan loppuun ennen kuin säikeen suorituksen voi keskeyttää. Ilman atomisuutta säikeen suoritusvarmistetun luonteen vuoksi on mahdollista, että säikeen suoritus keskeytetään esimerkiksi kesken laajahkon muistialueen kirjoitusoperaation ollessa käynnissä ja seuraavan suoritusvuoron saa säie, jonka tarkoitus on lukea juuri tältä muistialueelta, jolloin säie lukee virheellistä dataa, koska sen kirjoitus oli jäänyt kesken [Sanders, 2002] [Tasker, 1999].

Rinnakkaisohjelmoinnissa helposti vastaantuleva ongelma on myös niin sanottu näännytys, jota esiintyy myös Symbian OS käyttöjärjestelmässä, koska siinä on useimpien käyttöjärjestelmien tavoin mahdollistettu säikeiden priorisointi, joka on näännytyksen useimmin aiheuttava tekijä. Näännytyksen tapahtuessa jokin korkea prioriteettinen säie saa niin paljon prosessointiaikaa, että jonkin toisen säikeen tai säikeiden suoritus ei saa laisinkaan suoritusaikaa tai saa sitä niin harvoin, että sen toiminta häiriintyy. Myös liiallinen prosessointiaika yhdelle säikeelle voi olla ongelma kyseisen säikeen kannalla jolloin kyseessä on uuvutus, eli säie saa toimintansa kannalta liikaa suoritusaikaa. Sekä uuvutus, että näännytys ovat ongelmina ratkaistavissa yleensä oikeanlaisella ja tarkoitukseen sopivalla priorisoinnilla. [Magee and Kramer, 1999] Yleensä on järkevää asettaa käyttäjän kanssa suorassa vuorovaikutuksessa olevat säikeet sekä esimerkiksi ajastimet korkealle

prioriteetille ja vuorostaan laskutoimitukset sekä tietoliikenneyhteyksiä käsittelevät matalammalle prioriteetille, jotta ohjelmistot vastaisivat käyttäjän pyyntöihin säännöllisessä ajassa. [Tasker, 1999]

4.5. Atomisuuden toteutus

Yleisin tapa atomisuuden hallintaan on lukot joihin Symbian OS käyttöjärjestelmässä tarjotaan käsittelyrajapinta RMutex. Lukko suljetaan ennen sen säikeessä olevan operaation aloitusta, joka vaatii atomisuutta. Lukko vapautetaan operaation suorituksen jälkeen. Tällöin säikeen suoritusta ei voi keskeyttää suoritusvarmistuksen puitteissa kesken operaation ja toinen säie ei pääse operaation käyttämiin muistiresursseihin käsiksi ennen kuin lukon sulkenut säie on avannut lukon. Lukkojen käyttö itsessään voi aiheuttaa ongelmia muun muassa tilanteissa, jossa kaksi säiettä odottaa toistensa sulkemien lukkojen avautumista [Tasker, 1999].

5. Aktiivioliot

5.1. Yleistä

Symbian OS käyttöjärjestelmää käyttävien laitteiden resurssipulasta johtuen rinnakkaisohjelmoinnin perustoteutustavan eli säikeiden käyttö on pääasiallisesti liian raskas tapa suoriutua rinnakkaisuudesta. Jotta sovellukset kuitenkin vastaisivat käyttäjiensä pyyntöihin, on tarve suoritusvarmistetulle rinnakkaisuudelle perusteltua. Symbian OS käyttöjärjestelmän olioperustaisen luonteen sekä säikeiden pohjalta säikeille ja prosesseille on kehitetty kevyempi muoto, aktiivioliot, joiden ansiosta sovelluksissa tarvitaan yleensä ainoastaan prosessin pääsäie. Symbian OS käyttöjärjestelmän aktiivioliot eivät ole käyttöjärjestelmän varsinaisesta ytimeistä riippuvaisia ja niiden hallitseminen on mahdollista tavallisin ohjelmallisin komennoin. Keveydestään huolimatta aktiivioliot ovat tarpeeksi tehokkaita käyttöjärjestelmän vaatimuksia ajatellen [Tasker, 1999].

Yleensä aktiiviolioilla käsitetään ohjelmointikielissä olioiksi muodostetut järjestelmäsäikeet. Symbian OS käyttöjärjestelmässä tätä käsitettä vastaavat paremmin varsinaiset säikeet aktiiviobjektien ollessa erillinen, yhden säikeen varassa toimiva rinnakkaisen kaltainen kokonaisuus. Symbian OS käyttöjärjestelmässä aktiiviobjektit ovat kuin pienimuotoisia säikeitä säikeen itsensä toimiessa pienimuotoisena käyttöjärjestelmäytimenä joka aikatauluttaa aktiiviobjekteja. Aktiiviobjektit eivät ole säikeiden tavoin suoritusvarmistettuja, eli kaikille aktiiviobjekteille ei taata suoritusvuoroa eikä aktiiviobjektia voida keskeyttää kesken sen kriittisten toimenpiteiden [Koskimies, 2000][Tasker, 1999].

Aktiiviobjektit on suunniteltu pääasiassa sovellusten tapahtumankäsittelijöiksi. Tapahtumankäsittely perustuu yleensä sovelluksille, jotka esittävät pyynnön joka myöhemmin toteutuu, kuten esimerkiksi jonkun tietyn näppäimen painalluksen odottaminen. Aktiiviobjektien toiminta perustuu näille pyynnöille ja niiden täyttymisen käsittelylle. Yksittäinen aktiiviobjekti on vastuussa yhdestä pyynnöstä ja sen käsittelystä.

Aktiiviolioiden pyynnot ovat pääasiassa Symbian OS käyttöjärjestelmän tarjoamia asynkronisia pyyntöjä, jotka tunnistaa niille annettavasta pyynnön tila eli TRequestStatus viiteparametrilla, jonka arvo muuttuu pyynnön suorituksen mukaan. Asynkronisessa pyynnössä funktio palaa ennen kuin pyyntö on suoritettu ja pyynnön suorittaja eli funktion kutsuja voi jatkaa omaa toimintaansa pyynnöstä riippumattomana oliona. Asynkronisen pyynnön päätyttyä pyynnön tila muuttuu sisältää mahdollisen virhekoodin tai ilmoituksen virheettömästä suorituksesta. Pyynnön ollessa voimassa tila muuttuu ilmoittaa pyynnön olevan edelleen käynnissä. [Tasker, 1999]

5.2. Aktiiviolioiden aikataulutus

Aktiiviolioiden aikataulutuksen hoitaa niin sanottu aktiiviaikataulutaja, CActiveScheduler, joka käytännön tasolla sitoo säikeiden ja aktiiviolioiden toiminnan toisiinsa. Aikataulutajan omistaa ja sitä ajaa säie, yleensä prosessin pääsäie, ja sen tehtävänä on hoitaa aktiiviolioiden tekemien asynkronisten pyyntöjen käsittely. Aikataulutajaa käytetään aikataulutamaan järjestys, jossa niiden aktiiviolioiden, joiden pyynnot ovat täyttyneet, käsitellään.

Kutakin säiettä kohti voi olla vain yksi aktiiviaikataulutaja. Myös periyttämällä laajennettujen aktiiviaikataulutajien käyttö on mahdollista, jolloin muun muassa aktiiviobjektien pyyntöjen käsittelyssä tapahtuvien virheiden käsittely voidaan muokata sovelluksen tarpeiden mukaiseksi. Mikäli säikeessä halutaan käyttää jotain muuta kuin oletusaikataulutajaa, se pitää asentaa erikseen. Aikataulutaja voidaan poistaa säikeestä asentamalla tilalle tyhjä olio eli nolla. Ennen aktiiviolioiden käyttöä ne pitää lisätä aikataulutajaan, jotta se voisi hallita aktiiviolioiden pyyntöjen käsittelyn. Aikataulutaja ei ole järjestelmäolio ja se pitää tuhota ohjelmallisesti. Aikataulutajan tuhoaminen tuhoaa myös kaikki aikataulutajan hallinnassa olevat aktiivioliot [Symbian, 2002].

5.3. Aktiiviobjektien hallinta

Aktiiviolioiden käyttöön on käyttöjärjestelmässä tarjottu abstrakti luokka CActive, jonka erikoistamalla saa aktiivioliiorajapinnan käyttöönsä. Erikoistaminen tapahtuu periytyksen avulla.

Aktiivioliota luotaessa aktiivioliolle on annettava prioriteetti, jonka perusteella aktiiviolioiden aikatauluttaja määrittää aktiiviolioiden pyyntöjen käsittelyyn jaettavat suoritusvuorot. Rakennettavan aktiiviolion on oltava jonkin toisen olion jäsenmuuttuja jotta asynkroninen palvelu voitaisiin toteuttaa. Jotta Symbian OS käyttöjärjestelmän muistinhallinnasta johtuvat rajoitukset tulisivat täytettyä, on perityn olion muu rakentaminen tehtävä erillisessä, toissijaisessa rakentimessa, jota kutsutaan välittömästi varsinaisen rakentimen jälkeen. Tässä toissijaisessa rakentimessa on tavallista alustaa olion tarvitsemat, muistinvaraamista vaativat muuttujat sekä lisätä olio aktiiviolioiden aikatauluttajaan, joka mahdollistaa aktiiviolioiden rinnakkaisen kaltaisen toiminnan [Symbian, 2002].

Varsinaisen pyynnön asetus tapahtuu erillisellä metodilla, jonka nimeäminen tapahtuu perityn olion tarpeiden mukaisesti. Pyyntö asetavassa metodissa kutsutaan asynkronista pyyntöä, kuten esimerkiksi järjestelmän ajastinta tai tiedoston lukuun tarkoitettua metodia, jolle annetaan aktiiviolion jäsenmuuttujana oleva TRequestStatus viiteparametrina. Pyyntö asetuksen jälkeen aktiiviolio on määritettävä aktiiviseksi SetActive metodilla, jotta olioiden aikatauluttaja olisi tietoinen siitä, että juuri tällä aktiivioliolla on annettuna asynkroninen pyyntö. Metodissa joka pyynnön suorittaa on mahdollista suorittaa myös muita operaatioita, mutta koska vain yksi pyyntö voi olla aktiivinen kerrallaan, muita pyyntöjä ei voi samassa metodissa suorittaa [Tasker, 1999].

Kun aktiiviolion asettama pyyntö on täyttynyt eli kun pyynnön tila viiteparametrissa on joko virhekoodi tai ilmoitus virheettömästä suorituksesta ja kun aktiiviolio on saanut aktiiviolioiden aikatauluttajalta suoritusvuoron prioriteettinsa perusteella, on aktiiviolion käsiteltävä pyynnön täyttyminen. Pyyntöä käsittelemisen tapahtuu säikeistä opitulla tavalla RunL metodissa, jota aikatauluttaja kutsuu kun se antaa luvan aktiiviolion pyynnön käsittelylle. Pyyntöä käsitellessä on mahdollista asettaa aktiiviseksi uusi pyyntö. Uuden pyynnön asettaminen tapahtuu ensimmäisen pyynnön asettamisen tavoin. Asettamalla uuden pyynnön ensimmäistä pyyntöä käsitellessä aktiivioliosta on mahdollista muodostaa tietynlainen tilakone, joka esimerkiksi pitää yllä tietoliikenneyhteyttä tai suorittaa jotain pitkäkestoista laskutoimitusta. Myös tietyin aikavälein suoritettavat tapahtumat on mahdollista ajoittaa aktiiviolion avulla kutsumalla ajastinta aina uudelleen pyynnön käsittelyvaiheessa [Symbian, 2002] [Tasker, 1999]. Mikäli pyynnön käsittelyvaiheessa tapahtuu virheitä, ne annetaan erilliselle virheiden käsittelystä vastaavalle metodille. Virheiden käsittely on mahdollista siirtää myös aktiiviolioiden aikatauluttajalle.

Aktiiviolion asettama pyyntö on mahdollista keskeyttää, jolloin aktiiviolio poistetaan aktiiviaikatauluttajan aktiivisten olioiden listalta ja sen tekemän pyynnön tila asetetaan keskeytetyksi. Varsinainen keskeytyksen käsittely tapahtuu DoCancel metodissa, jolle ei aktiiviolorajapinnassa ole lainkaan toteutusta eli sen toteutus on periytetyn aktiiviolion vastuulla.

Aktiivioliot sijaitsevat säikeelle luodussa keossa eli järjestelmän muistissa ja siispä oliot on tarpeellista tuhota kun niitä ei enää tarvita. Aktiiviolioita ei tarvitse prosessien ja säikeiden tavoin sulkea, mutta ennen niiden tuhoamista täytyy varmistua, ettei niillä ole aktiivista pyyntöä, joten aktiivioliota tuhottaessa on järkevää keskeyttää aktiiviolion pyyntö siitakin huolimatta, ettei pyyntöä enää olisi [Symbian, 2002].

Jotta aktiiviolorajapinnasta periytetyssä luokassa olisi luokalle ominaista toiminnallisuutta, ja jotta aktiiviolio toimisi oikein, on siis syytä erikoistaa pyynnön käsittelyyn ja keskeytykseen sekä virhetilanteiden käsittelyyn tarkoitettut metodit ja muodostaa sekä varsinainen rakennin että toisen vaiheen rakennin ja purkaja.

Aktiiviobjektien ollessa suoritusvarmistamattomia aktiiviobjektien suoritusta ei voida keskeyttää kesken aktiiviobjektin RunL metodin. Tästä johtuen aktiiviobjektien keskenään jakaman muistin hallinta ei tuota ongelmia. Muistin käsittely on atomista ja esimerkiksi muistin kirjoitus ja lukuoperaatioita ei voida tehdä samanaikaisesti. Aktiiviobjekteja käytettäessä esimerkiksi lukot ja semaforit ovat siis tarpeettomia, mikä tekee aktiiviobjektien käytöstä säikeiden käyttöä helpompaa, koska muun muassa lukkiutumatilanteet ovat mahdottomia. [Tasker, 1999]

5.4. Aktiiviolioiden prioriteetit

Aktiiviolioiden priorisointi tapahtuu säikeiden tavalla erityisten prioriteettien avulla. Aktiivioliolle on mahdollista asettaa viiden eri tason prioriteetteja ja esimerkiksi joutokäyväälle aktiivioliolle tai käyttäjän syötteitä vastaanottavalle aktiivioliolle on tarjolla omat prioriteettinsa. Aktiivioliolla ei ole oletusarvoista prioriteettia vaan se on määriteltävä joka kerta aktiivioliota luotaessa. Säikeiden tavoin tyhjän prioriteettiarvon määrittäminen aktiivioliolle on mahdotonta.

Aktiiviolion prioriteettia on mahdollista muuttaa kesken aktiiviolion elinkaareen. Prioriteetin muuttaminen ei kuitenkaan ole mahdollista silloin kun aktiiviolio on esittänyt pyynnön, eli kun aktiiviolio on aktiivisessa tilassa. Useita säikeitä käytettäessä myös aktiiviolion aikatauluttajan omistavan säikeen prioriteetti vaikuttaa aktiiviolion todelliseen suoritukseen [Symbian, 2002].

5.5. Ongelmatilanteet

Säikeiden tavoin myös aktiivioliot voivat prioriteettien perusteella toimivan aikataulutamisensa vuoksi kohdata uuvutus ja näännytys ongelmia. Jokin olio voi korkean prioriteettinsa ja lyhyin aikavälein toteutuvien pyyntöjensä vuoksi saada jatkuvasti suoritusaikaa niin, että jokin toinen olio matalammalla prioriteetilla jää joko osittain tai kokonaan ilman suoritusaikaa jolloin sen toiminta häiriintyy ja tehtävät jäävät suorittamatta [Symbian, 2002] [Magee and Kramer, 1999]. Myös aktiiviolioiden tapauksessa uuvutukseen ja näännytukseen liittyvät ongelmat on pääosin ratkaistavissa oikeaoppisen priorisoinnin avulla, eli siten, että käyttäjän kanssa kommunikoivat aktiivioliot sekä esimerkiksi ajastimet ovat korkeaprioriteettisempia kuin taustalla toimivat laskelmat ja tietoliikenneoliot. [Tasker, 1999]

Aktiiviolioiden suoritusvarmistamaton toiminta voi myös aiheuttaa ongelmia. Koska aktiiviolioiden suoritusta ei voi keskeyttää, voi jokin pitkään kestävä asetetun pyynnön toteutumisen käsittely estää muiden olioiden suorituksen ja viivästyttää niiden pyyntöjen toteutumisen käsittelyä merkittäväällä tavalla. Tästä kärsivät erityisesti ajastimella toimivat aktiivioliot, jolloin ajastimen täytyttyä olio joutuu jonkun toisen olion suorituksen vuoksi odottamaan suoritusvuoroaan ja tällöin ajastin ei ole millään tavoin täsmällinen. Tähän aktiiviolioista liittymään ongelmaan ei varsinaisesti ole minkäänlaista ratkaisua, mutta ongelman vaikutusta voi pienentää pitämällä pyynnön täyttymisen käsittelevä RunL metodi mahdollisemman lyhyenä tai jakamalla käsittely osiin esimerkiksi tilakoneen avulla niin, että käsittelystä suoritetaan kerrallaan vain pieni osa, jonka jälkeen aktiiviolio asetetaan uudelleen aktiiviseksi esimerkiksi ajastimen avulla. Tällä tavoin käsittely jakaantuu osiin ja muut oliot saavat suoritusaikaa käsittelyn pitäessä taukoa [Symbian, 2002] [Tasker, 1999].

5.6. Esimerkki

Aktiiviolioiden käytössä ehkä hyödyllisin puoli on ajastimen käyttö tietyn tapahtuman toistamiseen. Perityn, ajastinta käyttävän aktiiviolion C++ otsikkotiedosto näyttää esimerkiksi seuraavalta.

```
class CExampleActiveObject : public CActive
{
public:
    CExampleActiveObject(TInt aPriority);
    void ConstructL();
    ~CExampleActiveObject();
```

```

    SetRequest();
private:
    void DoCancel();
    void RunL();
    RTimer iTimer;
};

```

Toteutustiedostossa rakennin kutsuu ainoastaan CActive luokan rakenninta sille annetulla prioriteettiparametrilla. Toissijainen rakennin tässä tapauksessa luo käytettävän ajastimen sekä lisää aktiivioliion aktiiviolioiden aikatauluttajaan.

```

void CExampleActiveObject::ConstructL()
{
    // luodaan ajastin
    iTimer.CreateLocal();
    // lisätään aktiiviolio omistavan säikeen aktiiviolioiden aikatauluttajaan
    CActiveScheduler::Add(this);
}

```

Pyynnöä asetettaessa aktiiviolio täytyy pyynnön asettamisen lisäksi asettaa erikseen aktiiviseksi, jotta aktiiviolioiden aikatauluttaja saisi tiedon aktiivioliion pyynnöstä.

```

void CExampleActiveObject::SetRequest()
{
    // asetetaan ajastin päättymään sekunnin kuluttua
    TInt timeOut = 1 * 1000000;
    iTimer.After(iStatus, timeOut);
    // asetetaan olio aktiiviseksi
    SetActive();
}

```

Pyynnön käsittely tapahtuu RunL metodissa, jota aktiiviolioiden aikatauluttaja kutsuu asetetun pyynnön toteuduttua. Käsittelymetodissa voi asettaa myös uuden pyynnön, mikäli esimerkiksi halutaan suoritettun tapahtuman tapahtuvan säännöllisesti tietyin väliajoin.

```

void CExampleActiveObject::RunL()
{

```

```

// pyynnön käsittelevä koodi, eli esimerkiksi tapahtuman suorittava
// koodi suoritetaan RunL metodissa ensimmäisenä.
// asetetaan ajastin päättymään jälleen sekunnin päästä.
TInt timeOut = 1 * 100000;
iTimer.After(iStatus, timeOut);
// asetetaan olio uudelleen aktiiviseksi.
SetActive();
}

```

Uuden pyynnön asetuksen jälkeen aktiiviolion suoritus tulee pyynnön toteutumisen jälkeen uudelleen samaan pyynnön käsittely, eli RunL metodiin. Mikäli aktiiviolio halutaan pysäyttää eikä pyynnön toteutumista haluta odottaa voidaan kutsua pyynnön keskeyttävää metodia.

```

void CExampleActiveObject::DoCancel()
{
    // keskeytetään ajastimen toiminta
    iTimer.Cancel();
}

```

Aktiivioliota tuhottaessa on syytä muistaa kutsua juuri pyynnön keskeyttävää metodia, ettei aktiiviolion tuhoaminen aiheuttaisi virhetilanteita aktiiviolioiden aikatauluttajassa.

```

CExampleActiveObject::~CExampleActiveObject()
{
    // jos olio on aktiivinen peruuta pyyntö
    if(IsActive()) DoCancel();
}

```

Purkajassa voi ennen pyynnön perumista varmistaa, että pyyntö oli aktiivisena. [Symbian, 2002]

6. Yhteenveto

Symbian OS käyttöjärjestelmä on matkapuhelimissa hyvin laajalle levinnyt käyttöjärjestelmä, jossa on otettu huomioon laitteiden erikoispiirteet sekä niistä johtuvat, normaalista poikkeavat ohjelmistoalustavaatimukset. Symbian OS käyttöjärjestelmässä on myös mahdollistettu rinnakkaisohjelmointi, mutta sen toteutus on ratkaistu hieman yleisistä rinnakkaisohjelmointimahdollisuuksista

poikkeavalla tavalla. Käyttöjärjestelmässä on tarjottu rajapinta ytimen säikeisiin, jotka poikkeavat hyvin vähän säikeistä muilla alustoilla sekä tarjoamiensa toteutusmahdollisuuksien että aiheuttamiensa ongelmien puolesta.

Säikeiden käyttöä ei kuitenkaan suositella niiden vaatiman prosessoritehon vuoksi ja onkin järkevää korvata säikeet aktiivioliolin, jotka tukevat rinnakkaisen kaltaista ohjelmointia. Aktiivioliot on kehitetty käyttöjärjestelmän oliosuuntautuneisuuden ja säikeiden pohjalta ja ne muistuttavatkin pitkälti säikeitä sekä toiminnaltaan ja aikataulutukseltaan. Aktiivioliot ovat säikeitä huomattavasti kevyempiä ja helppokäyttöisempiä rakenteeltaan, mutta kuitenkin tarpeeksi tehokkaita jopa monimutkaisten tietoliikenneyhteyksien hallintaan.

Viiteluettelo

- [Digia, 2003] Digia Inc, *Programming for the Series 60 Platform and Symbian OS*. John Wiley & Sons Ltd, 2003.
- [Koskimies, 2000] Kai Koskimies, *Oliokirja*. Satku, 2000.
- [Magee and Kramer, 1999] Jeff Magee and Jeff Kramer, *Concurrency: State Models and Java Programs*. John Wiley & Sons Ltd, 1999.
- [Mery, 2003] David Mery, Why is a different operating system needed?. Technical paper, October, 2003. Available as <http://www.symbian.com/technology/why-diff-os.html>.
- [Sanders, 2002] Peter Sanders, Creating Symbian OS phones. Technical paper, April, 2002. Available as <http://www.symbian.com/technology/create-symb-OS-phones.html>.
- [Symbian, 2002] Symbian OS version 7.0 developers library. 2002. Available as http://www.symbian.com/developer/techlib/v70docs/SDL_v7.0/doc_source/index.html.
- [Symbian, 2003] Symbian OS phones. Available as <http://www.symbian.com/phones/index.html>.
- [Tasker, 1999] Martin P. Tasker, Active objects. Technical paper, May 27, 1999. Available as http://www.symbian.com/developer/techlib/papers/tp_active_objects/active.html.

Laajojen hajautettujen sovellusten ja tietojärjestelmien valvonta

Hannu Tanhuamäki

Tiivistelmä.

Tämä tutkimus käsittelee laajojen hajautettujen sovellusten valvontaa. Tutkimuksessa tutkitaan valvonnan toteuttamista ja erityyppisten sovellusten ja arkkitehtuurien valvontaan liittyviä haasteita. Tutkimuksessa tarkastellaan valvontaan liittyvän tiedon tuottamista, perille saamista, analysointia ja esittämistä.

Avainsanat ja -sanonnat: valvonta, hajautetut sovellukset.

CR-luokat: D.2.9, K.6.3 ja K.6.4

1. Johdanto

Ohjelmistotuotannossa ollaan yhä enemmän siirtymässä kohti palvelupohjaisia ohjelmistoja. Palvelupohjaisuuden ideana on tuottaa palveluita useille sovelluksille hyvin määriteltyjen rajapintojen kautta. Sovellusten palvelut tuotetaan eri ohjelmaprosesseissa ja mahdollisesti eri palvelinkoneilla. Palveluita luotaessa ei välttämättä edes tiedetä, mitkä sovellukset tulevat käyttämään tuotettua palvelua ja miten tuotettua tietoa tullaan jatkojalostamaan. Valvontaan liittyvät asiat tulee huomioida tehtäessä yksittäistä palvelua tai palvelukokonaisuutta.

Valvontatekniikoista merkittävin lienee verkkolaitteista myös sovelluspuolelle levinnyt SNMP (Simple network management protocol). Uutena ja monipuolisempänä tekniikkana mainitsemisen arvoinen on Java -sovellusten JMX [Savarese, 2003].

Ohjelmistojen valvonta on tärkeää ottaa huomioon jo niitä suunniteltaessa. Kriittisimpien järjestelmien osalta toimimattomuus tai väärän tiedon tuottaminen voi aiheuttaa hengenvaaran. Valvontamekanismien hoitaminen saattaa vaikuttaa myös ostopäätökseen.

Ohjelmiston koostuessa useista prosesseista palvelukokonaisuus ei lamaannu täysin yhden sovelluksen ongelmatilanteessa. Tästä syystä jopa palvelun käyttäjän on vaikea havaita, että jotain on vialla. Ongelma onkin siinä, miten tällaista palvelukokonaisuutta voidaan valvoa ja reagoida vikatilanteisiin oikealla tavalla.

Ongelma sovelluksessa voi tarkoittaa montaa eri asiaa. Toisaalta sovellus voi kaatua ilman, että se antaa mitään merkkiä ongelmasta tai toisaalta ohjelmassa

syntynyt poikkeus saadaan kiinni ja siihen reagoidaan. Poikkeuksen kiinni saamisen etuna on se, että ongelmasta voidaan saada hyvinkin tarkkaa tietoa. Jos esimerkiksi tiedon tallentaminen levyille tai tietokantaan epäonnistuu, voidaan ongelma hallita ohjelman sisällä yrittämällä tiedon tallennusta hetken kuluttua uudelleen. Ongelma ei usein ratkea uudelleen yrittämisellä. Tallennus saattaa esimerkiksi vaatia levytilan vapauttamista. Tämän tiedon saaminen ylläpitäjälle voi olla ongelmallista tilanteessa, jossa palvelu koostuu useissa koneissa olevista palveluprosesseista

Valvontaa liittyvät viestit voidaan tuottaa useilla eri tavoilla. Ehkä yleisin tapa on lokiviestien käyttö. Lokia kirjoitetaan yleisimmin tiedostoon. Menetelmä on käyttökelpoinen perustilanteissa, mutta ongelmia syntyy, jos ongelma ohjelmiston toiminnassa koskeekin kiintolevyä, joka on täyttymässä tai hajoamassa.

Paikallisesti tietokoneella voidaan käyttää erillistä valvontaohjelmaa, joka tarkkailee ohjelmaprocessia ulkoapäin. Tällainen mekanismi kertoo luotettavasti ohjelmaproessin kaatumisesta, mutta ei muuta. Tilanne hankaloituu entisestään, kun ongelma onkin käyttöjärjestelmän tasolla. Hankala tilanne syntyy, kun sekä valvovan että valvottavan sovelluksen yhteinen äitiprosessi kaatuu.

Hajautettujen sovellusten toiminnallisuutta käsitellään monissa tutkimuksissa. Uusi palveluperustainen ajattelutapa ohjelmistokehityksessä tuo hajautettujen sovellusten valvontaan uusia näkökulmia. Tämä tutkimus tarkastelee hajautettujen sovellusten valvontaa eri arkkitehtuurien ja sovellustyyppien näkökulmasta ja kokoaa yhteen niissä huomioitavat erityispiirteet.

Tutkimuksessa tullaan antamaan kuva eri sovellusalueiden valvontaan liittyvistä kriittisistä ongelmakohdista. Eri sovellusalueille pyritään määrittämään tärkeimmät valvontakohteet, sekä menetelmät niiden valvontaan. Tässä tutkimuksessa osoitetaan periaatteet, joilla hajautettujen sovellusten valvonta voidaan järjestää kattavasti. Tutkimuksessa kuvataan, missä tilanteessa käytetty mekanismi antaa oikean tiedon. Tutkimuksen tuloksia voidaan käyttää apuna, kun halutaan luoda mahdollisimman kattava valvontamekanismi tai arvioitaessa olemassa olevan valvontajärjestelmän kattavuutta.

Tietotekniikassa on hyvin yleistä, että useat termit esiintyvät asiayhteydestä riippuen hieman eri merkityksessä. Seuraavassa on esitetty keskeisimmät termit siinä merkityksessä kuin niitä on tässä tutkimuksessa käsitelty. Hajautettu järjestelmä on määritelty seuraavasti: Hajautettu järjestelmä on sellainen järjestelmä, jossa verkotetut tietokonekomponentit kommunikoivat ja

koordinoivat toimintansa vain välittämällä (verkon yli) viestejä keskenään [Coulouris et al., 2001]. Tässä tutkimuksessa komponentit kuitenkin voivat olla myös samassa koneessa, mutta eri prosessissa. Käsite laaja hajautettujen sovellusten yhteydessä tarkoittaa tässä tutkimuksessa sitä, että sovelluksia on monissa eri tietokoneissa. Sovellukset jakavat tietoja keskenään jollakin tavalla verkon välityksellä. Valvonnalla tarkoitetaan eri sovellusten toimivuuden valvontaa. Sovelluksen katsotaan toimivan normaalisti, mikäli sen tuottamat palvelut toimivat ongelmitta. Ohjelman sisällä jostain syystä syntyvää poikkeusta, joka hallitaan ohjelman omilla keinoilla, ei tässä tutkimuksessa käsitellä valvontamielessä vikana.

Tämä tutkimus jakaantuu lukuihin luvun yksi johdanto jälkeen siten, että luvussa kaksi käsitellään sovellusten valvontaa prosessina ja syitä valvontaan. Luvussa kolme pohditaan valvottavia kohteita ja miten ne vaihtelevat eri arkkitehtuuriratkaisuissa sekä miten valvotaan valvojaa. Eri valvontaviestityyppejä, niiden muodostamista, liikkumista, varastointia käsitellään luvussa neljä. Luvussa viisi käsitellään valvonnan vaikutusta suorituskykyyn ja luvussa kuusi käsitellään viestien analysointia ja näyttämistä käyttäjälle. Luku seitsemän kokoaa yhteen tutkimuksen tulokset ja siinä pohditaan myös jatkotutkimusaiheita.

2. Valvonta

Tässä luvussa käsitellään valvontaa prosessina ja syitä valvontaan. Valvontaprosessin käsittelyssä kuvataan, mitkä eri osat siitä voidaan tunnistaa ja syistä listataan merkittävimmät.

2.1. Valvontaprosessi

Valvontaprosessi voidaan jakaa valvontatiedon tuottamiseen, prosessointiin, jakeluun ja esittämiseen [Mansouri-Samani and Sloman, 1992]. Tuottamisella tarkoitetaan tila- ja tapahtumaraportteja sekä näiden riippuvuuksia. Prosessointi käsittää tiedon oikeellisuuden tarkastamista, yhdistämistä ja suodattamista siten, että vain tarpeellinen tieto toimitetaan eteenpäin. Jakelulla tarkoitetaan tiedon toimittamista niille, jotka sitä tarvitsevat. Esittäminen tarkoittaa tiedon näyttämistä ihmiselle siten, että valvottavan järjestelmän tilasta saadaan oikea käsitys.

2.2. Sovellusten valvonnan tärkeys

Ennen kuin siirrytään sovellusten valvontaan, on aina syytä pohtia: miksi juuri tätä sovellusta tulee valvoa. Vastaus tekee tärkeän rajauksen kannattaako kyseistä sovellus siis lainkaan liittää valvonnan piiriin. Lisäksi tulee pohtia,

onko järjestelmän toiminnassa ollut aiemmin vikoja ja mitä siitä seuraa, jos vikoja esiintyy.

Wong [1998] listaa seuraavat syyt valvonnan järjestämiseen:

- parantaa ongelmien tunnistamista,
- parantaa ongelmien selvittämistä,
- laatuvaatimusten täyttämisen todentaminen,
- tehokkuusanalyysien parantaminen ja
- luotettavuusanalyysien parantaminen.

Hajautettujen järjestelmien kannalta olennaista on juuri vian havaitseminen ja paikallistaminen. Kun vian sijainnista on varmuus, voidaan korjaavat toimenpiteet kohdistaa oikein.

Kattavan valvonnan järjestäminen teettää asiakkaalle kustannuksia. Jos valvontaan liittyvät asiat on huomioitu jo alun perin sovelluksen suunnittelussa, voidaan saavuttaa kilpailuetu asiakkaan tehdessä kriittisen sovelluksen ostopäätöstä.

3. Mitä valvotaan

Tässä luvussa käsitellään valvottavien kohteiden valintaa sovelluksen tyypin ja arkkitehtuurin perusteella. Lisäksi käsitellään kysymystä: kuka valvoo valvojaa.

3.1. Kohteiden valinta

Tämän päivän sovellukset eroavat toisistaan monin tavoin, kuten tarkoituksen, koon, arkkitehtuurin ja kriittisyyden perusteella. Kaikkia sovelluksia ei selvästikään kannata valvoa. Valvontamekanismien tarpeellisuuteen vaikuttavia tekijöitä ovat sovelluksen kriittisyys, monimutkaisuus, laajennettavuuden tarve, millaiset määräyksen yrityksellä on valvonnasta sekä kenelle sovellus on suunnattu.

Sovellusten valvontaa voidaan tarkastella niiden arkkitehtuurien näkökulmasta. Sovellustyypit voidaan valvonnan kannalta jakaa keskitettyihin, hajautettuihin, internet-, sähköisenkaupan ja palveluperustaisiin sovelluksiin [Kreger *et al.*, 2002].

3.2. Eri sovellustyyppien valvonta

Keskitetyissä sovelluksissa, jollaisia ovat tyypillisesti esimerkiksi taloushallinnon sovellukset, halutaan taata palveluiden jatkuva saatavuus ja tehokkuus [Kreger *et al.*, 2002]. On tärkeää tunnistaa esimerkiksi, että yksittäisten prosessien liiallinen kuormittuminen voi estää koko palvelun

käytön. Asiakassovelluksia tarkasteltaessa merkittävintä ovat riittävän nopeat vasteajat palvelupyynnöissä. Hajautetuissa sovelluksissa palveluiden saatavuus on tärkeää, mutta ongelmalliseksi asian tekee se, että nyt pitääkin valvoa useaa palvelinta yhden sijasta. Postijärjestelmät ovat usein hajautettuja ja niissä saatavuuden, yhteyksien ja toimintavarmuuden turvaaminen on tärkeää.

Internet-sovelluksille tyypillinen ominaisuus on niiden arkkitehtuurin kerroksittaisuus. Palveluketjun toimivuus edellyttää kaikkien sovelluskerrosten moitteetonta toimintaa. Toiminnan turvaaminen edellyttää koko ketjun valvontaa selaimen, www-palvelinten, sovelluspalvelinten ja tietokantojen välillä. Internet sovelluksia käytetään esimerkiksi luomaan yhteyksiä internet-selaimella yrityksen keskitettyihin palveluihin.

Elektronisen kaupan järjestelmillä tarkoitetaan kokonaisuutta, jossa kaupan toimitusketju on joko kokonaan tai osittain siirretty internetiin [Kreger et al., 2002]. Palveluketjun turvaaminen on tällöin yrityksen toimintaedellytys. Sovellusten ja yhteyksien valvonta on haasteellista, koska valvottavista sovelluksista ei kaikki enää välttämättä ole yrityksen omassa organisaatiossa ja koska yhteyksissä yleensä käytetään epäluotettavaa http-protokollaa.

Palveluperustaisissa sovelluksissa valvonnan yhteystyypit vaihtelevat. Ne voivat olla joko yhteydellisiä tai tarvittaessa yhteyden muodostavia. Muuttuvien yhteyksien ja riippuvuuksien ajonaikaista muuttumista on vaikea valvoa.

3.3. Valvonnan valvonta

Tärkeää on myös saada varmuus siitä että valvontajärjestelmä tekee sen mitä sen tuleekin tehdä [Wong, 1998]. Ensimmäinen ajatus ratkaisuksi olisi varmaan: valvotaan valvojaa. Tällöin ei kuitenkaan voida olla varmoja siitä valvoja raportoisi luotettavasti omasta tilastaan. Wong [1998] esittää ongelman ratkaisuksi kolmea vaihtoehtoa.

Ensimmäisessä maksimoidaan luotettavuus minimoimalla valvontakoodin monimutkaisuus. Tällöin käytettäisiin parhaita ohjelmoijia ja parhaita ohjelmointikäytäntöjä sekä varmistettaisiin, että valvojat ovat kaikkialla samoja ja hyvin testattuja. Toiseksi voidaan käyttää kolmannen osapuolen valvontakoordinaattoria, joka valvoo muita valvontaprosesseja järjestelmässä. Lisäturvaksi voisi vastaavasti olla valvonta valvontakoordinaattorille. Kolmannessa ratkaisussa on kaksi itsenäisesti tehtyä valvontaohjelmaa, jotka valvovat samoja sovelluksia. Molempien valvojien tuottamat tiedot yhdistämällä saadaan erittäin suuri luotettavuus.

4. Valvontatiedon tuottaminen

Tässä luvussa käsitellään eri valvontaviestityyppejä ja niiden käyttötarkoituksia sekä niiden muodostamista. Lisäksi käsitellään viestien varastointia ja kuljettamista.

4.1. Viestityypit

Valvonnan viestityypin voidaan jakaa kahteen osaan sen mukaan, tuottavatko ne tietoja ohjelman ongelmattomasta toiminnasta vai ohjelman toimivuudessa havaituista ongelmista. Kun viestejä lähetetään normaalitilanteesta, tulee niitä lähettää säännöllisesti. Ongelma havaitaan, jos viesti ei tulekaan perille odotettuna aikana. Ongelmatilanteesta viesti lähetetään vasta kun ongelma on havaittu.

Suunniteltaessa valvontamekanismeja tulee pohtia kysymystä: Jos sovellus tuottaisi virheilmoituksen, tulisiko se perille. Perille tulosta voidaan varmistua esimerkiksi lähettämällä säännöllisesti myös positiivisia viestejä. Näitä viestejä kutsutaan "olen hengissä (alive)" tai "sydämensyke (heartbeat)" viesteiksi.

Valvontaa varten tuotetut viestit tulee jaotella eri kategorioihin niiden vakavuuden suhteen. Yleisesti käytössä oleva Log4j [Log4j] loki järjestelmä jakaa sanomat viiteen luokkaa, joita ovat

- fatal,
- error,
- warning,
- info ja
- debug.

Kohtalokas (fatal) luokkaa käytetään yleensä silloin, kun ohjelma on mennyt sellaiseen tilaan, jossa sen toiminta on erittäin rajoittunutta tai se ei toimi lainkaan. Virhe (error) kuvaa tilannetta, jossa ongelma aiheuttaa jonkin palvelun estymisestä tai olennaisen tiedon menettämisestä. Ohjelma jatkaa toimintaansa virheestä huolimatta. Varoitus (warning) kuvaa tapausta, josta voi jossain tilanteessa aiheutua ongelmia tai ongelmia voidaan ennustaa tapahtuvaksi. Tietosanoma (info) sisältää tavallisesti positiivista tietoa ohjelman toiminnasta, esimerkiksi tiedon ohjelman käynnistyksestä. Ohjelmointitieto (debug) on viesti, jota hyödynnetään silloin, kun ohjelman toimintoja halutaan tarkastella syvemmillä. Näytä hyödynnetään tyypillisesti ohjelmiston testauksessa.

Dokumentti rfc3164 [Lonvick, 2001] määrittelee BSD syslog -protokollan, jossa sanomien prioriteetti määritellään taulukon 1 mukaan.

Koodi	Prioriteetti	Käyttötapa
0	Hätä (Emergency)	järjestelmää ei voi käyttää
1	Hälytys (Alert)	toimenpiteisiin tulee ryhtyä heti
2	Kriittinen (Critical)	kriittiset olosuhteet
3	Virhe (Error)	virheolosuhteet
4	Varoitus (Warning)	varoitusolosuhteet
5	Ilmoitus (Notice)	normaali, mutta merkittävä olosuhde
6	Lisätieto (Informational)	tiedoksi viesti
7	Koodaustieto (Debug)	koodaustason viesti

Taulukko 1, BSD-protokollan prioriteetit [Lonvick, 2001]

4.2. Viestien muodostaminen

Suuri osa sovelluksista tuottaa valvontatietoa toiminnastaan lokitietojen muodossa. Valvontamielessä näitä on kuitenkin harvoin hyödynnetty, koska tiedot päätyvät lähes poikkeuksetta tiedostoihin paikalliselle kiintolevyille. Sieltä niiden seuraaminen on äärimmäisen hankalaa, ellei mahdotonta.

Valvottava ohjelmisto voi itsenäisesti tuottaa valvontaviestejä ohjelman sisältä, tai ohjelman tilaa voidaan tarkkailla ulkoapäin tekemällä sille tilakyselyjä. Tilakysely ulkoa voi tarkoittaa ohjelmaprosessin tarkkailemista tai jotain muuta ohjelman toiminnon tarkkailemista. Tarkkailija voi ottaa suoraan yhteyden valvottavaan kohteeseen ja pyytää siltä vastusta, jos tällainen menettely on mahdollistettu valvottavassa kohteessa.

Ohjelman tarkkailija on syytä sijoittaa fyysisesti eri koneeseen, jotta ei synny tilannetta jossa ns. äitiprosessin kaatuminen kaataa yhtä aikaa sekä valvojan että valvottavan. Usein on järkevää, että valvottavasta koneesta saadaan mahdollisimman paljon tietoa ohjelman ulkopuolelle. Kaikkea dataa ei ole kuitenkaan järkevää siirtää verkon yli valvontakoneeseen. Tällainen tilanne voidaan ratkaista siten, että valvottavaan koneeseen sijoitetaan agentti, joka valvoo kohdetta, mutta välittää eteenpäin vain merkittävimmät viesti.

Johnson [1989] jakaa viestit kahteen osaan niiden muodostamisen perusteella. Pessimistinen tapa toimittaa viestit yksitellen ja optimistinen kerää ja lähettää niitä eteenpäin paketteina. Pessimistinen takaa, että viallinen prosessi voidaan korjata erikseen, koska muut prosessit eivät etene ennen kuin ne tietävät, että edellinen prosessi on päättynyt ilman ongelmia. Optimistisessa tavassa vastaavasti prosessit etenevät kunnes ne saavat tiedon ongelmista. Tieto voi tulla viiveellä, jolloin myös muita ohjelmaprosesseja saatetaan joutua

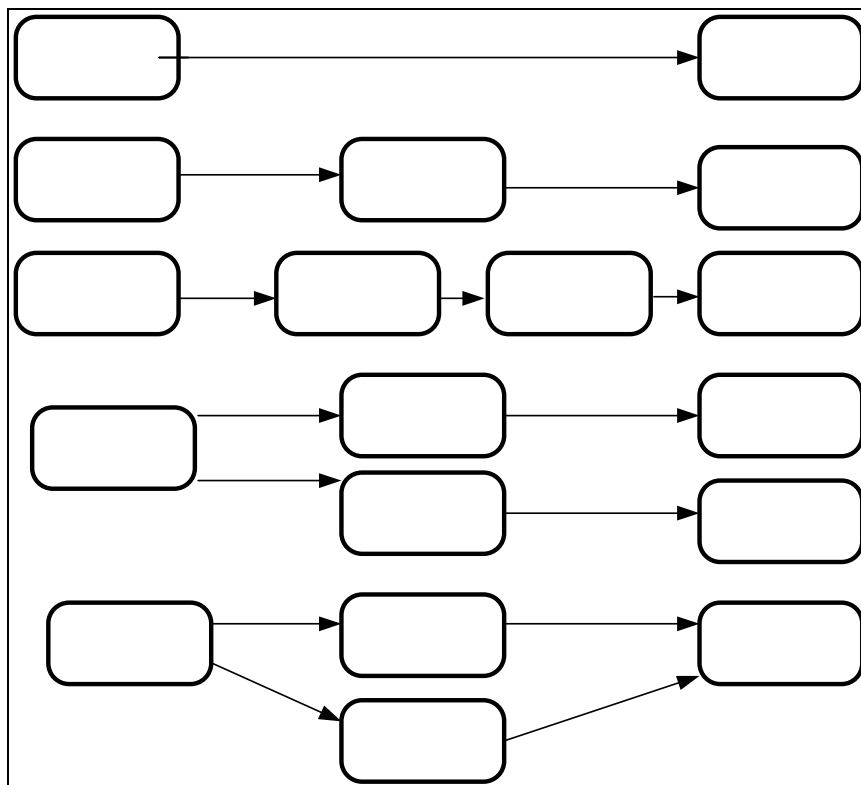
palauttamaan tilaan ennen ongelman esiintymistä. Muodostamistavan valinnassa joudutaan punnitsemaan vikojen vaikutuksia suhteessa tehokkuuteen.

Paikallisia valvonta-agentteja käytettäessä voidaan agentille määritellä hälytysrajoja, joiden ylitys vasta käynnistää hälytysviestien eteenpäin lähetyksen. Tarkkailemalla listojen kokoja, levytilaa prosessoriaikaa voidaan mahdollisista ongelmista saada tieto etukäteen.

4.3. Viestien perille saattaminen ja varastointi

Viestien perille menemisestä on hyvä saada varmistus. Järjestelmän virheilmoituksista ei ole hyötyä, jos ne jäävät paikalliselle koneelle esimerkiksi verkkovian takia.

Laajoissa järjestelmissä viestejä ei kannata tai niitä ei voi lähettää suoraan valvovalle koneelle. Tällöin käytössä on viestejä välittäviä yksiköitä. Viestit voi päätyä määränpäähän eri reittejä tai ne voivat haarautua eri valvontakoneille (kuva 1, viestien välitys). Esimerkiksi *BSD syslog* -protokollan periaatteena on, että valvottava ei tiedä lähettääkö se viestit välittimelle vai valvojalle, ja välitin voi lähettää kaikki tai vain osan informaatiosta eteenpäin [Lonvick, 2001].



Kuva 1, viestien välitys

Hajautetuissa järjestelmissä on toisinaan mahdotonta tunnistaa, kumpi tapahtumista tapahtui ensin [Lamport,1978]. Tästä syystä voi olla ongelmallista selvittää, kumpi tapahtuma oli aiheuttaja ja kumpi seuraus. Suunniteltaessa valvontaa lisäämällä viesteihin aikaleimat ja synkronoimalla kellot ongelmaa voidaan pienentää. Täyttä varmuutta ei pelkästään aikaleimoilla voida saavuttaa.

Valvottavat viestit tulee kerätä yhteen, jolloin niiden analysointi helpottuu. Jos näin ei toimita, ongelma voi syntyä myös aivan yllättävältä taholta. Pohditaan tilannetta, jossa järjestelmä antaa jatkuvasti varoituksia jostain asiasta monta kertaa sekunnissa. Jos viestit on määritelty tallennettavaksi samalle kiintolevyille missä valvottava järjestelmä sijaitsee, levytilan täytyminen voi estää ohjelman toiminnan.

5. Valvonnan vaikutus suorituskykyyn

Rakennettaessa valvontajärjestelmää tulee huomioida, että valvontaviestien generointi voi hidastaa varsinaisen järjestelmän toimintaa. Ongelmia syntyy, jos viestit lähetetään verkon yli ja operaation suoritus kestää siten kauan. Tällöin tulee tuottaa vain kaikkein välttämättömin data viiveiden minimoimiseksi.

Yleinen ongelma laajojen järjestelmien valvonnassa on tiedon määrä, joka kerätään ja välitetään eteenpäin [Wong, 1998]. Käyttöjärjestelmien tulee tällöin tukea tiedon tehokasta siirtoa, jotta hidastavat kohdat eivät muodosta niin sanottua pullonkaulaa, joka hidastaisi koko järjestelmän toimintaa.

6. Valvontaviestien käsittely

Tässä luvussa käsitellään valvontaviestien käsittelyn vaiheita. Niitä ovat suodatus, analysointi ja esittäminen käyttäjälle. Lisäksi käsitellään toipumista havaituista vioista.

6.1. Viestien suodatus

Viestitulva on ongelma, joka saattaa syntyä tilanteessa, jossa yksittäinen vika aiheuttaa ketjumaisesti ongelmia muille. Vastaava tilanne syntyy myös, kun verkkovian vuoksi jonossa olevat sanomat purkaantuvat. Ensimmäinen keino on tiedon suodatus [Mansouri-Samani and Sloman, 1992].

Suodatusta voidaan tehdä eri vaiheissa. Jos tietoja vastaanotetaan useita reittejä pitkin, voidaan niitä suodattaa lähettäjän perusteella. Suodattimilla poistetaan esimerkiksi vanhentuneet viestit käsittelystä.

6.2. Viestien analysointi

Analysointi tarkoittaa tila- ja tapahtumaraporttien yhdistämistä sekä niistä arvojen ja poikkeamien esittämistä [Mansouri-Samani and Sloman, 1992]. Tiedon analysointi parantaa tiedon ymmärrettävyyttä ja vikojen ennustettavuutta.

Vikoja voidaan ennustaa esimerkiksi siten, että havainnoidaan häiriöiden määrän tihentymistä, CPU käytön voimakasta lisääntymistä, vasteaikojen kasvamista jne. Analyysien tuottama tieto on kuitenkin suuntaa antava, eikä sen perusteella välttämättä voida tehdä kovinkaan pitkälle meneviä johtopäätöksiä.

6.3. Valvontatiedon esittäminen käyttäjälle

Kattavakin valvontajärjestelmä on hyödytön, jos tietoa ei kyetä esittämään loogisesti käyttäjälle. Mansouri-Samani and Sloman [1992] kuvaavat viisi ominaisuutta tiedon esittämiseen, jotka olisi hyvä löytyä käyttöliittymästä.

Käsitetasot tulee kuvata käyttäjälle yleisemmältä tasolta tarkempaan. Näkymiä voi olla esimerkiksi järjestelmä-, alijärjestelmä-, kone-, ohjelmisto-, prosessi- ja toimintotasolle.

Valvontatiedon sijoittelua näytöllä tulee voida muuttaa. Kokoamalla tietoja eri puolilta, käyttäjä voi muodostaa itselleen selkeämmän ja havainnollisemman kokonaisuuden tilanteesta. Havainnollisen kuvan avulla ongelman ratkaiseminen helpottuu.

Ajan hallinta näytöllä on merkittävä ominaisuus. Käyttäjän tulee voida käsitellä historiatietoa selaamalla tai hakujen avulla. Helpottavia ominaisuuksia ovat myös tilapäinen pysäyttäminen, askel askeleelta selaus, sekä reaaliaikainen ja hidastettu seuranta.

Erilaiset yhtä aikaa seurattavissa olevat näkymät mahdollistavat useiden kohteiden seuraamisen eri näkökulmista. Tästä voi olla apua esimerkiksi seurattessa häiriötä ja sen vaikutusta.

Tapahtuman tarkennettu näyttäminen on tarpeen vikaa korjattaessa. Valitsemalla tapahtuma näytöltä tulee olla mahdollista saada kaikki mahdollinen informaatio tapahtumasta. Tapahtuma- tai tilatiedosta näytetään tällöin mm. aikaleimat ja kohteen tunnistetiedot.

6.4. Toipuminen havaituista vioista

Kehittyneimmissä järjestelmissä ongelmille voidaan määritellä yksilöllisen tapa, jolla havaittuun ongelmaan reagoidaan. Yksinkertaisimmillaan tämä voi olla viesti valvontakoneen näytöllä. Tyypillisesti käytetään kuitattavia viestejä ja erilaisia värikoodauksia niissä havainnollistamaan viestien tärkeysluokittelua. Usein tärkeimmät viestit välitetään järjestelmän ylläpitäjille

esimerkiksi sähköpostilla tai tekstiviestinä matkapuhelimeen. Tällöin ongelma korjaantuu kuitenkin vasta siinä vaiheessa, kun ylläpitäjä on korjannut vian.

Joissain kriittisissä järjestelmissä voi olla perusteltua rakentaa valmiit mekanismit ongelmien varalle. Jos sovelluksen toimivuutta tarkkaillaan ulkoa päin, voidaan sovelluksen kaatumisen tai valvontayhteyden katkeamisen jälkeen pysäyttää sovelluksen prosessi ja käynnistää se uudelleen. Näin meneteltäessä on kuitenkin vaarana, että vian aiheuttaja on edelleen olemassa ja vikaa ei näin ollen voidakaan ohittaa.

7. Lopuksi

Tutkimuksessa tarkasteltiin valvontaa prosessina ja kattavan valvonnan järjestämiseen liittyviä ongelmakohtia. Valvottavien kohteiden valinnassa tärkeintä on sovelluksen kriittisyys. Eri sovellustyyppien valvonnassa tulee tunnistaa, mitä juuri niistä tulee valvoa.

Sovellusten järjestäminen pohjautuu valvontaviesteihin, jotka jaetaan luokkiin niiden kriittisyyden perusteella. Viestejä voidaan tuottaa järjestelmän ongelmattomasta toiminnasta tai häiriöistä. Ne voidaan toimittaa valvojalle joko suoraan tai välittäjien kautta.

On tärkeää huomioida, että valvonnalla on aina vaikutus sovelluksen suorituskykyyn. Virheellisesti järjestetty valvonta voi estää sovelluksen toiminnan.

Viestit suodatetaan ja analysoidaan ennen kuin ne esitetään valvontajärjestelmän käyttäjälle. Suodatuksessa poistetaan tarpeeton tieto ja analysoinnissa yhdistetään viestejä ja tehdään päättelyitä. Lopuksi tieto esitetään käyttäjälle. Käyttöliittymässä tulee olla mahdollista järjestellä tietoja ja suorittaa erilaisia hakuja.

Tämä tutkimus on katsaus hajautettujen sovellusten valvontaan. Tutkimuksen tiedot on kerätty kirjallisuuskartoituksen avulla perehtymällä aihetta käsitteleviin tutkimuksiin ja artikkeleihin. Tutkimuksen ollessa katsaus aiheeseen jää erilaisten numeeristen arvojen, kuten sanomien lukumäärien tai pakettikokojen, tarkastelu vähiin. Katsauksen pohjalta aihetta voisi syventää esimerkiksi tutkimalla valvontatiedon analysointia tai valvontatiedon esittämistä käyttäjälle.

Tutkimus kuvaa laajojen hajautettujen sovellusten valvonnan järjestämiseen liittyvät vaiheet ja käsittelee valvonnan haasteita eri sovellustyyppien näkökulmasta. Tutkimusta voidaan käyttää apuna arvioitaessa olemassa olevaa valvontajärjestelmää tai luotaessa uutta.

Viiteluettelo

- [Coulouris *et al.*, 2001] George Coulouris, Jean Dollimore, Tim Kindberg. Distributed systems - concepts and design. Addison-Wesley, 2001
- [Johnson, 1989] David Bruce Johnson, Distributed system fault tolerance using message logging and checkpointing. Rice university, 1989.
- [Kreger *et al.*, 2002] Heather Kreger, Ward K. Harold, Leigh Williamson, *Java and JMX: Building Manageable Systems*. Addison-Wesley 2002.
- [Lampport,1978] Leslie Lamport, Time, clock, and the ordering of events in a distributed system. Massachusetts Computer Associates, Inc. Communications of the ACM, **21** (7), July 1978, 558-565.
- [Log4j], Log4j, The Apache Jakarta Project. <http://jakarta.apache.org/log4j/docs/manual.html>
- [Lonvick, 2001] C. Lonvick, The BSD syslog protocol, rfc3164, august 2001. Also available as <http://www.faqs.org/rfcs/rfc3164.html>.
- [Mansouri-Samani and Sloman, 1992] Masoud Mansouri-Samani and Morris Sloman, Monitoring Distributed Systems. Imperial college research report No. DOC92/23 22 september 1992. Also available as <http://citeseer.nj.nec.com/mansouri-samani92monitoring.html>
- [Savarese, 2003] Daniel F. Savarese, JMX for managing java applications. *JAVAPro* October, 2003. Also available as http://www.ftponline.com/javapro/2003_10/magazine/columns/proshop/default.aspx.
- [Wong, 1998] Geoff Wong, A Quality Monitoring Framework. Software engineering research centre RMIT university. Also available as <http://www.serc.rmit.edu.au/70erlangtr/SERC-0078.pdf>

Apuvälinepalvelun tietojärjestelmät terveydenhuollossa

Outi Toivanen

Tiivistelmä.

Apuvälinepalveluiden merkitys tulee kasvamaan ikärakenteen muuttumisen vuoksi, siksi on tärkeää, että myös apuvälinepalveluilla on hyvä ja toimiva tietojärjestelmä. Apuvälinepalvelun tietojärjestelmä muodostuu kahdesta osasta; apuvälinetietopankista, joka on apuvälineiden hankintapaikka, ja apuvälinerekisteristä, mikä hallinnoi paikallista apuvälinemassaa. Tässä työssä esitellään kaksi järjestelmää; KuntoApu ja ApuNet.

Avainsanat ja -sanonnat: terveydenhuolto, tietojärjestelmä, apuväline, apuvälinepalvelu

CR-luokat: H.5.3

1. Johdanto

Tutkimukseni tarkoitus on selvittää teknologian hyötykäyttö apuvälinepalvelujen tiedonhallinnassa. Apuvälinepalvelu on yksi tärkeä osa sosiaali- ja terveydenhuollon palvelua. Mitä kauemmin kuntalaisten toimintakykyä ja omatoimisuutta tuetaan sitä kokonaistaloudellisempia kustannussäästöjä saadaan kunnassa aikaiseksi. Kunnilla on kuitenkin hyvin laaja päätäntävalta apuvälinepalvelujen toteuttamiseen lainsäädännön puitteissa. Käytäntö eri kunnissa on hyvinkin kirjava aina manuaalisista paperitulosteista toimiviin tietojärjestelmiin.

Kaikki apuvälinepalvelut käyttävät jonkinmoista hallinnollista järjestelmää. Tutkimukseni tarkoitus onkin esitellä apuvälinepalvelutoimintaa yleensä ja esittää muutama markkinoilla oleva järjestelmä; Festumin ApuNet ja Polyconin KuntoApu -järjestelmä.

Toisessa luvussa kerrotaan perustietoa Suomen terveydenhuollosta. Lisäksi kerron hieman Suomen terveystalouden hallinnollisesta rakenteesta ja suomalaisten ikärakenteesta ja sen muutoksista tulevaisuudessa. *Kolmannessa luvussa* kerron Satakunnanmakropilotin ja useiden eri kaupunkien sosiaali- ja terveystoimen näkemyksen siitä mikä määritellään apuvälineeksi ja mitä tarkoittaa apuvälinepalvelu. *Neljännessä luvussa* kerron Makropilotin (Sosiaali- ja terveydenhuollon kehittämishanke) vaatimusmäärittelyn millainen apuvälineiden hallintajärjestelmän tulisi olla; mitkä ovat ne kriteerit ja toiminnot mitä vaaditaan. Lisäksi kerron hyvin lyhyesti muutamasta hyvinkin

oleellisesta asiasta, kuten tietoturvallisuus ja maksupolitiikka. *Viidennessä luvussa* otetaan yleissilmäys tämän hetken tutkimustilanteeseen terveydenhuollossa valtakunnan ja kuntien tasolla.

Kuudennessa luvussa kerron Makropilotin apuvälinepalvelujen laatusuosituksista löydettyistä ongelmista, jotka hidastuttavat ja vaikeuttavat toimintaa. Luvun lopussa esitän muutamia laatusuosituksessa tulleita kehitysehdotuksia/vaatimuksia tulevaisuuden varalle. *Seitsemännessä luvussa* esittelen kaksi markkinoilla olevaa järjestelmää, jotka on tarkoitettu hallinnoimaan apuvälinepalvelun tietokantaa. ApuNet on esimerkkinä apuvälinetietopankista ja KuntoApu apuvälinerekisteristä.

2. Terveydenhuolto Suomessa

Tässä toisessa luvussa kerrotaan perustietoa Suomen terveydenhuollosta. Lisäksi kerron hieman Suomen terveyspalvelujen hallinnollisesta rakenteesta ja suomalaisten ikärakenteesta ja sen muutoksista tulevaisuudessa.

Suomen jokaisella kansalaisella on oikeus riittävään terveydenhuoltoon. Suomessa julkiset terveydenhoitopalvelut jakautuvat perusterveydenhuoltoon ja erikoissairaanhoidon. Perusterveydenhuollosta huolehtii kuntien terveyskeskukset, jotka voivat olla joko kuntien omia, yhteisiä useamman kunnan kanssa tai yksityisiltä palveluntarjoajilta ostettuja. Lisäksi kuntien on järjestettävä perusterveydenhuollon lisäksi erikoissairaanhoidon.

Suomen jokainen kunta kuuluu johonkin sairaanhoitopiiriin, joita on Suomessa on kaksikymmentä, jos Ahvenanmaa otetaan mukaan laskuihin. Jokaiseen sairaanhoitopiiriin kuuluu yksi keskussairaala ja useita aluesairaaloita. Keskussairaaloista viisi on yliopistollisia sairaaloita ja ne sijaitsevat ympäri Suomea: Helsingissä, Kuopiossa, Oulussa, Tampereella ja Turussa [Makupalat, 2003]. Sairaanhoitopiirin tehtävä on järjestää ja tuottaa erikoissairaanhoidon palveluja alueensa väestölle. [STM, 2003]

Pidämme hyvin korkeassa arvossa Suomen korkealaatuista terveydenhuoltoa, joka on turvattu perustuslain 19 pykälään kirjaamalla siihen jokaisen oikeus riittäviin sosiaali- ja terveyspalveluihin. Lainsäädännön tasolla vastuu näistä palveluista on jaettu: valtiolla on ohjaamisvastuu ja kunnilla järjestämisvastuu. [STM2, 2002] Kansalaisilla on siis yhtäläinen oikeus terveyspalveluihin asuinpaikasta ja taloudellisesta asemasta riippumatta. 90 %) sosiaali- ja terveyspalveluista rahoitetaan verovaroin (kunnallis- ja valtionveroilla). 10% kustannuksista rahoitetaan asiakas- ja potilasmaksuilla. Lisäksi Suomen valtio maksaa kunnille sosiaali- ja terveydenhuollon kustannuksista valtionosuutta, jonka määrä riippuu muun muassa kunnan asukkaiden ikärakenteesta, sairastavuudesta ja työllisyydestä. Sosiaali- ja

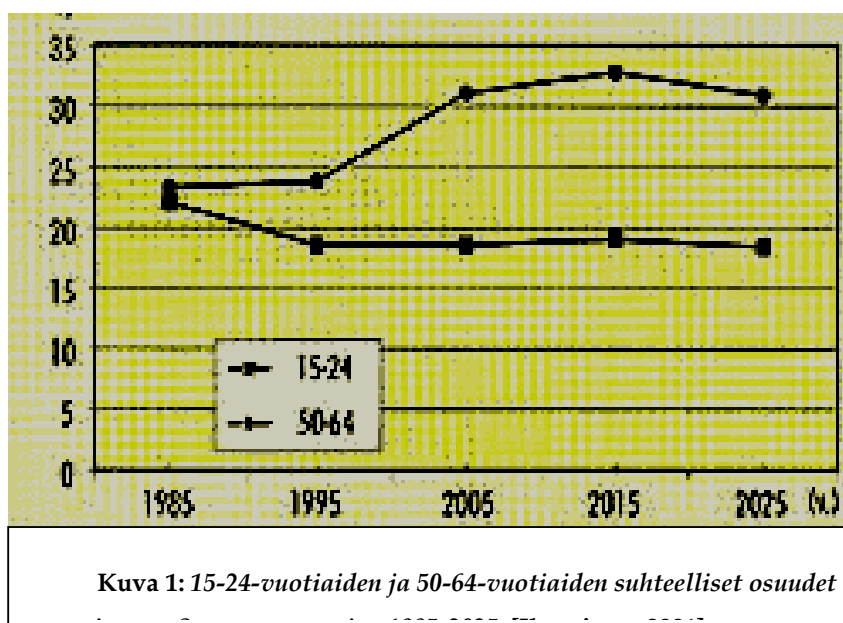
terveydenhuollon valtionosuuden kunta voi käyttää haluamallaan tavalla. [STM3, 1999]

2.1. Ikärakenne

Työvoiman ikärakenteet muuttuvat lähivuosina näkyvällä tavalla. Yli 50-vuotiaiden osuus työvoimasta kasvaa merkittävästi ja alle 25-vuotiaiden osuus pysyy matalana. Työpaikkojen keski-ikä nousevat ja nuorista alkaa olla pulaa monella

alalla.

Ennusteiden mukaan senioreja on työvoimassa nme lähes kaksi kertaa enemmän kuin junioreja vuoden 2005 jälkeen. Tilanteeseen ei ole



odotettavissa muutosta ainakaan vuoteen 2025 mennessä (kuva 1). Syynä pitkään jatkuvaan epäsuhtaan seniorien ja juniorien välillä ovat suurten ikäluokkien jälkeisten ikäluokkien (1950-60 luvulla syntyneiden) koko verrattuna viime vuosikymmenten pieniin ikäluokkiin. [Ilmarinen, 2001]

2.2. Ikärakenteen muutoksen vaikutukset

Väestön ikärakenteen vanheneminen tällä vuosikymmenellä tuo suuria muutoksia sosiaali- ja terveyssektorin kustannuksiin. Kun vanhusväestön osuus lisääntyy, sosiaali- ja terveyspalvelujen tarve kasvaa. Suuri osa niin sanotuista suurista ikäluokista jää eläkkeelle tällä vuosikymmenellä. Työmarkkinoille tulevat ikäluokat ovat pienempiä kuin eläkkeelle siirtyvät ikäluokat, mikä vaikuttaa siihen, että menot kasvavat ja tulot pienenevät. Tämä johtaa kuntien verotulojen pienenemiseen, jolloin kunnallisten palvelujen rahoittaminen muuttuu entistä vaikeammaksi. Ikääntymiseen liittyvät muutokset: rajoitukset nivelten liikkuvuudessa, vähentynyt kudosten elastisuus, lihasvoiman ja fyysisen toimintakyvyn heikkeneminen, hidastunut psykomotoriikka, muutokset havainnoinnissa ja päätöksenteossa, näkö- ja kuuloaistin muutokset, kylmä- ja kuumatoleranssin heikkeneminen, tuki- ja

liikuntaelinoireiden sekä sairauksien määrän kasvu, lisääntynyt kaatumisriski, hitaampi kuntoutuminen tapaturmista tai sairauksista, kohonnut työstressi ja liikunnallisen passiivisuuden lisääntyminen [Ilmarinen, 2001]. Nämä kaikki ikämuutokset vaikuttavat niin työpaikoilla kuin kotonakin. Vanhuusiän vaivoja voidaan helpottaa oikeanlaisilla apuvälineillä.

3. Yleistä apuvälineistä ja apuvälinepalveluista

Tässä kolmannessa luvussa kerron Satakunnanmakropilotin ja useiden eri kaupunkien sosiaali- ja terveystoimen näkemyksen siitä mikä määrittellään apuvälineeksi ja mitä tarkoittaa apuvälinepalvelu.

Yleisesti on käsitys, että hoitoa saa jos on sairas. Myös esimerkiksi Oulun [2003] sosiaali- ja terveystoimen sivuilla määrittellään apuvälineen saajaksi henkilö, jonka jokapäiväistä elämää vaikeuttaa sairaus tai vamma, joten taustaoletuksena tässäkin tutkimuksessa on, että henkilöllä on jokin rajoite.

Apuvälineeksi voidaan sanoa mitä tahansa laitetta, välinettä tai esinettä mikä auttaa henkilöä selviytymään paremmin ja vaivattomammin elämästä, esimerkiksi tietokonetta, laskinta ja silmälasia voitaisiin pitää apuvälineinä, koska ne helpottavat useiden ihmisten elämää.

WHO on määritellyt apuvälineen erityisesti valmistetuksi tai yleisesti saatavilla olevaksi miksi tahansa tuotteeksi, tarvikkeeksi, varusteeksi tai tekniseksi järjestelmäksi, jota toimintarajoitteinen henkilö käyttää lisätäkseen itsenäistä suoriutumistaan päivittäisissä tehtävissään. Apuväline ehkäisee, estää, kompensoi, helpottaa tai neutralisoi vaurion tai vamman aiheuttaman heikentyneen toimintakyvyn tai sairauden aiheuttaman haitan. [makropilotti, 1999]

Apuvälinepalvelu voi auttaa myös liikkumisessa, siirtymisessä, kommunikoinnissa, päivittäisissä toiminnoissa kuten syöminen, peseytyminen ja pukeutuminen lisäksi apuvälineet voivat auttaa ympäristön hallintaa. [Porvoo, 2003] Apuväline voi olla siis totuttujen pyörätuolien ja kävely-/tukikeppien lisäksi esimerkiksi puhuva vaaka tai kello, liukueste, rannekello tai herätyskello pistemerkinnöin tai erikoisvalmisteinen kodinhoitoväline jne.. [NKL, 2003]

Apuvälinetarvearvio arvio tehdään joka kerta yksilöllisesti lainaamossa, asiakkaan kotona tai apuvälineen hankintapaikassa [Oulu, 2003]. Apuvälinetarpeen arvioiminen, apuvälineen hankinta, käytön opetus ja lainaaminen kuuluvat olennaisena osana terveydenhoitoon [Porvoo, 2003]. Terveystoimen ei ole ainoa, joka myöntää apuvälineitä, koska myös kansaneläkelaitos, vakuutus- ja työeläkejärjestelmät, koulutoimi, työhallinto ja Valtiokonttori myöntävät apuvälineitä. Täten myös apuvälinepalveluja ohjaavia lakeja ja asetuksia on monia. Lait ja asetukset eivät kuitenkaan takaa

sitä, että palvelujärjestelmän toiminta olisi asiakkaan kannalta paras mahdollinen. [makropilotti, 1999]

4. Apuvälineiden hallintajärjestelmät

Tässä neljännessä luvussa kerron Makropilotin (Sosiaali- ja terveydenhuollon kehittämishanke) vaatimusmäärittelyn millainen apuvälineiden hallintajärjestelmän tulisi olla; mitkä ovat ne kriteerit ja toiminnot mitä vaaditaan. Lisäksi kerron hyvin lyhyesti muutamasta hyvinkin oleellisesta asiasta, kuten tietoturvallisuus ja maksupolitiikka.

Apuvälineiden hallintajärjestelmä muodostuu kahdesta erillisestä tietojärjestelmästä, apuvälinetietopankista ja apuvälinerekisteristä.

4.1. Apuvälinetietopankki

Apuvälinetietopankki on yleinen kaikkien alan organisaatioiden käytettävissä oleva, jonne palveluntarjoajat tallentavat tiedot apuvälineistä ja palveluista. [Henriksson-Leivo *et al*, 2000]

Sisällön on tarkoitus kattaa sekä hankkijoiden, käyttäjien, suunnittelijoiden ja kaikkien asiasta kiinnostuneiden tarpeet esimerkiksi eri ammattiryhmät sosiaali- ja terveydenhuollon alalla, apuvälineitä valmistavat, kehittävät ja markkinoivat yritykset (apuvälinpalvelujen tarjoajat) sekä eri tahot, jotka voivat joutua palvelemaan rajoitteista henkilöä, kuten esimerkiksi asunnon muutostyöasioiden parissa työskentelevät suunnittelijat ja rakentajat. Tietopankilla on myös joitain vaatimuksia, jotka voidaan jakaa seuraaviin pääryhmiin; Tietopankin toiminnalliset vaatimukset, hallinta, tietojen ja linkkien ylläpito, hankinta tai tilaustoiminto, hakutoiminto ja keskustelu sekä palaute forum. Sisällön kannalta voidaan myös mainita muutama tärkeä seikka. Välineluokitus ISO 9999 koodi, käytetään välineluokituksen mukaista nimeä, merkki, alatyyppejä, kuva, tekniset tiedot esim. korkeus, leveys, paino, käyttötarkoitus/soveltuvuus eri käyttötarkoituksiin ja -ympäristöihin, takuutiedot, huolto-ohjeet, vuokra-aika (leasing) jne.. Myöskin järjestelmän käytettävyys eri käyttäjäryhmien osalta tulee huomioida kuten esimerkiksi näkö- ja kuulovammaisten osalta. [Henriksson-Leivo *et al*, 2000]

4.2. Apuvälinerekisteri

Kaikki apuväline-palvelua tarjoavat tahot tarvitsevat apuvälinerekisteriä kuten terveyskeskukset, sairaalat, kuntien sosiaali- ja koulutoimi, Kela, vakuutuslaitokset, työvoimahallinto ja kuntoutuslaitokset. [Henriksson-Leivo *et al*, 2000]

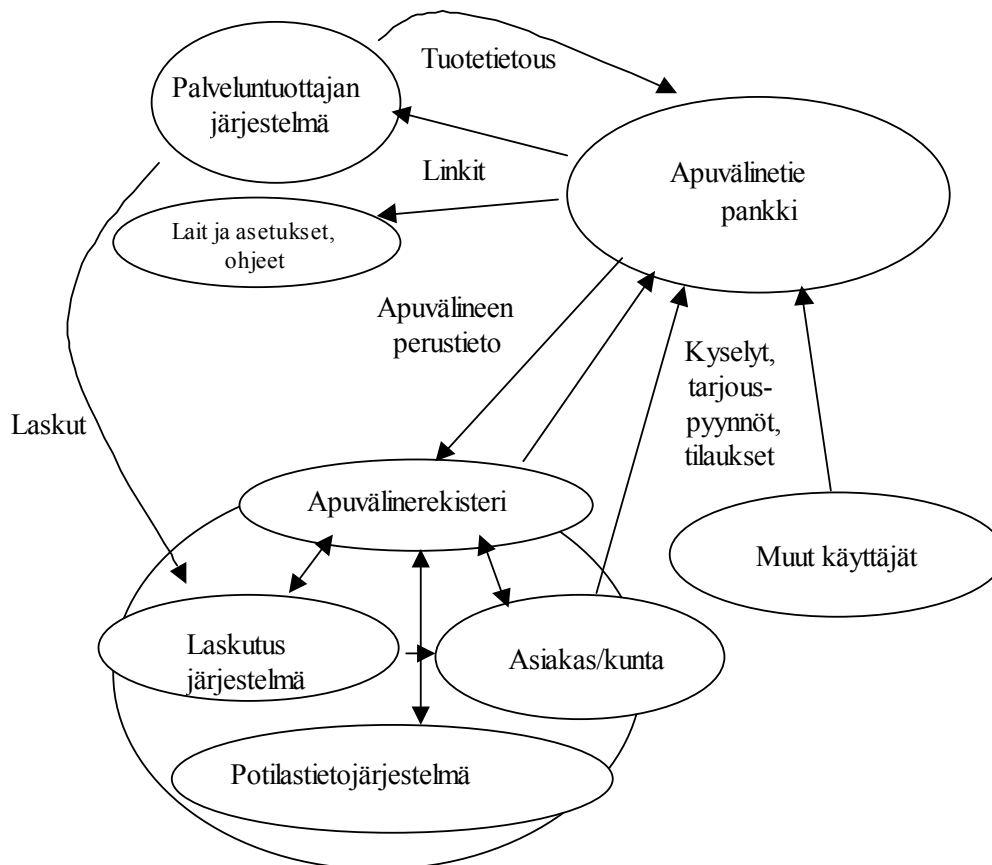
Rekisterin sisältää tiedot erilaisista apuvälineistä, joita ovat yksilöidyt lainattavat apuvälineet, massatavarat, henkilökohtaiset apuvälineet ja varaosat.

Yksilöidyt, lainattavat apuvälineet ovat koodattuja ja sarjanumerollisia. Massatavara on koodattua mutta ei sisällä sarjanumeroa, koska niiden yksilöiminen ei ole tärkeää, koska tavara on yleensä halpaa. Rekisteröiminen ja kuluseuranta on kuitenkin välttämätöntä, koska välineiden odotetaan palautuvan takaisin. Massatavaroita on esimerkiksi halvemmat suurennuslasit, tukiliivit, kävelykepit ja pienapuvälineet. Henkilökohtaiset apuvälineet, kuten massatavaratkin, ovat koodattuja mutteivät sarjanumerollisia. Henkilökohtaisten tavaroiden ei oleteta palautuvat takaisin kuten massatavaroiden. Henkilökohtaisia tavaroita on esimerkiksi peruukit ja silmälasit. Lisäksi on olemassa pelkän varastopaikan tarvitsijat, eli varaosat, joilla ei ole koodia. Lisäksi järjestelmässä on apuvälinetietojen lisäksi asiakkaan perustiedot, hyväksytyt potilaskohtaiset apuvälinepäätökset, palvelun hintatiedot, laskutustiedot, huoltotiedot, poistotiedot, palvelu- ja asiakasorganisaation tiedot, varastopaikat, kuljetuspalvelut sekä palautetieto. [Henriksson-Leivo *et al*, 2000]

Toiminnallisista vaatimuksista voitaisiin mainita henkilötietojen ja apuvälineen perustietojen kirjaaminen, hyväksyntämenettely, uuden apuvälineen hankinta ja laskutus, varaustoiminto, apuvälineen luovutus ja palautus apuvälineyksikköön, kuljetus, virheellisten tietojen poistaminen/muokkaaminen, palaute, apuvälineen huollon tilaaminen ja kirjaaminen, sekä tehdyt muutostyöt, lainaushistoria sekä hakutoiminnot asiakkaan tai apuvälineen perusteella. [Henriksson-Leivo *et al*, 2000]

Käyttäjaoikeuksien ylläpidosta vastaa ATK- osasto, käyttöoikeudet myöntää yksikön esimies. [Henriksson-Leivo *et al*, 2000]

Apuvälinerekisterin tulee kommunikoida eri järjestelmien kanssa ja se liittyy muihin taustajärjestelmiin aluetietojärjestelmänsä kautta. Taustajärjestelmiä voi olla esimerkiksi potilastietojärjestelmä, laskutusjärjestelmä, ajanvarausjärjestelmä, Kela, Väestörekisteri tai laskutusjärjestelmään liityntä. Alla olevasta kuvassa näkyy apuvälinerekisterin, tietopankin ja muiden järjestelmien väliset yhteydet selvemmin. [Henriksson-Leivo *et al*, 2000]



Kuva 2: Yhteydet [Henriksson-Leivo *et al*, 2000]

4.3. Tietosuoja

Järjestelmässä, joka on yhteydessä useampaan järjestelmään ja jolla on useampia käyttäjiä tulee olla tarkkaan määritellyt tietosuojakriteerit ja kaikkien osioiden tietosuojaaja pitää rajoittaa käyttäjäkohtaisilla määrittelyillä. Arkistoinnissa noudatetaan lakia potilaan asemasta ja oikeuksista, henkilötietolakia ja arkistolainsäädäntöä[Henriksson-Leivo *et al*, 2000].

4.4. Maksupolitiikka

Apuvälinepalvelut ovat maksuttomat. Myös apuvälineiden huolto kuuluu palveluun.

Lääkinnälliseen kuntoutukseen sisältyvien apuvälineiden kustantaminen kuuluu terveydenhuollolle ja toimii porrastetusti siten, että halvemmat apuvälineet myöntää kunnan terveyskeskus ja kalliimmista apuvälineistä vastaa keskussairaala. Vakuutusyhtiöt kustantavat apuvälineet esim. liikenneonnettomuudessa vammautuneelle. Lisäksi vammaispalvelut kustantavat asunnon muutostöitä ja kiinteitä apuvälineitä. Kela kustantaa opiskelussa ja työssä tarvittavia esim. näkövammaisten kalliita lukulaitteita. [Porvoo, 2003 ja Salo,2003]

5. Laatusuositukset ja kehitteillä olevat hankkeet

Tässä viidennessä luvussa otetaan yleissilmäys tämän hetken tutkimustilanteeseen terveydenhuollossa valtakunnan ja kuntien tasolla.

Stakesissa tehdyn Apuvälineiden saatavuus 2000- selvityksen mukaan suurin osa kuntoutusalan työntekijöistä toivoi valtakunnallisia linjauksia tai ohjeita apuvälinepalvelujen yhdenmukaistamiseksi. Tällä hetkellä apuvälineiden saatavuus ja luovutuskäytännöt vaihtelevat organisaatioittain ja alueittain. [Stakes, 2003e]

Sosiaali- ja terveysministeriö ja Suomen Kuntaliitto julkistivat Stakesin esityksen pohjalta valtakunnallisen apuvälinepalvelujen laatusuosituksen vuoden 2003 alussa. Laatusuositustyö perustuu valtioneuvoston hyväksymään sosiaali- ja terveydenhuollon tavoite- ja toimintaohjelmaan vuosille 2000-2003. Apuvälinepalvelujen laatusuosituksen tavoitteena on, että kuntalaiset saavat hyviä apuvälineitä ja apuvälinepalveluja tarpeen mukaisesti ja tasa-arvoisesti asuinpaikastaan riippumatta. Tavoitteena on myös, että apuvälineet ovat asianmukaisessa käytössä ja ne tukevat käyttäjänsä itsenäistä suoriutumista. [Stakes, 2003c]

Tällä hetkellä käynnissä on yksi valtakunnallinen apuvälinehanke, jossa koitetaan valjastaa ikäihmisten tietous apuvälineitä palvelevaksi tietoudeksi [Stakes, 2003f] ja useita kunnallisia hankkeita, tällä hetkellä noin kymmenen [Stakes, 2003a]. Lisäksi on alueellisia ITSE hankkeita. ITSE hankkeet kehittävät vanhusen ja vammaisten henkilöiden itsenäistä suoriutumista. [ITSE, 2003]

6. Ongelmia

Tässä kuudennessa luvussa kerron Makropilotin apuvälinepalvelujen laatusuosituksista löydetyistä ongelmista, jotka hidastuttavat ja vaikeuttavat toimintaa. Luvun lopussa esitän muutamia ISAKin JYTY projektissa tulleita kehitysehdotuksia tulevaisuuden varalle.

Stakesin apuvälinepalvelujen laatusuosituksessa [Stakes, 2003e] löydettiin useita erilaisia ongelmia. Niistä nähdään kuinka puutteellisia nykyajan teknologisoituneessa yhteiskunnassa tietojärjestelmät ovat. Esittelen, ne ongelmat, jotka viittaavat jollain lailla tietojärjestelmien ja teknologian puutteellisuuteen ja joita voitaisiin parantaa laatusuosituksella sekä hyvillä tietojärjestelmillä.

Apuvälineiden käytön seurannassa ja käyttöturvallisuuden valvonnassa on suuria puutteita, sillä kierrätettäviä ja käytössä olevia apuvälineitä ei huolleta riittävästi, tarkatusta koskevat kirjalliset ohjeet puuttuvat, muutostöiden jälkeen turvallisuuden vastuukysymyksissä on epäselvyyksiä ja lisäksi

apuvälineiden käyttöohjeet eivät aina seuraa apuvälineiden mukana [Stakes, 2003e]. Tämä aiheuttaa sen, että apuväline ei toimi kuten pitäisi ja voi olla jopa vaaraksi käyttäjälleen. Lisäksi puutteellinen seuranta mahdollistaa sen, että joillakin henkilöillä voi olla apuvälineitä, joita he eivät enää tarvitse.

Apuvälineprosessi myös kestää liian kauan. Prosessia pitkittää palveluprosessiin liittyvä byrokratia, yhteistyön puute, voimavarojen vähyyys ja osaamattomuus. Tällä on välittömiä seurauksia: apuvälineiden tarvitsijan toimintakyky heikkenee, mikä lisää kustannuksia ja aiheuttaa muiden palveluiden tarvetta. [Stakes, 2003e]

Apuvälinepalveluprosessin kuten sovituksen, huollon ja varastoinnin toteuttamiseen tarvittavat tilat ovat usein ahtaita, erillään toisistaan ja epähygieenisinä. Tämä heikentää toiminnan laatua. [Stakes, 2003e] Huonot tilat myös vaikuttavat apuvälineiden säilyvyyteen ja hyväkuntoisuuteen.

Ongelmia aiheuttaa myös henkilöstön vähäinen määrä sekä tietämättömyys. Henkilöstöllä ei ole tarvittavaa tietoutta markkinoilla olevista apuvälineistä, yhteistyötahoista tai henkilöillä ei ole työtä vastaavaa koulutusta. [Stakes, 200e] Tämä aiheuttaa sen, että asiakas ei saa asiantuntevaa ja laadukasta palvelua, lisäksi se estää eri tahojen yhteistyön.

Apuvälineiden yhteishankinnat ja -varastointi sekä kierrätys ovat vähäisiä. [Stakes, 2003e] Tämä lisää kuntakohtaisia kustannuksia tai estää kokonaan joidenkin kalliimpien apuvälineiden hankinnan.

Apuvälinepalveluprosessiin hakeutumisessa on erilaisia käytänteitä (mm. lähetekäytänne). Toiminnan seurantavälineet ovat huonot, kirjaamistavat ovat epäyhteneviä ja rekistereissä käytettävät luokitukset ovat erilaisia. [Stakes, 2003e] Tämä hankaloittaa eri järjestelmien tietojen vertailua.

Rajanveto-ongelmia on myös siinä, minkä organisaation vastuulle apuvälineen luovutus missäkin tapauksessa kuuluu. Kaikilla organisaatioilla ei myöskään ole käytössään sähköisiä apuvälineiden seurantajärjestelmiä. Sosiaalitoimi ei juuri lainkaan rekisteröi myöntämiään apuvälineitä. [Stakes, 2003e] Missään ei ole nähtävissä yhtenäistä käytäntöä siitä, miten tulisi toimia missäkin tilanteessa.

Terveystieteidenhuollossa olevat järjestelmät eivät sovi keskenään yhteen, valtakunnallista seurantajärjestelmää tai apuvälinealan tietopankkia ei ole. Nämä ongelmat estävät eri organisaatioiden yhteisten varastoinnin ja lainaamisen toteuttamista. Myös organisaatiotason ja valtakunnallisen luotettavan seurantatiedon kerääminen vaikeutuu. [Stakes, 2003e]

Kaikki yllä mainitut ongelmat olisi ratkaistavissa hyvillä ja toimivilla tietojärjestelmillä sekä ottamalla käyttöön esimerkiksi Stakesin tekemä laatusuositus, ratkaisusta lisää seuraavassa alaluvussa.

6.1. Käytännön kehittäminen

Seija Parkkinen[2003] on löytänyt useita samoja ongelmia ISAK tutkimuksessa JYTY kunnissa (Eno, Joensuu, Kiihtelysvaara, Kontiolahti, Liperi, Outokumpu, Polvijärvi ja Pyhäselkä). Parkkisen [2003] raportissa on kerätty tietoutta asiasta viime vuosikymmeneltä saakka. Parkkinen esittää useita parannuskeinoja tämänhetkiseen huonoon tilanteeseen. Esitän tässä mielestäni tämänhetkistä tilannetta helpottavia ratkaisuja.

Apuvälineiden rekisteröintikäytännön kehittäminen on laaja asiakokonaisuus. Ensisijainen kirjaamiseen liittyvä kehittämisasia on siirtyminen tietotekniikan hyväksikäyttöön kaikilla alueilla. Lisäksi yhtenäisten, saman luokitusjärjestelmän alaisten rekisteröintien käyttöönottoa tulisi kehittää. Esimerkiksi SFS-EN ISO 9999-luokitusjärjestelmä auttaisi jatkossa kehittämään tiedonkulkua ja yhtenäistämään toimintamalleja. Luokitusjärjestelmä tulisi ottaa käyttöön kattavasti kaikilla apuvälinepalvelua toteuttavilla tahoilla.[Parkkinen, 2003]

Apuvälineiden luokitusjärjestelmän, ISO -standardin, käyttöönotolla saataisiin monia hyötyjä. Yhtenäinen apuvälineluokitus, ulotettuna myös sosiaalitoimen kautta myönnettäviin apuvälineiksi tai asunnon varusteiksi luokiteltuihin välineisiin, loisi perustan apuvälineiden tilastointiin ja olisi samalla pohja esimerkiksi varastonhallinta- ja seurantajärjestelmien kehittämiselle.[Parkkinen, 2003]

Luokitusjärjestelmän kautta saataisiin koko alueelle apuvälinepalveluun yhtenäinen terminologia ja näin apuvälineisiin liittyvän informaation hakeminen helpottuisi. Yhteinen termistö helpottaisi tiedonkulkua työntekijöiden kesken alueen sisällä mm. suosituksissa ja päätöksissä ja mahdollistaisi yli kuntarajojen tapahtuvia apuvälinepalvelun osioita. [Parkkinen, 2003]

7. Markkinoilla olevia järjestelmiä

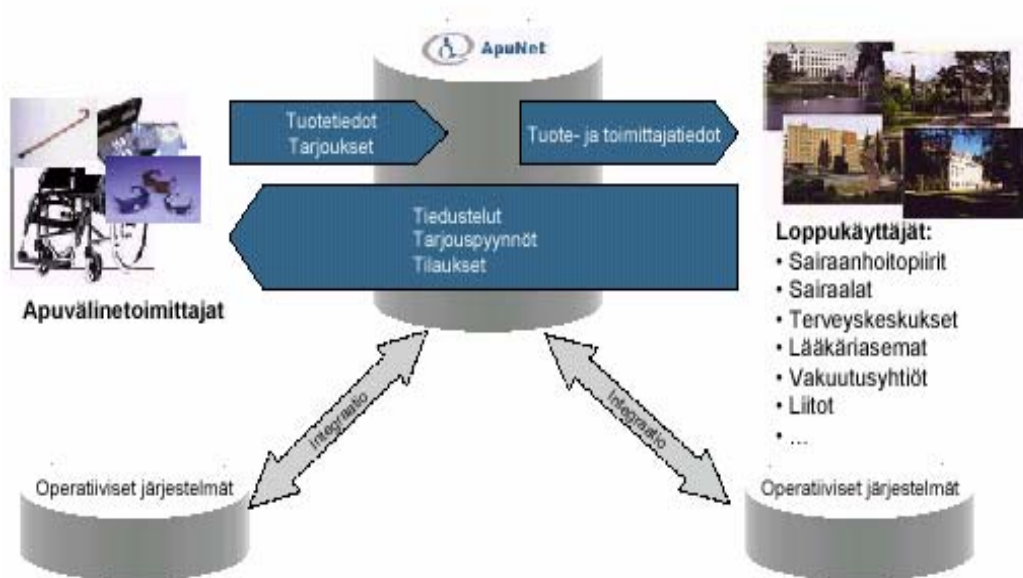
Tässä seitsemännessä luvussa esittelen kaksi markkinoilla olevaa järjestelmää, jotka on tarkoitettu hallinnoimaan apuvälinepalvelun tietokantaa. ApuNet on esimerkkinä apuvälinetietopankista ja KuntoApu apuvälinerekisteristä. tietojärjestelmillä sekä ottamalla käyttöön esimerkiksi Stakesin tekemä laatusuositus, ratkaisuisista lisää seuraavassa alaluvussa.

7.1. ApuNet

ApuNet Festum Oy:n tarjoama apuvälinetietopankki . [Festum, 2003] Apuvälinetietopankki on yleinen kaikkien alan organisaatioiden käytettävissä oleva apuvälineiden hankintapaikka. Apuvälinerekisteri voi olla sosiaali- ja

terveydenhuolto-organisaation tai yksityisen palveluntarjoajan omistama ja hallinnoima. [Henriksson-Leivo *et al*, 2000] Apuvälinetietopankin tarkoitus on sekä tarjota apuvälineiden toimittajille keskeinen myyntipaikka että helpottaa apuvälineitä tarvitsevien työskentelyä.

Alla olevasta kuviosta nähdään kuinka ApuNet toimii (Kuva2). Apuvälinetoimittajat lähettävät järjestelmään tuotetiedot ja tarjoukset, jotka ovat loppukäyttäjien nähtävissä. Loppukäyttäjät voivat lähettää järjestelmän kautta apuvälinetoimittajille tiedusteluja, tarjouspyyntöjä ja tilauksia. Lisäksi apuvälinetietopankin on mahdollista integroitua molempien käyttäjien operatiivisiin järjestelmiin, jolloin tiedon siirto on nopeampaa ja helpompaa.



Kuva 3: ApuNetin toimintaympäristö [Messo 2003]

Toiminta vastaa myös loppuraportin [Henriksson-Leivo *et al*, 2000] asettamia vaateita:

"Apuvälinetietopankkiin palveluntarjoajat tallentavat tiedot apuvälineistä ja palveluista joita käyttäjät voivat selata etsiessään sopivaa tuotetta tai palvelua loppuasiakkaan tarpeisiin."

ApuNet -hankkeessa toteutetaan apuvälineiden hankintapaikka WEB - palveluna. Järjestelmä sisältää tiedot eri toimittajien tarjoamista apuvälineistä sekä apuvälineiden hankintaan liittyvän toiminnallisuuden. [Messo, 2003]

ApuNetin käyttöliittymä on selainpohjainen, joten sitä on helpompi ja nopeampi opetella käyttämään kuin täysin uudenlaista järjestelmää. Koska järjestelmää voi käyttää myös internetin kautta, on sen tiedot on helposti saatavilla riippumatta sijainnista. [Festum, 2003].

Satakunnansairaanhoitopiiri on tehnyt sopimuksen Festumi Oy:n kanssa ja heidän määrittelemien määritysten pohjalta on tuotettu valmis ApuNet. Hankintapaikan toteutus on osa Satakunnassa käynnistettyä alueellisen apuvälinetoiminnan kehittämissuunnittelua, joka sisältää sekä toiminnan että tietojärjestelmien kehittämishankkeita [Prizz, 2003].

Messo [2003] kertoi haastattelussa, että ApuNet palvelu ei ole vielä käytössä, mutta sen pilotointi on jo käynnistynyt. Varsinainen toiminta käynnistyy vuodenvaihteessa. Hankintapaikan toteutuksesta on vastannut Festum Oy, joka myös toimii kokonaispalvelun tarjoajana. Jatkossa palvelu laajenee koko Suomen kattavaksi apuvälineiden hankintapaikaksi eli WEB-kaupaksi, jossa kaikki toimittajat ovat tavoitettavissa yhdessä paikassa. [Messo, 2003]

Messon [2003] mukaan palvelun käyttäjäkohderyhmänä ovat muun muassa: sairaanhoitopiirit, sairaalat, terveyskeskukset, lääkäriasemat, vakuutusyhtiöt, liitot ja yhdistykset ja niin edelleen. Heille järjestelmän käyttö on maksullista. Messon [2003] mukaan palvelun piiriin tultaneen saamaan noin 200 apuvälinetoimittajaa, joille järjestelmän käyttö on maksuton. Apuvälinetoimittajat ja jälleenmyyjät tuovat hankintapaikkaan yhteystietonsa ja tiedot tarjoamistaan tuotteista.

Jorma Messo [2003] listaa Apunetin hyödyiksi kustannussäästöt, tietojen keskittämisen yhteen paikkaan ja siten ajantasaisten tietojen helppo tavoitettavuus. Lisäksi järjestelmän suurimmista eduista voitaisiin mainita markkinoiden laajuus ja siten myös tuotevalikoiman laajuus, mikä helpottaa suoraan loppukäyttäjälle sopivimman tuotteen löytymistä. Markkinoiden laajuus edistää myös kilpailua ja on täten suoraan vaikutuksessa hinnoittelupolitiikkaan. Kilpailu voi myös helpottaa tilattavien apuvälineiden hankintakustannuksia, koska järjestelmässä voidaan vertailla eri toimittajien tuotteita ja hintoja. [Messo, 2003]

Uusi järjestelmä myös helpottaa ja nopeuttaa tilausten tarjouskäsittelyä, kun kaikki tilaukset ovat yhdessä järjestelmässä. Tilausten nopeampi käsittely mahdollistaa nopeamman ja asiakasystävällisemmän toimintaketjun. Koska apuväline on usein mittatilaustyötä, toimii ApuNet eri yhteistyökumppaneiden kanssa saadakseen valmiiksi lopullisen tuotteen, mikä lähetetään asiakkaalle. Koska kaikki tiedot ovat samassa järjestelmässä mahdollistaa se helpon, nopean, kontrolloidun ja seurattavan toimintaketjun esimerkiksi ApuNetin Ostoskori -toiminnon kautta, johon on merkitty tarjouspyyntö, vertailu ja tilaukset. [Messo, 2003]

Lopuksi Messo [2003] toteaa, että järjestelmä antaa mahdollisuuden keskittyä potilastoimintaan ja täten antaa myös parempaa palvelua.

7.2. KuntoApu

KuntoApu on yksi Polyconin neljästä tuotteesta. Se on ollut markkinoilla nyt yhteensä kuusi vuotta; ensimmäinen versio oli käytössä 1998 ja paranneltu kakkosversio valmistui maaliskuussa 2000. KuntoApu on nyt käytössä kahdeksassa eri sairaanhoitopiirissä, jonka käyttäjäpiiriin kuuluu kaksi yliopistollista sairaalaa, kuusi muuta keskussairaalaa sekä useita aluesairaaloita ja terveyskeskuksia. Käyttäjää on sairaalan tai terveyskeskuksen koon ja työkäytäntöjen mukaan muutamasta pariin sataan. [Polycon, 2003a] Maanlaajuisesti käyttäjiä on tällä hetkellä yhteensä vähän yli tuhat. [Esselström, 2003]

Polyconin KuntoApu on Windows-pohjainen apuvälineiden ja kuntoutuksen ostopalvelujen hallintajärjestelmä. Järjestelmällä hallinnoidaan apuvälinevarastot, apuvälineiden lainaukset ja palautukset sekä apuvälineiden tekninen huolto ja siihen liittyvät työmääräykset. [Polycon, 2003d].

KuntoApu on tällä hetkellä apuvälinekeskusten, keskussairaaloiden, aluesairaaloiden ja terveyskeskusten käytössä niiden apuvälineyksiköitä, kuntoutusyksiköitä, korvat-, silmät- ja hengitysyksiköt auttavana järjestelmänä. Lisäksi käyttäjäryhmiin kuuluu myös käytännöistä riippuen taloustoimistot, potilastoimistot, hankintatoimistot, tekninen huolto ja niin edelleen. [Esselström, 2003]

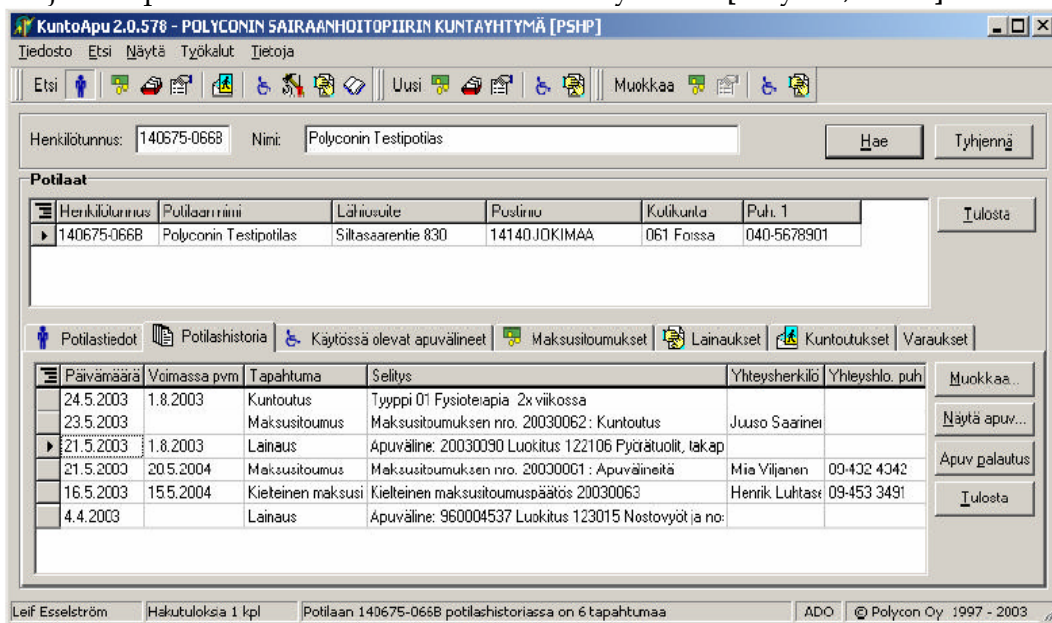
KuntoApu järjestelmän etuna on, että se voi olla joko itsenäinen kokonaisuus tai se voi jakaa tietokannan useamman toimijan kanssa kuten esimerkiksi Länsi-Pohjan keskussairaalassa on tehty; siellä keskussairaala ja 6 terveyskeskusta käyttää yhteistä järjestelmää. [Esselström, 2003]

KuntoApu -ohjelma voidaan linkittää esimerkiksi sairaalan oman henkilörekisteriin, laskutukseen/reskontraan, exceliin sekä muihin järjestelmiin.[Polycon, 2003a] KuntoApu -ohjelman integroituvuus kuitenkin riippuu siitä kuinka jäykkä toinen järjestelmä on. Esimerkiksi Novon Pegasos ja TietoEnatorin Effic [Esselström, 2003], mitkä ovat käytössä useissa terveydenhuollon palveluissa, haluavat tarjota kokonaisvaltaisia paketteja eivätkä mielellään tee liittymiä muihin järjestelmiin tämän vuoksi. Käytännössä isommat yritykset eivät kuitenkaan aina tee "täydellisiä" ratkaisuja, vaan yleensä osioita puuttuu tai ne ovat toteutukseltaan puutteellisia. KuntoApu on kuitenkin joustava ja tällä hetkellä liittymiä on noin 20 ulkoiseen järjestelmään, mutta kuten todettua niin liittymän teko vaatii yleensä, että kummatkin liitettävät järjestelmät ovat siihen valmiita. KuntoApuun on ensi vuoden aikana todennäköisesti tulossa liittymä Pegasokseen. [Esselström, 2003]

Tärkeimmät KuntoApu ohjelman toiminnot ovat apuvälinelainaustoiminnot ja maksusitoumukset [Esselström, 2003]. Muita

moduleita on raportointi, tarratulostus ja viivakoodituki, sairaanhoidon ostopalvelut, potilaskirje, apuvälineiden työmääräykset ja huollot sekä suositukset. [polycon, 2003a]

KuntoApu ohjelman tietosuojaa on varmistettu määrittelemällä jokaiselle KuntoApu -järjestelmän käyttäjälle käyttäjätunnukset ja käyttöoikeudet. Lisäksi järjestelmän kaikesta käytöstä jää merkintä tietosuojalokiin, mistä voidaan jällempäin tarkistaa mahdolliset väärinkäytökset. [Polycon, 2003a]



Kuva 4: KuntoApu järjestelmä [Polycon, 2003a]

Kuvassa 3 on esimerkki KuntoApu järjestelmästä. Näkymän välilehdellä näkyy yhden potilaan potilashistoria. Muita välilehtiä on potilastiedot, käytössä olevat apuvälineet, maksusitoumukset, lainaukset, kuntoutukset ja varaukset. [Polycon, 2003a]

8. Yhteenveto

Ikärakenteen muutos yhteiskunnassa aiheuttaa terveydenhuollon puolelle paineita kehittää toimivampia järjestelmiä ja käytäntöjä. Apuvälinepalvelu on yksi tärkeistä terveydenhuollon osista, sillä se tukee henkilön itsenäistä selviytymistä. Jotta kuntien kaikki resurssit saataisiin käyttöön on toimivat ja yhteenintegroituvat järjestelmät keskeisellä sijalla, sillä niiden avulla voidaan tehdä yhteistyötä eri kuntien ja ehkä jopa valtakunnan tasolla. Tulevaisuudessa näkisin merkittäväksi voimavaraksi kuntien joustavan yhteistyön ja voimavarojen jakamisen.

Apuvälineiden hallintajärjestelmät, joita tutkielmani käsittelee, muodostuvat kahdesta erillisestä tietojärjestelmästä, apuvälinetietopankista ja apuvälinerekisteristä. Apuvälinetietopankki on yleinen kaikkien alan organisaatioiden käytettävissä oleva, jonne palveluntarjoajat tallentavat tiedot apuvälineistä ja palveluista. Apuvälinerekisteri on sosiaali- ja terveydenhuolto-organisaation tai yksityisen palveluntarjoajan omistama ja hallinnoima. Alueellista apuvälinerekisteriä käytetään hallitsemaan alueellista apuvälinemassaa, apuvälineiden varastointia sekä huolto- ja muutostöitä. Näiden kahden järjestelmän joustava tiedonkulku on ehto joustavalle ja nopealle palvelulle. Jotta apuvälinepalveluissa päästäisiin tehokkaaseen toimintaan olisi jokaisella apuvälineyksikössä työskentelevällä oltava yhtäläiset tiedot eri mahdollisuuksista. Täten turvattaisiin myös laadukas ja asiantunteva palvelu riippumatta asuinpaikasta.

Viiteluettelo

- [Arinen et al, 1998] Sisko Arinen, Unto Häkkinen, Timo Klaukka, Jan Klavus, Risto Lehtonen, Seppo Aro. *Suomalaisten terveys ja terveyspalvelujen käyttö*, Terveystieteiden tutkimuskeskuksen 1995/96 päätulokset ja muutokset vuodesta 1987. 1998
Saatavana: <http://www.stakes.fi/palvelut/chess/Terva.pdf>
(15.12.2003)
- [Dialogi, 7/2003] Stakesin julkaisema lehti. 7/2003. s.45a. *Apuvälineet Satakunnassa*. Itse-hanke on Pave. Saatavana:
<http://www.stakes.fi/dialogi/03/dia22003/45a.htm> (15.12.2003)
- [Esselström, 2003] Polyconin CEO Leif Esselströmin haastattelu. KuntoApu. 1.12.2003
- [Festum, 2003] Apunetin kotisivut. Saatavana:
http://www.festum.fi/sol_apunet.jsp?lang=fi (15.12.2003)
- [Helsinki, 2003] Helsingin näkövammaisten apuvälinekeskuksen kotisivut.
Saatavana: <http://www.nkl.fi/apuvk/> (15.12.2003)
- [Henriksson-Leivo, 2001] Taina Henriksson-Leivo. Satakunnan makropilotti ry. *Apuvälineiden hallinta, loppuraportti*. 2001 Saatavana:
http://www.makropilotti.fi/loppuraportit/mp_apuvalineet.rtf
(15.12.2003)
- [Henriksson-Leivo et al, 2000] Taina Henriksson-Leivo, Pirkko Uusi-Oukari, Erkki Asikainen, Elina Lehmus, Anne Joensuu, Leo Siira. *Makropilotti. Apuvälineiden hallinta Tietojärjestelmien vaatimusmäärittely*. 2000. Saatavana:
<http://www.makropilotti.fi/raportit/Apuvalineiden%20hallinta,%20tietojarjestelmien%20vaatimusmaarittely.doc> (15.12.2003)
- [Ilmarinen, 2001] Juhani Ilmarinen, *Ikääntyminen on työelämän haaste seuraavan sukupolven ajan*, Työterveyslaitos, 2001. Saatavana:
<http://www.ttl.fi/Internet/Suomi/Tiedonvalitys/Verkkolehdet/Tyoterveiset/2001-01/02.htm> (15.12.2003)
- [ITSE, 2003] Sosiaali- ja terveysalan kehittämiskeskuksen ITSE hankkeet.
Saatavana:
<http://www.stakes.fi/ITSE-hanke/aluehankkeet/index.htm>
- [Makropilotti, 2003] Satakunnan makropilotti kotisivut. Sosiaali- ja terveydenhuollon kehittämishanke; Satakunnan Makropilotti.
Saatavana: <http://www.makropilotti.fi/> (15.12.2003)
- [Makropilotti, 1999] Satakunnan makropilotti. Apuvälineiden hallinta uusi toimintamalli. Saatavana:

- http://www.makropilotti.fi/raportit/Apuvalineiden_hallinta_toimintamalli_19991201.doc (15.12.2003)
- [Messo, 2003] Festumin myyntijohtaja Jorma Messon haastattelu. ApuNet. 20.11.2003.
- [Makupalat, 2003] Makupalan terveys linkit
Saatavana: <http://www.makupalat.fi/sairas3.htm> (15.12.2003)
- [NKL, 2003] Näkövammaistenkeskusliitto. Apuvälinekuvasto. Saatavana:
<http://www.nkl.fi/apuvk/kuvasto/hinnasto.html> (15.12.2003)
- [Oskenet, 2003] Sosiaali- ja terveysalan tietoteknologiasivusto.
Saatavana: <http://www.oskenet.fi/> (15.12.2003)
- [Oulu, 2003] Oulun kaupungin sosiaali- ja terveystoimen nettisivut:
Apuvälineet ja hoitovälineet. Saatavana:
<http://oulu.ouka.fi/sote/terveys/apuvalineet.htm> (15.12.2003)
- [Parkkinen, 2003] Seija Parkkinen. *Apuvälineiden ylläpitotoiminnan kehittäminen JYTY – kunnissa. Seutukunta-kohtainen raportti.*
Saatavana: <http://www.isak.fi/projektit/raportti.pdf> (15.12.2003)
- [Polycon, 2003a] KuntoApu esite. Saatavana:
<http://www.polycon.fi/kuntoapu/KuntoApu.pdf> (15.12.2003)
- [Polycon, 2003b] KuntoApu uutiset tammikuu 2003. Saatavana:
<http://www.polycon.fi/kuntoapu/uutiskirje1.pdf> (15.12.2003)
- [Polycon, 2003c] KuntoApu uutiset toukokuu 2003.
Saatavana: <http://www.polycon.fi/kuntoapu/uutiskirje2.pdf>
(15.12.2003)
- [Polycon, 2003d] Polyconin kotisivut. KuntoApu. 12.12.2003. Saatavana:
<http://www.polycon.fi/kuntoapu/index.html> (15.12.2003)
- [Porvoo, 2003] Porvoon sosiaali- ja terveyskeskuksen fysioterapian osasto.
Saatavana: <http://www.porvoo.fi/sosterv/apuvalineet.htm>
(15.12.2003)
- [Prizz, 2003] Maakunnallisen kehittäjäorganisaatio PrizzTech Oy:n kotisivut.
Saatavana: <http://www.prizz.fi/hcice.asp?id2=121> (12.12.2003)
- [Respecta, 2003] Respectan kotisivut. *Apuvälinekuvasto*. Saatavana:
<http://www.respecta.fi/materiaalit/kuvastohakemisto.htm>
(14.12.2003)
- [Salo, 2003] Salon seudun terveydenhuolto. Apuvälinepalvelu.
Saatavana: <http://www.wtk.salonseutu.fi/kuntpalv.htm> (15.12.2003)
- [Sosiaali- ja terveysministeriön opas, 2003] *Apuvälinepalveluiden laatusuositus*. Helsinki, 2003. (Sosiaali- ja terveysministeriön oppaita ISSN 1236-116X; 2003:7) ISBN 952-00-1373-3. Saatavana:

- http://pre20031103.stm.fi/suomi/eho/julkaisut/apuvaline/opas03_7.pdf (15.12.2003)
- [Stakes, 2003a] Apuvälinepalveluiden kehittämishankkeita. (15.12.2003)
Saatavana:
<http://www.stakes.fi/apudata/Hankkeet/hankkeet.htm>
(2.12.2003)
- [Stakes, 2003b] Lista apuvälinealan tietokannoista.
Saatavana: <http://www.stakes.fi/apudata/muita.html> (11.11.2003)
- [Stakes, 2003c] Apuvälineiden laatusuositus. Saatavana:
<http://www.stakes.fi/apudata/apuv%E4linepalv-laatusuositus/suositus.htm> (12.12.2003)
- [Stakes, 2003d] ITSE hanke.
Saatavana: <http://www.stakes.fi/ITSE-hanke/> (12.12.2003)
- [Stakes, 2003e] Laatusuosituksen ongelmat. 2003. Saatavana:
<http://www.stakes.fi/apudata/apuv%E4linepalv-laatusuositus/ongelmat.htm> (14.12.2003)
- [Stakes, 2003f] Stakesin valtakunnallinen hanke. Kohteesta toimijaksi -teknologiaa ikäihmisten ehdoilla. 2003. Saatavana:
<http://www.stakes.fi/apudata/Hankkeet/Koko%20maa.htm>
(14.12.2003)
- [STM, 2003] Suomen sosiaali- ja terveysministeriö, Terveyspolitiikka. 2003/10/26. Saatavana:
<http://www.stm.fi/Resource.phx/vastt/tervh/thpol/index.htx>
(13.12.2003)
- [STM2, 2003] Sosiaali- ja terveysministeriön kotisivut. *Vastaus kansanedustaja Matti Vanhasen ym. välikysymykseen kansalaisten perusoikeuksien toteutumisesta terveydenhuollossa.* 25.9.2002 Saatavana:
<http://pre20031103.stm.fi/suomi/tiedote/tied02/haotiedote2202.htm> (2.12.2003)
- [STM3, 1999] Sosiaali- terveysministeriö. Sosiaali- ja terveystoimen palvelut. 7.9.1999.
Saatavana:
<http://pre20031103.stm.fi/suomi/pao/julkaisut/vanhuspo/vanhpo6.htm> (2.12.2003)
- [Tampere, 2003] Tampereen kaupungin sosiaali- ja terveystoimen kotisivut. Apuvälinepalvelut. Saatavana:
<http://www.tampere.fi/sote/esh/apuv.htm> (2.12.2003)

Sukututkimusohjelmat

Virpi Tuohisto

Tiivistelmä

Tässä tutkimuksessa pyrin kartoittamaan Suomessa käytettäviä sukututkimusohjelmia. Näkökulmia ovat ennen kaikkea olemassa olevien ohjelmien lähempi tarkastelu, käytettävyys ja käyttäjäkokemukset.

CR-luokat: J.7

Avainsanat: ATK-ohjelmat

1. Johdanto

Sukututkimus on ehkä yksi vanhimmista harrastuksista; ihmisiä on aina kiinnostanut keitä he ovat ja missä on heidän juurensa. Tietotekniikan valtakaudella myös sukututkimus on helpottunut tietoteknisten apuvälineiden myötä, näistä tärkeimpänä sukututkimusohjelmat.

Jos mukaan lasketaan myös ulkomaiset sukututkimusohjelmat, markkinoilla on tarjolla ohjelmia kymmenittäin. Tällaisen ohjelmamäärän edessä käyttäjä on usein hyvin hämillään kun oman ohjelman valinnan aika koittaa. Tässä tutkimuksessa onkin tavoitteena kartoittaa tarjontaa ja perehtyä ohjelmien hyviin ja huonoihin puoliin, niin käytettävyyden kuin standardien noudattamisen näkökulmasta. Lisäksi tutkimus valottaa jonkin verran myös sitä mitä ohjelmien kehittäjiltä odotetaan, nyt ja lähitulevaisuudessa.

2. Yleistä sukututkimuksesta

2.1. Mitä sukututkimus on?

Sukututkimuksesta puhuttaessa monelle tulevat ensimmäisenä mieleen sataa käyvät vanhukset tutkimassa epämääräisiä mikrofilmejä arkistojen takahuoneissa. Varsinainen sukututkimuksen nousukausi oli noin 40 vuotta sitten ja siitä lähtien harrastajien määrä on tasaisesti noussut. Tämän myötä tästä harrastuksesta on tullut myös tavallisten suomenkielisten yhteiskuntaluokkien ja nuorempien väestöryhmien kiinnostuksen kohde -toisin kuin aikaisemmin kun tutkimusta harrastivat lähinnä ruotsinkieliset ja ylemmät yhteiskuntaluokat. Suomen sukututkimusseuran jäsenmäärä on

nykyään noin 6000 [Suomen sukututkimusseura, 2003] mikä kertoo harrastuksen yleisyydestä.

Se miten sukututkimus eli genealogia määritellään, ei olekaan mikään varsin yksinkertainen asia. Sukututkimusta voisikin tavallaan sanoa olevan olemassa niin monenlaista kun on tutkijoitakin: yksi kerää listaa esivanhemmistaan, toinen perehtyy talohistorioihin ja kolmas kerää tietoa perinnöllisistä sairauksista lääketieteellisissä tarkoituksissa. Sampio [2003, s.11] määrittelee sukututkimuksen historiatieteen erikoishaaraksi joka tutkii ihmisten polveutumista ja sukulaissuhteita sekä elämää syntymästä kuolemaan.

Vaikka tutkimusten lähtökohdat ja tavoitteet ovatkin kovin erilaisia, on kaikilla tutkijoilla samat lähtökohdat, tietolähteet ja ongelmat. Lisäksi yhdistävä tekijä on kiinnostus menneisyyteen, syystä tai toisesta. Jollakin syy on yksinkertaisesti kiinnostus omista esivanhemmista, toinen taas haluaa selvittää kotitalonsa tai tilansa asujaimistoa mahdollisimman pitkälle ja toinen taas tutkii perinnöllisiä sairauksia lääketieteellisestä näkökulmasta. Kaikki näistä erilaisista tutkijaryhmistä tarvitsevat jäljennöksiä vanhoista kirkonkirjoista päästäkseen käsiksi menneisyyteen, tavan tallettaa tietoa ja mahdollisesti vaihtaa löytämiänsä yksityiskohtia muiden alaan perehtyneiden kanssa.

2.2. Sukututkimus nykyaikana

Kuten nykyään monet muutkin harrastukset on myös sukututkimus sähköistynyt tietokoneitten lisääntyessä, eikä ihmeekään sillä laajoja tietomääriä käsittelevä tieteenala on omiaan vastaanottamaan tietokantaohjelmia ja muita hallintasovelluksia. Tämän hetkistä muutosta voidaan verrata vaikutukseltaan aikaan jolloin mikrofilmit korvasivat sukututkimuksessa alkuperäiset kirkonkirjat joita ei tietenkään loputtomasti voitu antaa yksittäisen tutkijan käyttöön.

Tietotekniikka ei kuitenkaan ole sukututkimuksessa vain väline joka helpottaa samoja vanhoja rutiineita ja korvaa siniruutuisen vihon ja lyijykynän, vaan tietotekniikka on tuonut myös kokonaan uusia toimintatapoja ja ilmiöitä. Tällaisia ovat muun muassa yksittäisten tutkijoiden laajat kotisivut sukupuineen, on-line tietokanta kirkonkirjoista ja laajat sähköpostiin perustuvat Internetin keskustelupalstat. Näiden uusien ilmiöiden myötä sukututkimus elääkin taas yhtä kultakauttaan: yhä useampi kiinnostuu tästä marginaaliryhmien harrastuksesta, syitä saattavat olla uudet mielenkiintoiset työkalut, tutkimuksen helpottuminen tietokantojen myötä tai vaikkapa Internetin keskusteluryhmien lämmin ja avulias vastaanotto.

3. Sukututkimusohjelmista yleensä

3.1. Mitä sukututkimusohjelmat ovat ja mitä niiltä odotetaan

Palander [2002, s.7] määrittelee sukututkimusohjelman seuraavasti:

”Sukututkimukseen tarkoitetuilla tietokoneohjelmilla eli sukututkimusohjelmilla on helppo hallita suuria tietomääriä. Esimerkiksi valituista lähtöhenkilöistä voidaan tulostaa erilaisia esi- tai jälkipolvitulosteita. Lisäksi voidaan lisätä, muuttaa tai poistaa henkilöitä ja henkilöiden tietoja”

Sukututkimusohjelma voikin periaatteessa olla mikä vain sovellus jonka avulla jopa kymmeniätuhansia henkilöitä sisältävä tietomassa saadaan pysymään järjestyksessä. Tietoa tulee voida muuttaa, poistaa, lisätä ja tietenkin katsoa. Tähän tavoitteeseen päästään tietenkin myös esimerkiksi Access tietokannalla tai hyvin laadittu Excel taulukkokin saattaa riittää, ainakin aluksi.

Kun sukutietoa alkaa pikkuhiljaa kertyä tietokantaan, tietojen tulostus jonkinlaisen raporttina alkaa olla ajankohtaista - viimeistään kun juuri selvinneestä suvun historiasta kiinnostuneet sukulaiset antavat aihetta selkeille tulosteille. Tällaisessa tilanteessa itse koottu tietokanta tai taulukosto saattaa unohtua ja tilalle tarvitaankin järeämpi kaupallinen ohjelma jolla erilaisten havainnollisten raporttien luominen tapahtuu usein kädenkäänteessä.

Jos tutkija päättää hankkia kaupallisen ohjelman, mukaan kuvaan astuu raha. Useimpien sukututkimusohjelmien hinnat liikkuvat noin sadan euron tuntumassa. Kaikki ohjelmat täyttävät varmasti yleisimmät vaatimukset tiedon tallennuksesta, hausta ja raporttien tulostamisesta - tosin yksityiskohtiin mentäessä eroja rupeaakin löytymään yleensä kymmenittäin. Tässä tilanteessa jokainen ohjelmaa valitseva on yleensä hyvin yksin: tarjolla olevista ohjelmista on valittava se mikä tuntuu parhaalta juuri omassa käytössä - se mikä toimii naapurilla, ei välttämättä ole hyvä konsti omassa kodissa.

3.2. Yleisimmät sukututkimusohjelmat

Tällä hetkellä viisi yleisintä Suomessa käytettyä sukututkimusohjelmaa ovat Sukujutut (28.4 % kyselyyn vastanneista), Sukuohjelmisto 2000 sekä sen aikaisempi versio Sukuohjelmisto 8.0 (24.4 %), Juuret (10.3 %), Genus (9.0 %) ja Brother's Keeper (5.1 %) [Tiedot perustuvat kyselyyn josta on lisää luvussa viisi]. Näiden lisäksi muita melko yleisiä ohjelmia löytyy noin kymmenen kappaletta mutta kaikkien ohjelmien joukosta selkeimmin esiin nousevat juuri viisi edellä mainittua.

Seuraavissa viidessä kohdassa esittelen lyhyesti nämä yleisimmät ohjelmat. Tämän jälkeen tässä tutkimuksessa keskitytään pääsääntöisesti vain näihin.

Suuri osa ohjelmien taustatiedoista on löydettävistä lähdeluettelossa mainituilta kotisivuilta.

3.2.1. Sukujutut

Sukujutut on kotimainen ohjelma – kuten varmasti jo nimestäkin saattaa kuulla. Sen ylläpitäjä ja kehittäjä on ATK-palvelu Luhtasaari. Ainakin tähän mennessä ohjelman kehitykseen on todella panostettu ja uusia versioita uusine ominaisuuksineen on saatavilla riittävän usein. Muuta merkittävään positiivista Sukujutuissa on muun muassa sen loistava pärjääminen KotiPC lehden ohjelmatestissä ja ohjelman toimivuus myös vanhemmilla koneilla ja Windowsin versioilla aina 3.1:sta uusimpaan XP:hen. Negatiivista Sukujuttujen yleisilmeessä on lähinnä sen hinta: yleisimpiä ohjelmia vertailtaessa 105€ sijoittuu listan kärkipäähän.

3.2.2. Sukuohjelmisto

Sukuohjelmiston takaa löytyy Kaarle Kaila joka on tehnyt myös gradun sukututkimustiedon käsittelystä, tämän mukaan siis ainakin tuon osa-alueen voidaan olettaa olevan kohdallaan tässä ohjelmassa. Selkeä negatiivisuus tämän ohjelman kohdalla löytyy siitä että aikaisempi versio ei toimi Windows XP:ssä ja uudempi taas vaatii vähintään Windows 2000:n toimiakseen. Hinnan osalta Sukuohjelmisto sijoittuu hiukan keskiarvon yläpuolelle 75 eurollaan.

3.2.3. Juuret

Kolmantena ohjelmana on Consegur Oy:n tuottama Juuret. Juuret on Sukujuttujen lailla melko ymmärtäväinen myös vanhempia tietokoneita kohtaan, tosin vanhojen Windowsien osalta se ei aivan yllä Sukujuttujen yhteensopivuuteen vaan vaatii vähintään Windows 95:n. Viiden suosituimman ohjelman ryhmästä Juuret on ainoa josta on tulossa piakkoin myös melko harvinaiset Linux- ja MacOS yhteensopivat versiot, varsinkin kotimaisille Linux ohjelmille markkinoilla tuntuu olevan selkeä tilaus. Hintaa Juurille kertyy noin 100 euroa.

3.2.4. Genus

Genus on ohjelma jolla on jo pitkät perinteet Suomalaisten sukututkijoiden parissa. Ohjelman takaa löytyy kotimainen yritys nimeltään B.R.A. Tieto. Tilausmahdollisuuksiltaan Genus on selkeästi yksi parhaista, ohjelmatyyppejä on tarjolla useita erilaisia ja esimerkiksi ohjekirjoja ei ole pakko tilata ohjelman mukana kuten joissakin muissa ohjelmissa on. Lisäksi aikaisempien versioiden päivittäinen vaikuttaa kohtuullisen helpolta ja edulliselta. Monipuolisten tilausominaisuuksiensa vuoksi Genuksen hintakin vaihtelee paljon,

ominaisuuksista ja oheistuotteista riippuen hinta vaihtelee 20:n ja 80 euron välillä.

3.2.5. Brother's Keeper

Viiden ohjelman joukkoon mahtuu myös yksin englanninkielinen kokonaisuus, tämä on Brother's Keeper. Ensitutustumisella Brother's Keeperin yleisimago ei selviä kovinkaan helpolla ja ohjelman kotisivulta on vaikeaa löytää muuta kuin tilaus- ja päivitystietoja. Brother's Keeperin hinta on 45 dollaria, kuten sivuilla pilke silmäkulmassa todetaan ohjelman voi joskus saada halvalla jos valuuttakurssit heiluvat. Jos englanninkielinen ohjelma ei pelota ja rohkeus ja taidot riittävät ohjelman tilaamiseen ulkomailta on Brother's Keeperin varmasti oikein vartenotettava vaihtoehto.

4. Ohjelmien käytettävyys

4.1. Yleistä

Kaikista ohjelmista tai ohjelmaryhmistä puhuttaessa käytettävyys on aina yksi mielenkiintoinen tarkastelukohde. Onneksi nykyään käytettävyyteen, eli siihen kuinka helppoa ja loogista ohjelman käyttö on, kiinnitetään yhä useammin huomiota – ja onhan tämä ohjelman valmistajankin edun mukaista: käyttäjät pysyvät pidempään helppokäyttöisen ohjelman parissa.

Sukututkimusohjelmissa käytettävyyden kannalta olisi tärkeää päästä nopeasti ja helposti alkuun uuden tietokannan luomisen kanssa, lisäominaisuuksiin puolestaan tutustutaan yleensä vasta siinä vaiheessa kun tietokannassa on jo jonkin verran tietoa ja näin myös ohjelma on jo ehtinyt tulla jonkun verran tutuksi. Toisaalta myös suurien kokonaisuuksien, tässä tapauksessa sukupolvien – ja linjojen hahmottaminen pitäisi olla helppoa, eli joissakin rajoissa: sitä parempi mitä enemmän näytöltä voi kerralla nähdä.

4.2. Viiden yleisimmän sukututkimusohjelman käytettävyys

Seuraavassa on esitelty pääpiirteitä kunkin viiden ohjelman perusnäyttöjen käytettävyydestä. Huomiota on kiinnitetty erityisesti siihen kuinka helppoa alkuun pääseminen kunkin ohjelman kanssa on. Osa tekstistä pohjautuu Nielsenin heuristiikkoihin jotka löytyvät liitteestä 10.1.

4.2.1. Sukujutut 9.9

The screenshot shows the 'KESKUSIKKUNA' software interface. The title bar indicates the path 'c:\program files\sukujutut'. The main window contains several sections:

- Navigation:** 'Tiedosto', 'Muokkaa', 'Näytä', 'Tulosta', 'Sekalaiset', 'Optiot', 'Näyttö', 'Lähde', 'Talo', 'Kielivalinnat', 'Ohje'.
- Search and Filter:** 'Uusi' button, dropdown menu with 'Mies', 'Tuntematon', 'Nainen', and navigation arrows. A counter shows '2336 / 3319'. 'Hae' and 'Etsi' buttons.
- Personal Information:** 'Etunimet' (Kustaa, Kustaanpoika), 'Sukunimi' (Wirman), 'Muu nimi'. Buttons for 'Isä', 'Pura', 'Äiti'. 'Syntynyt' (30.6.1849), 'Kuollut' (2.3.1886, Tampere), 'Arvo ja ammatti' (Puuvillatehtaan työmies), 'Kieli' (Ei koodia).
- Family Information:** 'Puolisot' section with 'Uusi P', 'Kytke 1', 'Pura 8', 'Näytä 3', '1/1 ja Puoliso 1/1', and 'Perheteksti' button. 'Vihity' dropdown, '3.8.1872', 'Eronneet' field.
- Children:** 'Lapset' section with 'Poika', 'Tytär', 'Kytke 2', 'Pura 9', 'Näytä', 'Vaihda', '6 kpl', and 'Valitun lapsen lapsia'.

Family members listed:

- + Maria Lovisa Rosberg, Wirman, s. 27.4.1842 Sääksmäki Voipaala - (2337)
- + Kustaa Edvin Wirman, Vaaja s. 5.10.1872 Messukylä, k. 20.4.1914 T.
- Aleksandra Maria Wirman s. 8.2.1874 Messukylä - (2344)
- Iida Josefiina Wirman s. 13.12.1877 Tampere, k. 25.4.1890 Tampere
- Arvid Aleksanteri Wirman s. 4.9.1880 Tampere - (2346)
- August Jalmar Wirman s. 1.4.1882 Tampere - (2347)
- Fanny Aleksandra Wirman s. 28.2.1885 Tampere - (2348)
- + Reino Edvin Vaaja s. 22.10.1903 Tampere.

Bottom status bar: 'Lasten järjestystä perheessä muutetaan hiiren oikealla näppäimellä' and 'Päivitetty 13.8.2003 15:24'.

Kuva 4.2.1-1 Sukujuttujen perusnäyttö

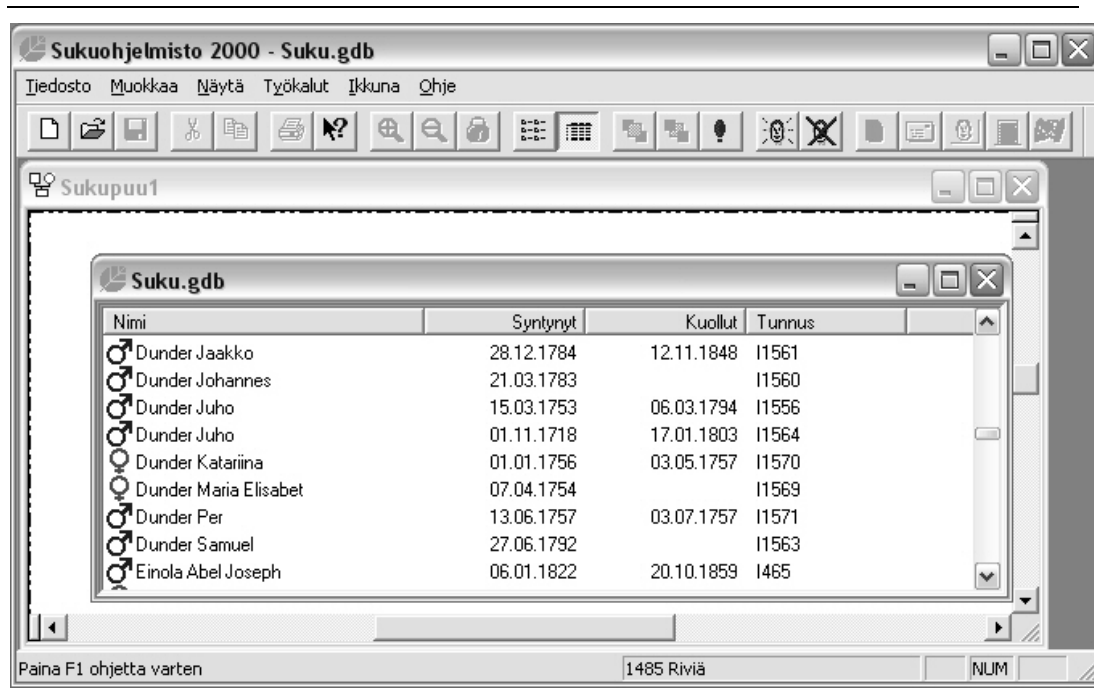
Sukujuttujen yleisilme ei ole kovinkaan koreileva, mutta sitäkin toimivampi: ohjelman avauduttua käyttäjän edessä on kaikki mitä hän tarvitsee. Kuten kuvassa 4.2.2-1 näkyy, pääikkuna sisältää tyhjiä nimettyjä tyhjiin kenttä joihin lisätään henkilön tiedot, tämän jälkeen painikkeella "isä" isän tiedot ja samaa tietojen lisäys kaavaa voidaan sitten periaatteessa toistaa loputtomiin, oli kyseessä sitten lapsi, aviopuoliso tai muu perheenjäsen. Erehdyksiä on kutakuinkin mahdotonta tehdä. Lisäksi tietojen syöttämistä nopeuttavat pikanäppäimet ja tallennuskin tapahtuu kätevästi aina kentästä toiseen siirryttäessä.

Perushenkilötietojen lisääminen ohjelmaan on siis todella helppoa, Sukujuttujen ongelma piileekin monimutkaisemmissa toiminnoissa jotka on sijoitettu valikkoriville hieman epämääräisiin järjestyksiin. Muutenkin valikot saavat pienen negatiivisen maininnan koska yksikään niistä ei noudata totuttua Windows tapaa jossa Tiedosto valikko on ensimmäisenä sisältäen hallintatyökalut, Muokkaa seuraavana kopio, liitä ynnä muilla toiminnoillaan varustettuna.

Ikkunan asettelun puolesta Sukujutut on melko looginen: käyttäjä näkee normaalisti jopa 5 sukupolvea yhtä aikaa; kaksi kulloisenkin päähenkilön molemmin puolin. Kuten aikaisemmin on mainittu, tämä on joskus suorastaan loistava ominaisuus, joskus taas se saattaa tehdä näytöstä sekavan ja hankalasti hahmotettavan.

Kaikin puolin Sukujuttujen käyttäminen vaikuttaisi olevan helppoa ja mukavaa. Perustoiminnot toimivat hyvin ja valikoita ja niiden sisältöäkin oppii ajan myötä käyttämään kohtuullisen sujuvasti

4.2.2. Sukuohjelmisto 2000



Kuva 4.2.2-1 Sukuohjelmisto 2000:n perusnäyttö

Yllä oleva kuva 4.2.2-1 on sukuohjelmiston perusnäytöstä. Perusnäyttö on lupaava monine Windows sovelluksista tuttuine painikkeineen, lisäksi valikot noudattavat yleisiä sääntöjä niin asettelultaan kuin sisällöltään. Kun kuvakkeita tutkii tarkemmin, kaikki niistä eivät kuitenkaan ole niin loogisia.

Oletustilassa Sukuohjelmisto avaa eteen tyhjän A4 arkkia muistuttavan valkoisen alueen, tästä eteenpäin käyttäjä onkin sitten omillaan. Jonkin aikaa kestävän etsimisen jälkeen työkalurivillä oleva uuden henkilön lisäämistä tarkoittava painike todennäköisesti löytyy ja näin päästään itse asiaan, henkilötietojen syöttöön. Kun henkilön tiedot on syötetty alkaakin seuraavan ongelman selvittely: kuinka tälle juuri luodulle henkilölle lisätään vaikkapa äiti? Ongelman ratkaisu ei olekaan mikään helppo eikä itsestään selvä, vaan enemmänkin päinvastainen. Tästä siis iso miinus.

Lopputuloksena Sukuohjelmistosta jää hieman epämääräinen tunne: ohjelmassa on selkeää yritystä myös käytettävyyden kannalta mutta jotenkin toteutus tuntuu jääneen puolitiehen: varsinkin käyttöliittymän helpon ja nopean opittavuuden puolesta. Ohjelman sujuvasta käyttöönotosta ei varmasti selviä useinkaan ilman ohjekirjaa tai hyvää neuvontaa.

4.2.3. Juuret 2.1

The screenshot shows the 'Juuret v. 2.1' software window. The title bar reads 'Juuret v. 2.1 - Suku.jrt - [Laatu, Elisabet, * 1707, Akaa Saviniemi]'. Below the title bar is a menu bar with 'Tiedosto', 'Muokkaa', 'Asetukset', 'Toiminnot', 'Ikkuna', and 'Ohjeet'. A toolbar contains icons for file operations. The main interface has tabs for 'Haku', 'Henkilö', 'Sukupuut', 'Vanhemmat', 'Puolisot', 'Lapset', 'Multimedia', and 'Talot'. The 'Henkilö' tab is active, showing a form for 'Sivu 1' with the following fields:

Sukunimi Laatu	Etilite	Eunimi Elisabet			<input type="radio"/> Mies <input checked="" type="radio"/> Nainen <input type="radio"/> Ei tietoa
Avionimi	Etilite	Isän etunimi Marti			
Edellinen sukunimi	Etilite	Äidin etunimi Vappu			1912/1485
Syntymäaika 23.12.1707	Pitää / Kaupunki Akaa Saviniemi	Kvlä / Kaupundinosa	Talo / Katu	Nro	Maa
Kasteaika	Pitää / Kaupunki	Kvlä / Kaupundinosa	Talo / Katu	Nro	Maa
Riipiaika	Pitää / Kaupunki	Kvlä / Kaupundinosa	Talo / Katu	Nro	Maa
Mhki aika	Pitää / Kaupunki	Kvlä / Kaupundinosa	Talo / Katu	Nro	Maa
Kuolinaika	Pitää / Kaupunki	Kvlä / Kaupundinosa	Talo / Katu	Nro	Maa
Hautausaika	Pitää / Kaupunki	Hautausmaa	Kuolinsvv		Maa
Lähiosoite	Postinumero	Postitoimipaikka	Maa		
Koulutus	Sotilasarvo	<input type="checkbox"/> Elää	<input type="checkbox"/> Tiedot tarkistettu	Kieli <input checked="" type="radio"/> Suomi <input type="radio"/> Ruotsi	
Ammatti	Silmien väri	Hiuksen väri	<input type="checkbox"/> Ei suvunjatkaja		
			<input type="checkbox"/> Adoptiolapsi		

At the bottom of the form are buttons for 'Uusi', 'Poista', 'Edellinen', and 'Seuraava'.

Kuva 4.2.3-1 Sukuohjelmisto 2000:n perusnäyttö

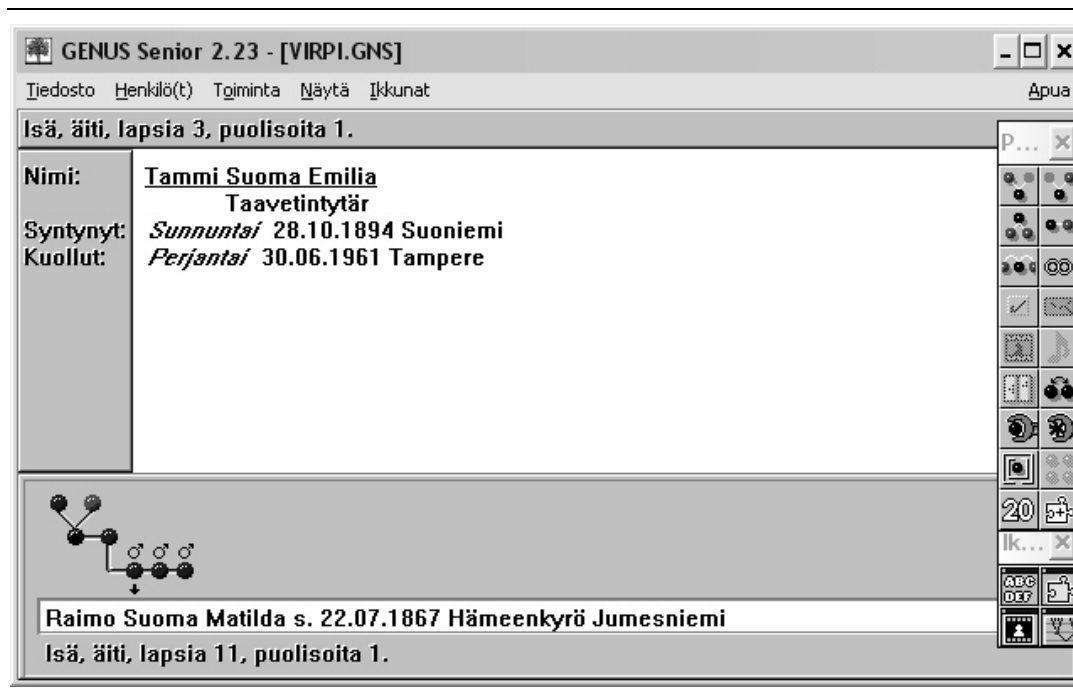
Kuten myös Sukuohjelmistossa Juurissa huomio kiinnittyy heti siihen kuinka tyyppillisen Windows sovelluksen näköinen se on, kuten kuvasta 4.2.3-1 voidaan nähdä, painikkeet ja valikot noudattavat tuttua ja turvallisen tuntuista kaavaa. Toinen hyvä, Sukujuttujen kanssa yhteneväinen, piirre on perusnäyttö, joka sisältää tyhjiä lomakkeeseen sijoitettuja kenttiä joita voi välittömästi alkaa täyttämään valitsemansa henkilön tiedoilla. Alkuun pääseminen ei tällä ohjelmalla todellakaan kestä kauaa.

Negatiivisiakin puolia toki muuten niin hyvästä ohjelmasta löytyy. Muutamalla harhaklikkauksella käyttäjä saattaa käynnistää hyvinkin pitkäkestoisia raskaita toimenpiteitä joita ei voi edes keskeyttää ja joissakin tilanteissa toimenpiteen edistymisestä kertovassa ikkunassa ei edes ilmoiteta

että mitä ohjelma parhaillaan tekee. Tämän kaltainen virhe rikkoo selkeästi periaatetta järjestelmän tilan näkyvyydestä vastaan. Toinen negatiivinen puoli on välilehtien ja niille sisällytettyjen näkymien ja toimintojen paljous, käyttäjän on kohtuullisen helppoa eksyä välilehtiäidakkoon osaamatta enää tietä takaisin kotiin.

Kokonaisuudessaan Juuret vaikuttavat hyvin suunnitellulta ja helppokäyttöiseltä ohjelmalta omista esitiedoista riippumatta. Esimerkiksi lomakenäkymän suunnittelussa on otettu hyvin huomioon se että tällaisella perinteisen arkielämästä tutun paperilomakkeen kaltaisella syöttöruudulla kynnys käytönaloittamiseen pienenee huomattavasti.

4.2.4. Genus 2.23



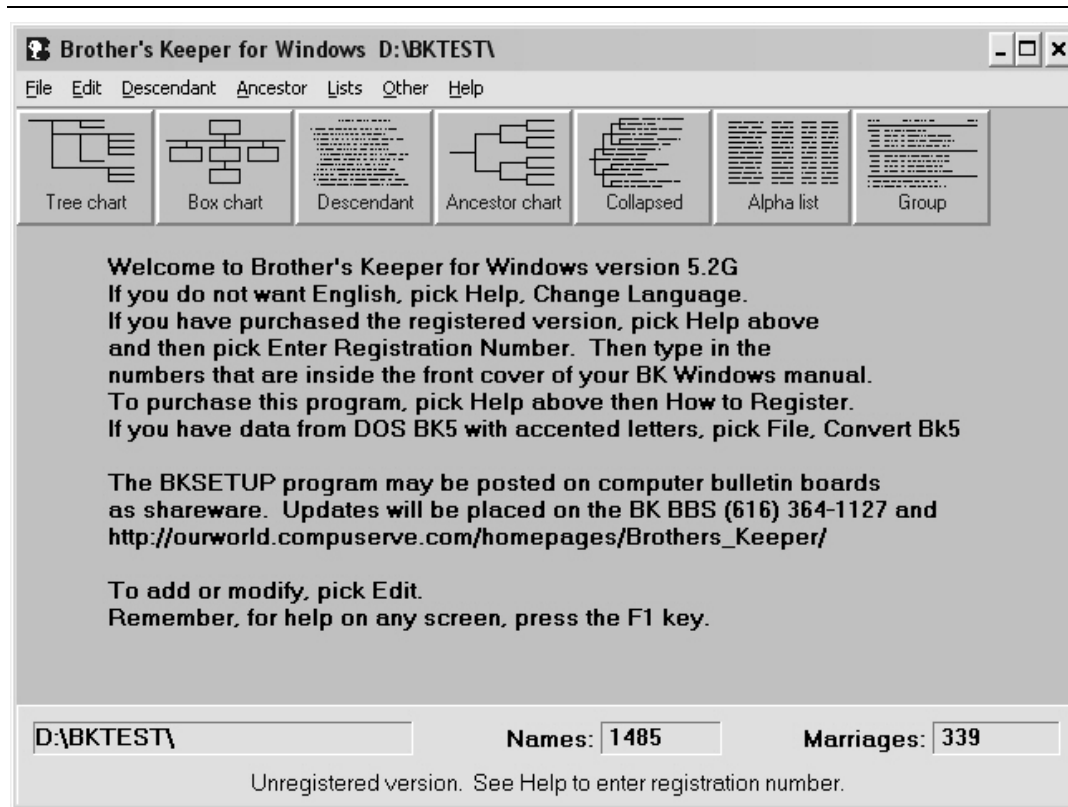
Kuva 4.2.4-1 Genuksen perusnäyttö

Kuva 4.2.4-1 esittää Genuksen päänäyttöä. Ensimmäisellä vilkaisulla Genus näyttää mukavalta joskin ehkä hiukan vanhanaikaiselta. Tarkemman tarkastelun jälkeen ongelmaksi nousee kuitenkin irrallisena leijuva työkaluvalikko jonka kuvakkeet eivät ole kovinkaan selkeitä eikä niillä ole minkäänlaisia tekstiselitteitä, tässä tilanteessa on siis pakko turvautua joko ohjekirjaan tai kokeilla jokainen painike yksitellen läpi. Lisäksi päänäytössä hämääviä elementtejä ovat muun muassa epämääräisesti toimivat navigointiin käytetyt pallot jotka muodostavat jonkinlaisen sukupuun ja alleviivattu nimikenttä joka näyttää epäilyttävän paljon linkiltä.

Kun ohjelman käytössä pääsee jotenkuten alkuun, eivät ongelmat kuitenkaan lopu. Yksi hämmennystä aiheuttava yksityiskohta on se että yllättäen pääikkunan takaa löytyykin uusia ikkunoita, eli näiden myötä edessä on taas lisää selvitystyötä ohjelman käytöstä. Kaiken kaikkiaan käyttöliittymästä jää sekava vaikutelma, yksinkertaisemmilla valikoilla ja harvemmillä ikkunoillakin olisi varmasti tultu toimeen.

Hyvää Genuksessa on muun muassa selkeät ja hyvin ryhmitellyt valikot, tosin apu tarvittaessa kyseisen valikon löytäminen oikeasta reunasta voi kestää hetkisen. Kokonaisuudessaan Genus on mukava paketti josta varmasti saisi vielä paremman nykyaikaistamalla sitä ja samalla liittämällä huomiota peruskäyttäjän havainnointikykyyn ja yleisiin toimintatapoihin

4.2.5. Brother's Keeper



Kuva 4.2.5-1 Brohter's Keeper ohjelman pääikkuna

Brother's Keeper ohjelmassa ensimmäinen käytettävyysongelma saattaa tietenkin olla kieli, mutta koska tällaiset ongelmat ovat niin suhteellisia, jätän kieleen liittyvät seikat tässä huomioimatta. Vaikka Brother's Keeper tosin on siinä määrin miellyttävä ohjelma että siihen voi hankkia kielipaketteja joiden myötä ohjelmaa voi käyttää haluamallaan kielellä.

Ensimmäinen silmäys Brother's Keeperiin antaa mukavan yleiskuvan: ohjelma toivottaa tervetulleeksi tekstillä joka on nähtävissä kuvassa 4.2.5-1, tämän jälkeen voikin siirtyä katselemaan suuria ja selkeitä painikkeita tai siististi järjestettyjä perinteitä noudattavia valikoita. Tässä kohdassa tilanne on kuitenkin huonompi kuin miltä näyttää: hetken tutkailun jälkeen käyttäjä todennäköisesti huomaa että hänellä ei ole minkäänlaista aavistusta miten saisi aloitettua uuden tietokannan ja lisättyä siihen henkilöitä.

Kun tietokantaan on sitten saatu lisäiltyä jokunen henkilö päästään seikkailemaan valikoihin ja muihin toimintoihin, tämäkään ei välttämättä tuo mukanaan erityisen mukavia kokemuksia. Esimerkiksi yksi melko yleinen tilanne on se että ohjelmassa valitaan joku toiminto, jos toimintoa ei halutakaan suorittaa päädytään peruutuspainikkeella näyttöön jollaista käyttäjä ei ole koskaan nähnytkään - eksyminen tai jonkun asteinen hätäntyminen saattaa siis onnistua hyvinkin helposti. Tämän lisäksi toimintoja sisältävissä ikkunoissa painikkeiden paikat eivät tunnu noudattavan minkäänlaista kaavaa vaan seikkailevat ylös ja alas, vasemmalta oikealle.

Pitkällistenkin yritysten jälkeen Brother's Keeper tuntuu melko toivottamalta tapaukselta. Kuten joissakin muissakin tapauksissa tämäkin ohjelma vaatii useimmiten rinnalleen käyttöoppaan tai jopa opettajan, toimintojen läpikäyminen kohta kohdalta erehtyen ja eksyen on aivan suuri urakka ilman tukea suoritettavaksi.

4.3. Yhteenvetoa käytettävyydestä

Ohjelmia verrattaessa niistä löytyi suuriakin eroja: joissakin kutakuinkin kaikki tuntui olevan riittävän hyvin kohdallaan, toisissa taas mikään ei toiminut niin kuin voisi olettaa. Muutamissa ohjelmissa oli melko kiitettäviä lopputuloksia saatu aikaan jo pelkästään tyypillisiä Windowsin asetteluja noudattaen, toisaalta taas arkielämän kaavakkeita muistuttavat lomakenäkymät olivat omiaan auttamaan alkuun ja alentamaan käyttökyynnystä.

Mikko Hartikainen ja Saija Patomäki esittelevät toiminnan teorian yhteydessä [Raisamo (toim.), 2002, s.141] tavoitteellisuuden periaatetta, tämän periaatteen mukaan tavoite on kohde jonka saavuttamiseksi muu työ tehdään, eli yksittäiset toiminnot ja välivaiheet eivät ole varsinaisia tavoitteita vaan työkaluja laajemman tavoitteen saavuttamisessa. Edellä mainittu pätee hyvin myös nyt käsiteltäviin ohjelmiin: ohjelman opettelusta ja käytöstä ei missään vaiheessa saisi tulla itse tarkoitus vaan näiden tulisi olla vain avustavia työkaluja sukuselvitystä tehtäessä.

5. Käyttäjäkokemukset

5.1. Taustatietoa kyselystä ja siihen vastanneista henkilöistä

Kyselyyni valitsin vastaajiksi yleisimpien sukututkimukseen liittyvien keskustelupalstojen jäsenet. Kyselylomakkeen vastaajat saivat käyttöönsä www-sivulta. Lomakkeella heiltä kyseltiin muun muassa taustatiedot kuten ikä ja sukupuoli, mitä sukuohjelmaa he käyttävät ja kuinka he päätyivät juuri kyseiseen ohjelmaan. Tavoitteena näillä kysymyksillä oli selvittää ketkä sukututkimusohjelmia käyttävät ja millä perustein käyttäjät valitsevat ohjelmansa sekä kuinka oikeaan valinta on osunut.

Vastauksia sain kyselyyni 82 kappaletta joista kaksi jouduin hylkäämään liian puutteellisten tietojen vuoksi. Näistä kahdeksastakymmenestä sukuohjelmien käyttäjästä naisia oli 26 prosenttia ja miehiä 74, ikäjakauma puolestaan suurin ryhmä oli selkeästi 45–69-vuotiaat. Tutkimustuloksia katsottaessa on kuitenkin muistettava, että kyselyyn vastanneet henkilöt käyttävät säännöllisesti Internetiä ja muita sähköisiä palveluita. Jos kyselyyn taas olisivat vastanneet myös henkilöt jotka eivät edes kunnolla tiedä mikä Internet on, olisivat tulokset saattaneet olla jossain määrin erilaisia. Perinteisten käsitysten mukaan tämä olisi saattanut lisätä esimerkiksi yli 69-vuotiaiden naisten osuutta vastaajista.

5.2. Kyselyn tulokset

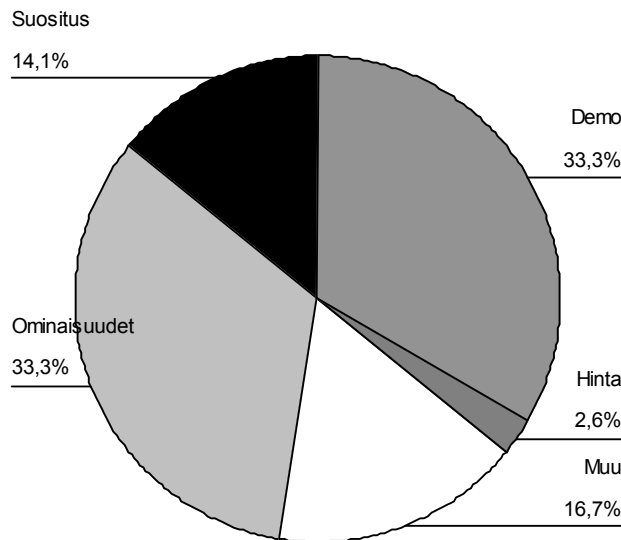
5.2.1. Käytetyimmät ohjelmat

Kuten jo kohdassa 3.2 on todettu, yleisimmät käytetyt ohjelmat vastaajien parissa ovat Sukujutut, Sukuohjelmisto, Juuret, Genus ja Brother's Keeper. Näistä erityisesti Sukujutut ja Sukuohjelmisto olivat suuren joukon suosiossa. Näiden viiden ohjelman jälkeen listalla seuraavien ohjelmien prosenttiosuudet ovat huomattavasti pienempiä. Osalle erikoisemmista ohjelmavalinnoista syyt olivat lähestulkoon itsestäänselvyksiä; tällaisia tekijöitä ovat muun muassa poikkeava käyttöjärjestelmä kuten Linux tai MacOS, suomenruotsalaisuus jolloin ohjelmaa halutaan käyttää ruotsiksi tai yhteensopivuus jonkun toisen harvinaisemman ohjelmiston kanssa. Toisaalta taas jotkut vastaajista olivat käyttäneet samaa ohjelmaa jo yli kymmenen vuotta ja halusivat jatkaa ohjelman käyttämistä vaikka se olisi poistunut suurilta markkinoilta jo ajat sitten ja kehitystyö olisi suurelta osin päättynyt.

5.2.2. Ohjelman valinnan perusteet

Kyselylomakkeessa olin antanut valmiiksi vaihtoehtoja joilla ohjelma on valittu, näitä olivat ohjelman hinta, ystävältä tai muulta taholta saatu suositus,

demoversio jonka kautta ohjelma on havaittu hyväksi juuri omat käyttötarkoitukset huomioon ottaen, laajat ominaisuudet tai muu syy jonka vastaaja sai täyttää erilliseen tekstikenttään. Alla olevassa kuvaajassa 5.2.2-1 on erilaisten vastausten jakauma.



Kuvaaja 5.2.2-1 Ohjelman valinnan perusteet

Kuten kaaviosta selviää useimmat vastaajista ovat valinneet ohjelmansa demon tai tiedossa olevien ominaisuuksien perusteella. Toisaalta tällaisessa tilanteessa herää kysymys mistä tiedot ominaisuuksista ovat lähtöisin jos demoa ei ole ollut. Tätä kautta kohta "ominaisuus" liittyy läheisesti myös luokkaan demo eikä vaihtoehto "suositukseen" varmasti ole näin ajatellen täysin tuulesta temmattu. Oli niin tai näin on kuitenkin hienoa että ihmiset luottavat omiin kokemuksiinsa ja tuntemuksiin ohjelmaa valittaessa.

Seuraavaksi eniten ohjelmia on valittu perustuen muihin syihin, näitä syitä ovat pääsääntöisesti olleet toisten ohjelmien yhteensopimattomuus, kielivalinnat ja jossain määrin puhdas sattuma. Toisaalta taas tähän ryhmään kuuluvat myös ne käyttäjät jotka ovat tehneet ohjelmansa itse puhtaasta mielenkiinnosta tai saadakseen käyttöönsä varmasti juuri itselleen tärkeimmät ominaisuudet.

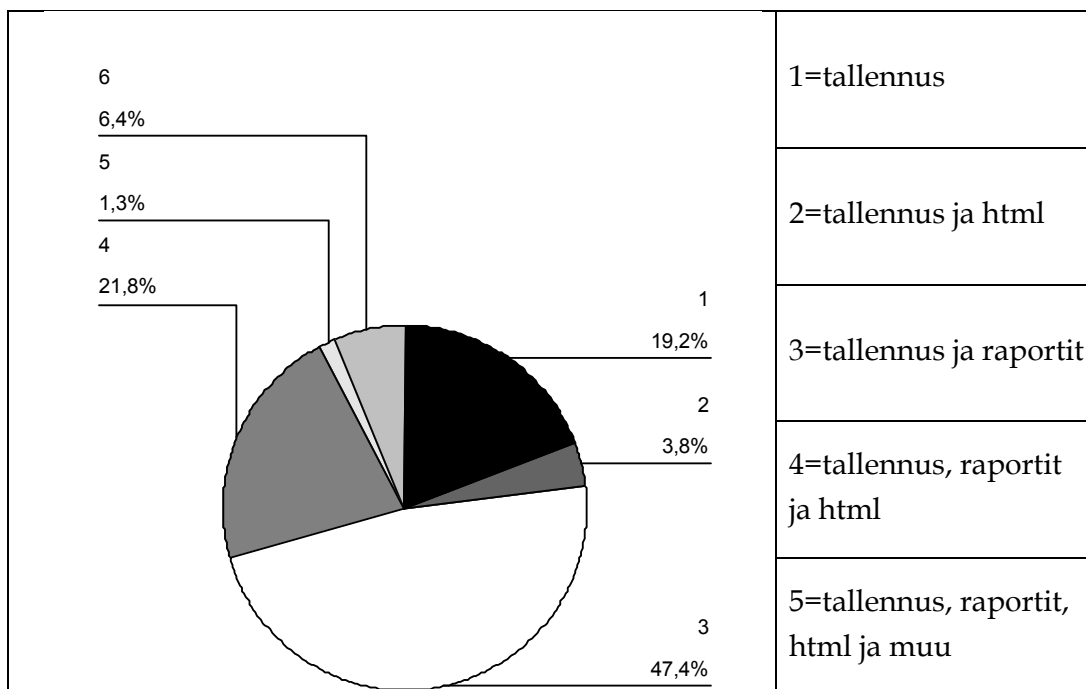
Neljänneksi tärkein syy valinnassa on suositus. Tämä suositus on saattanut tulla ystävältä, puolilitutulta sukututkijalta tai esimerkiksi KotiPC lehden järjestämän ohjelmatestin tuloksista. Kaikkia näitä suurempia ja

voimakkaampia suosittelijoita ovat kuitenkin varmasti Suomen sukututkimusseura tai erilaiset paikallisjärjestöjen pitämät sukututkimuskurssit. Sukututkimusseuran arvosteluihin tutkijan on helppoa ja turvallista luottaa. Toisaalta taas monella sukututkimuskurssilla opetellaan kädestä pitäen jonkun ohjelman käyttö ja kukapa sitä haluaisikaan heti siirtyä opettelemaan uuden erilaisen ohjelman käyttöä kun yksi on juuri saatu sujuvasti hallintaan.

Vähiten painoarvoa valintatilanteessa on saanut ohjelman hinta. Tämä saattaa johtua siitä että ohjelmien hinnoissa ei ole valtavia eroja, noin 20 euron heitto suuntaan tai toiseen ei yleensä ole taloudellisesti vaikuttava - varsinkaan silloin jos kaikki muu ohjelmassa vaikuttaisi olevan kohdallaan. Toisaalta taas hankinta kuten hyvä sukututkimusohjelma tehdään yleensä vai kerran tai pari, joskus ei yksikin kerta riitä kymmeniksi vuosiksi.

5.2.3. Ohjelmien käyttötavat

Puhuttaessa ohjelmien valinnasta on tietenkin myös tiedettävä mihin ohjelmaa tulisi voida käyttää. Tässä kohdassa valittavina olivat sukutietojen tallennus, raporttien tulostus, HTML-sivujen tuottaminen ja muu itse määriteltävä käyttötarkoitus. Näistä vaihtoehtoista oli mahdollista valita myös useampia. Alla olevassa kuvaajassa 5.2.3-1 on listattu esille tulleet käyttötarkoitukset ja niiden yhdistelmät.



	6=tallennus, raportit ja muu
--	------------------------------

Kuten kuvaaja osoittaa suurin osa vastaajista käyttää ohjelmaansa melko perinteisiin toimintoihin; tietojen tallennukseen ja niiden tulostukseen. Tämä on täysin luonnollinen tulos koska jossain määrin sukuohjelmat ovat ottaneet paikkansa lähinnä kotikortistojen ja vihkoarkistojen edistyneempinä korvaajina, ainoa huomattava ero edellä mainittuihin on se että tiedon muokkaaminen ja järjestyksessä pitäminen on tietotekniikka hyödyntäen paljon helpompaa, nopeampaa ja vaivattomampaa. Tämän lisäksi mahdollisuus erilaisten kaavioiden, sukupuiden, tilastojen ja jopa kokonaisten sukukirjojen tulostukseen on selkeä etu verrattuna vanhoihin tiedontallennustapoihin.

Yksi tietotekniikan mukanaan tuomista uusista ilmiöistä on käyttötarkoituksissakin kakkossijalle sijoittuva Internet ja sen myötä ilmaantunut tarve erilaisille kotisivuille ja tätä myötä tietenkin myös sukututkimuksista kertoville www-sivuille. Internetissä tietojen julkaisu on huomattavasti helpompaa ja joustavampaa kuin perinteisten sukukirjojen julkaisu. Sukututkimus on siitä poikkeava www-julkaisujen ala että siinä käsin koodaaminen ei yleensä tule kysymykseenkään: tavoitteena oleva HTML-sivu saattaa tulla sisältämään jopa satoja tai tuhansia henkilöitä. Tällaisessa tilanteessa onkin mukavaa antaa tehtävä ohjelman hoidettavaksi.

Seuraavaksi suurin käyttötarkoitusr ryhmä on pelkkä tiedon tallennus. Tässä tilanteessa ohjelmalta ei odoteta muuta kuin että se pitää tiedot järjestyksessä ja niitä saa helposti lisättyä tietokantaan. On melko yllättävää, että käyttäjiä joiden käyttötapoihin kuuluu vain edellä mainittu, on näinkin paljon. Sitä mistä tämä ilmiö on lähtöisin, on vaikeampi arvioida. Voitaisiin kuitenkin päätellä, että käyttäjä on vielä epävarma ohjelmansa kanssa, eikä halua tai mielenkiintoa astetta kehittyneempiä ominaisuuksia kohtaan tästä syystä ole vielä löytynyt. Toisaalta taas on mahdollista että joillekin tutkijoille riittää kotikoneella oleva siisti henkilötietokanta jonka tulostaminen paperille ei ainakaan vielä ole tullut ajankohtaiseksi.

Loput käyttötarkoituksista muodostavat yhdessäkin melko pienen joukon. Kaksi näistä sisältää käyttötarkoituksen "muu" joka sisältää muun muassa GEDCOM tiedostojen luonti joiden avulla tietoa voidaan helposti vaihtaa ohjelmasta toiseen ja tutkijalta toiselle. Näistä kahdesta ryhmästä toinen käyttää ohjelmaansa kaikkiin viiteen tarkoitukseen, eli ovat todellisia sukututkimusohjelmien tehokäyttäjiä. Toisaalta taas kolmas ryhmä sisältää käyttäjät jotka pelkästään tallentavat tietoa ja luovat niihin perustuvia www-

sivuja, tästä ryhmästä löytyvätkin varmasti jonkinlaiset sukututkimusmaailman edelläkävijät.

5.2.4. Arviot omasta ohjelmasta

Numeroasteikko nelosesta kymmeneen on kaikille tuttu ja perinteinen arviointimenetelmä, niinpä valitsin sen yhdeksi arviointitavaksi kyselykaavakkeekseni. Jokaisen vastaajan tuli kuvata tällä asteikolla tyytyväisyyttään käyttämäänsä ohjelmaan. Arvosanoissa ei missään kohdassa ilmennyt suurta hajontaa ja keskiarvoksikin muodostui peruspositiivinen 8.3. Kaikki muut viidestä pääohjelmasta saivat keskiarvosanakseen kahdeksan, Sukujutut ja Brother's Keeper puolestaan yhdeksikön.

Kokonaisuudessaan vastaajat tuntuivat siis olevan kohtuullisen tyytyväisiä valintaansa. Vain kohtuullisen harvat (19.2 %) käyttävät useampia ohjelmia täydentämään toisiaan. Keskimääräinen aika jonka ohjelma on ollut käytössä, on noin viisi vuotta mikä on pitkä käyttöaika mille tahansa yhdelle ohjelmalle. Vaikka jotkin vastaajat ilmoittivat käyttäneensä valitsemaansa ohjelmaan jo yli kymmenen vuotta, saattoivat toiset taas olla juuri harkitsemassa kohtuullisen uuden ohjelman vaihtoa esimerkiksi päivitysversioiden hitaan ilmestymistahdin, epävakauden, huonon luotettavuuden tai laajojen yhteensopivuusongelmien vuoksi.

5.3. Johtopäätöksiä kyselystä

Tyypillinen kyselyyn vastaaja oli siis yli 45-vuotias mies joka on käyttänyt Sukujutut ohjelmaa noin viisi vuotta. Aikanaan hän valitsi ohjelman perustuen sen laajoihin ominaisuuksiin tai demon myötä kertyneisiin hyviin kokemuksiin. Ohjelmaansa hän käyttää pääsääntöisesti tallennukseen ja raporttien ja kaavioiden tulostukseen.

Vaikka sukututkimus on viimeisinä vuosikymmeninä löytänyt tiensä myös nuorempien ihmisten harrastuslistaan, on se kuitenkin kyselyynkin pohjautuen edelleen selkeästi keski-ikäisten ja sitä vanhempien harrastus. Tämä onkin yksi näkökulma joka tekee kyseisistä ohjelmista melko mielenkiintoisen: käyttäjäryhmään kuuluu paljon jopa eläkeikäisiä, mutta ohjelman sekä sen myötä muun tietotekniikan hyödyntäminen ei suinkaan vaikuta, vasten yleisiä oletuksia, olevan heille mikään suuren luokan uudenaikainen mörkö vaan luonteva työkalu joka auttaa omassa harrastuksessa.

Vesa Mäensivu [2002, s.154] pohtii digitaalista kahtiajakautumista ja sen todennäköisyyttä. Tämän pohdinnan mukaan suurimpia riskejä tietoyhteiskunnan kehityksessä on jonkun väestöryhmän ajautuminen digitaalisen maailman ulkopuolelle, riskiryhmiin kuuluvat ennen kaikkea vanhukset joille uuden oppiminen on hankalampaa. Kahtiajakautumisen

näkökulmasta sukututkimusohjelmien käyttäjätulos paljastaa melko positiivisia tuloksia: suuri osa vastaajista oli yli 50-vuotiaita jotka käyttävät aktiivisesti tietokonetta, opiskelevat uusia ominaisuuksia ja hallitsevat vähintään kohtuullisen hyvin jo oppimansa asiat. Tältä pohjalta historiaan pohjautuva harrastus voikin siis olla suureksi avuksi myös 2000-luvun teknologiayhteiskunnassa.

6. Standardit

Kuten minkä tahansa muunkin ohjelman tulisi myös sukuohjelmien ja niiden tuottamien tulosteiden olla mahdollisimman yhteensopivia muiden sovellusten kanssa ja seurata tarpeellisilta osin yleisiä standardeja. Sukututkimusohjelmien kyseessä ollessa esiin nousee kaksi yksityiskohtaa joihin tulisi kiinnittää huomiota tästä näkökulmasta, näitä ovat GEDCOM-tiedostot (GEDCOM muodostuu sanoista Genealogical Data Communication) jotka on tarkoitettu tiedonsiirtoon eri ohjelmien välillä ja html-tiedostot joita hyödyntäen sukutaulustoja voidaan julkaista vaikkapa omalla kotisivulla.

6.1. GEDCOM

Kaarle Kaila esittelee Pro gradu tutkielmassaan [2002] GEDCOM:ia melko perusteellisesti. Tutkielmaan perustuen tämä tiedostomuoto on tiivistettynä kehitetty sukutietojen siirtämiseen ohjelmasta tai koneesta toiselle. Alkunsa kehitys sai Mormonikirkon perhehistorian osastolla 1987, kun ensimmäinen versio (3.3) standardista julkaistiin.

Vuonna 2004 GEDCOM:sta on käytössä versio 5.5, versioiden myötä tiedostoformaattiin on tullut muutoksia mutta perusidea on säilynyt samana. Kyseisen version kotisivulle on listattu tiedostotyyppin sisältämät kentät ja niiden käyttötarkoitukset. Ensimmäisellä vilkaisulla tämän kaltainen listaus vaikuttaa melkoisen monimutkaiselta mutta onneksi useimpien sukuohjelmienkäyttäjistä ei koskaan tarvitse tutustua tapaan jolla nämä tiedosto muodostetaan -käytössä oleva ohjelma osaa, tai ainakin sen tulisi osata laittaa esimerkiksi tietueet ja niihin linkittyvät yksilöt tietoineen kohdalleen yleisesti ymmärretyllä tavalla.

Tällä hetkellä käytössä olevista ohjelmista läheskään kaikki eivät ole täysin GEDCOM yhteensopivia vaan tiedonsiirroissa saattaa tapahtua tiedon menetystä tai päällekkäisyyksiä. Yksi suuri tavoite tulevia versioita ajatellen olisikin kehittää ohjelmia paremmin yhteensopivaan muotoon: tällä tavalla käyttäjien olisi helpompi esimerkiksi vaihtaa tai päivittää käyttämänsä ohjelma uudempaan versioon -tästä hyötyisivät siis sekä käyttäjät että ohjelmien kehittäjät.

6.2. HTML

Nykyisellä Internetin aikakaudella yhä useammat kotikäyttäjätkin ovat kiinnostuneet kotisivujen ylläpitämisestä, tämän kehityksen myötä myös monet sukututkijat ovat innostuneet julkaisemaan tutkimuksiaan Internetissä. World Wide Web Consortium pyrkii ylläpitämään html tiedostomuotoon liittyviä standardeja joihin voi tutustua esimerkiksi kyseisen järjestön kotisivuilla. Yksi nykypäivän tavoite olisi saada www-sivujen ylläpitämät huomioiman paremmin kyseiset standardi. Sukututkijat eivät useinkaan ole sivuston ylläpitäjiä perinteisessä muodossa vaan suuren osan HTML-koodauksesta suorittaa ohjelma, näiltä osin olisikin toivottavaa että myöskin ohjelman tuottama koodi olisi standardien mukaista.

Yksi HTML:n tuottamiseen pystyvä sukututkimusohjelma on Sukujutut. Kyseisellä ohjelmalla voidaan luoda esimerkiksi hyvinkin pitkä, lukemattomia sisäisiä linkkejä sisältävä esivanhempien taulusto jollaisen tuottaminen käsin koodaamalla olisi kohtuuttoman raskasta. Esimerkiksi Palander [2000, s.24] kuitenkin antaa teoksessaan Sukututkijan tietokoneopas 10 käskyä kotisivun tekemiseen, ohjeet käsittelevät muun muassa sivun sisäistä navigointia, päivittämisen tärkeyttä ja sivujen toimivuutta –onkin tärkeää että sivun ylläpitäjä tutustuisi myös tämän kaltaisiin ohjeistuksiin vaikka itse sivun tuottaminen tapahtuisikin ohjelmallisesti. Olivat tiedot tai taidot millaiset vaan, HTML-ominaisuus sukuohjelmassa on selkeästi vain hyvä asia. Ohjelmalla tuotetuissa sivuissakaan laatuksymyksiä ei kuitenkaan tule unohtaa; Sukujuttujen tuottamisessa www-sivustoissa olisi selkeästi parantamisen varaa, esimerkiksi osa sisäisistä linkeistä ei toimi, kaikki haluttu teksti ei tulostu useimmilla selaimilla ja suositeltua HTML elementtien järjestystä ei ole noudatettu. World Wide Web Consortiumin sivuilla toimiva koodia tarkistava sovellus löytääkin kohtuullisen lyhyestä sukukatulustosta reilusti yli sata kohtuullisen vakavaa virhettä.

Tällä hetkellä HTML-tiedostojen tuottaminen sukuohjelmilla ei siis ole mikään erityisen yleinen tai hyvin toimiva ominaisuus. Tulevaisuudessa verkon käyttö tulee varmasti vain lisääntymään joten voitaisiinkin olettaa että ohjelmien tuottajat keskittyisivät piakkoin erityisesti tämän ominaisuuden kehittämiseen.

7. Yhteenveto

Kokonaisuudessaan suomalaiset sukututkimusohjelmat ovat verrattain laadukkaita, tämän voi huomata viimeistään luvussa viisi esitellyistä käyttäjäkokemuksista. Usein käyttäjä voikin luottaa omaan vaistoonsa ja kokemuksiinsa ohjelmaa valitessaan. Kuten kuitenkin tässä tutkimuksessa on

todettu, puutteita ja selkeitä vikojakin löytyy –oli kyseessä sitten käytettävyys tai ohjelman tuottamat tulosteet.

Tutkimuksen perusteella ohjelmien kehittäjiä voidaankin hyvällä omallatunnolla patistaa panostamaan käytettävyyteen sekä muihin yksityiskohtiin tämän hetkistä enemmän. Toimivuus, selkeys, luotettavuus ja käytön helppous ovat kuitenkin aina ne kaikkein tärkeimmät ominaisuudet, oli kyseessä mikä ohjelma tahansa.

Viiteluettelo

- [Kaila, 2002] Kaarle Kaila 2002, Sukututkimustiedon Käsittely, 2002 -
<http://www.kk-software.fi/kalle/kkgradu.pdf> [26.10.2003]
- [Palander, 2000] Seppo Palander, Sukututkijan Tietokoneopas, Gummerus, Saarijärvi, 2000
- [Palander, 2002] Seppo Palander, Sukututkijan Tietokoneohjelmat, Gummerus, Saarijärvi, 2002
- [Raisamo, 2002] Roope Raisamo (toim.), Käyttöliittymäteoriat ja - mallit, Juvenes print, 2002
- [Sampio, 2003] Seppo Sampio, Sukututkimuksen Avaimet, Gummerus, Saarijärvi, 2003
- [Mäensivu, 2002] Vesa Mäensivu, Ikääntyvien viestintävalmiudet ja digitaalinen epätasa-arvo, Gummerus, Jyväskylä, 2003
- Suomen sukututkimusseuran kotisivu, <http://www.genealogia.fi/> [26.10.2003]
- Brother's Keeper ohjelman kotisivu, http://ourworld.compuserve.com/homepages/Brother's_Keeper/ [26.10.2003]
- Genus Senior ohjelman kotisivu, http://www.bratiето.fi/html/genus_senior.html [26.10.2003]
- Heuristic Evaluation, <http://www.useit.com/papers/heuristic> [26.10.2003]
- Juuret Sukututkimusohjelman kotisivu, <http://www.juuret.fi/> [26.10.2003]
- SukuJutut-sukututkimusohjelman kotisivu, <http://www.sukujutut.fi/>, [26.10.2003]
- Sukuohjelmisto 2000 ohjelman kotisivu, <http://www.kk-software.fi/sukuo/index.htm> [26.10.2003]
- The GEDCOM Standard Release 5.5, <http://www.gendex.com/gedcom55/55gcappa.htm> [26.10.2003]
- World Wide Web Consortiumin kotisivu, <http://www.w3.org/>, [26.10.2003]

Liitteet

Nielsenin heuristiikat

Alla on listattu kymmenen ohjetta hyvään käyttöliittymäsuunnitteluun, osaa listan kohdista olen muokannut juuri tätä tutkimusta silmälläpitäen.

1. Järjestelmän tilan näkyvyys: käyttäjän tulisi kaikissa vaiheissa tietää mitä ohjelma parhaillaan tekee.
2. Yhteys todelliseen maailmaan: Käyttöliittymässä tulisi käyttää mahdollisimman paljon tavallisesta elämästä tuttuja ilmaisuja ja toimintamalleja
3. Käyttäjät tekevät usein virhevalintoja esimerkiksi valikoissa, siksi erilaisista ikkunoista tulisikin olla helppo ja nopea paluu edelliseen tilaan
4. Komentojen ja toimintojen nimien tulee olla yhdenmukaisia ja noudattaa totuttuja tapoja
5. Käyttöliittymää suunniteltaessa tulisi pyrkiä minimoimaan käyttäjän tekemien virheiden mahdollisuus.
6. Ohjelman käyttäminen ei saa edellyttää toimintojen ulkoa opettelua
7. Käyttöliittymän tulee auttaa aloittelevia käyttäjiä ja toisaalta antaa sujuvia etenemismahdollisuuksia kokeneemmille käyttäjille.
8. Ulkoasun suunnittelussa on pyrittävä minimalistisuuteen
9. Virheiden estäminen: virhetilanteen sattuessa käyttäjälle tulisi välittää tieto siitä mikä aiheutti virheen ja miksi
10. Tarvittaessa saatavilla tulisi olla selkeitä ja helppokäyttöisiä opasteita

C++:n ja Prologin käyttö toisiaan täydentäen

Jouni Vaaramo

Tiivistelmä.

Avainsanat ja -sanonnat: deklaratiiivinen, proseduraalinen, C++, Prolog, rajapinta, ohjelmointikielet.

CR-luokat: A.2

1. Johdanto

Tässä tutkimuksessa on tutkittu C++- ja Prolog-ohjelmointikielten käyttämistä toisiaan täydentäen. Tutkimuksen hypoteesina on ollut, että C++:aa ja Prologia voidaan käyttää toisiaan tukien ja että tästä yhdessä käytöstä voi olla jotain hyötyä. Tutkimuksen hypoteesin määrittely näin löyhästi antaa mahdollisuuden päätyä myöntävään tulokseen varsin helposti. Hypoteesi on kuitenkin ohjannut tutkimuksen kulkua ja tutkimuksen tavoitteena on ollut koota tietoa näiden kahden ohjelmointikielen soveltamisesta yhdessä. C++ on valittu käsiteltäväksi suuren suosionsa perusteella ja Prolog on valittu tutkimukseen lähinnä omaperäisen paradigmansa perusteella.

Ohjelmointikielten ominaisuuksia ja soveltuvuutta erilaisten ongelmien ratkaisemiseen on tutkittu kohtuullisen paljon. Samalla erilaisiin käyttöihin sopivia uusia ohjelmointikieliä on kehitetty. Ainakin kokeilu mielessä on suunniteltu ohjelmointikieliä, jotka sisältävät samaan aikaan sekä proseduraalisen että deklaratiiivisen paradigman ominaisuuksia. Tämän kaltaiset moniparadigmaiset kielet eivät kuitenkaan ole yleistyneet, lukuun ottamatta joitain Prologista paranneltuja kieliä ja tulkkeja kuten Prolog++. Etenkään juuri C++:n ja Prologin vahvuuksia hyödyntävää ohjelmointikieltä ei ole kehitetty.

Toisaalta on kehitetty menetelmiä suunnitella moniparadigmaisia järjestelmiä, joissa alijärjestelmät suunniteltaisiin ja ohjelmoitaisiin sopivaa paradigmaa käyttäen. Nimenomaan C++:n ja Prologin käyttämisestä toisiaan täydentäen tälläkään tavalla ei ole tutkimuksia. Rajapintoja näiden kahden kielen välille on suunniteltu ja toteutettu. Tämän tutkimuksen yhtenä tavoitteena on luoda lähtökohtia ja tietoja käytännön suunnitteluun, esittämällä tapoja saada Prolog ja C++ toimimaan yhdessä.

Seuraavassa eli toisessa kappaleessa käsitellään deklarattiivisen ja proseduraalisen ohjelmoinnin ominaispiirteitä. Kolmannessa kappaleessa on katsaus eri tapoihin käyttää C++- ja Prolog-kieliä toisiaan tukien. Tutkimuksessa käytetään termiä *järjestelmä* kuvaamaan jollain ohjelmointikielellä toteutettua kokonaisuutta. Tässä tutkimuksessa puhuttaessa Prologin mahdollisuuksista tarkoitetaan, ellei erikseen mainita, että ominaisuus on toteutettu SWI-Prolog tulkissa.

2. C++:n ja Prologin ohjelmointiparadigmat

Tässä kappaleessa tarkastellaan C++- ja Prolog-ohjelmointikielien ominaisuuksia. Nämä ohjelmointikielien ominaispiirteet toimivat samalla perusteluina tarpeelle käyttää näitä kahta kieltä toisiaan tukien, sen sijaan että käytettäisiin vain toista näistä kielistä.

2.1. Deklaratiivinen ja proseduraalinen paradigma

Ohjelmointikielen tutkimuksessa on usein luokiteltu ohjelmointikielet tiettyä ohjelmointiparadigmaa käyttäviksi. Tässä tutkimuksessa keskitytään tutkimaan nimenomaan ohjelmointikieliä C++ ja Prolog. Käsitteitä *proseduraalinen* ja *deklarattiivinen* käytetään käsittelyn apuna. Yleensä C++ luokitellaan proseduraaliseksi kieleksi ja Prolog luokitellaan deklarattiiviseksi kieleksi. Ohjelmointikielien luokittelu ei kuitenkaan ole yksikäsitteistä ja siinä on vaihtelua paljonkin lähteestä riippuen. Siksi lukijan kannattaa huomata erityisesti, että C++ sisältää monen eri paradigman ominaisuuksia eli se on *hybridikieli*. C++ on myös olio-ohjelmointikieli. Prolog puolestaan on vain eräs monista erilaisista deklarattiivisistä ohjelmointikielistä ja termi logiikkaohjelmointi saattaisi olla täsmällisempi. Tässä käytetään termejä proseduraalinen ja deklarattiivinen lähinnä englannin kielisten tutkimusten termistöä mukaillen.

Oleellista tässä on huomata, että proseduraalisella paradigmalla ohjelmoitujen ohjelmien yhteinen ominaisuus on se, että niiden suorittaminen on aina jossain tietyssä vaiheessa eli *tilassa*. Käytännössä tämä tarkoittaa ohjelman kulkua ohjelmakoodissa rivi riviltä. Ohjelmoitaessa ohjelmoija ohjaa ohjelman kulkua määrittelemällä miten kukin prosessi toimii ja mitä datalle milloinkin tulee tehdä. Vaikka olioparadigma ja proseduraalinen paradigma eroavat toisistaan, niin tässä yhteydessä ne rinnastetaan toisiinsa.

Deklaratiivisessa ohjelmoinnissa ohjelmalla ei ole tiettyä tilaa, vaan ohjelma sisältää datan ja laskennan tuloksen jo itsessään valmiina. Deklaratiivisen ohjelmoinnin ideaalina on, että ohjelmassa määritellään

sääntöinä mitä halutaan ohjelman tekevän ja jätetään ottamatta kantaa siihen miten tietokone käytännössä toimii. Prolog-ohjelma tukee deklarativisuutta melko hyvin; Prologilla ohjelmoitaessa määritellään sääntöjä ja faktoja, joista ohjelma päättelee vastauksen kysymykseen. Prolog-ohjelman perusyksikkö on siis predikaatti, joka määrittelee relaation. [Sterling and Shapiro, 1986]

2.2. C++:n vahvuudet ja heikkoudet

C++ kielen vahvuuksiin kuuluu sillä ohjelmoitujen ohjelmien tehokkuus [Sethi, 1989]. Tehokkuus tässä tarkoittaa ohjelmien suoritusnopeutta ja toisaalta muistin hallinnan taloudellisuutta. *Roskien keruulla* tarkoitetaan yleensä kääntäjän tai tulkin toimesta hoidettua automaattista muistin vapauttamista. C++ kielestä tämä automaattinen roskien keruu on jätetty pois lähinnä historian oikusta. Ohjelmoijan on C++ kielessä kuitenkin yleensä helppoa huolehtia muistin vapauttamisesta ja todellinen roskien keruukin on toteutettavissa.

C++ kielen käytön yleisyyden syynä voitaneen pitää sitä, että C:n tai C++:n käyttö on usein välttämätöntä, koska tarjolla on usein rajapinta vain näille kielille. Tämä C++-kielen osaamisen ja käytön yleisyys on siis muodostunut kielen vahvuudeksi käytännön työssä.

Olio-ohjelmoinnin suunnittelumenetelmät ovat pitkälle kehittyneitä. Olioparadigman keskeisin ominaisuus lienee juuri tiedon kapselointi luokkiin ja olioihin. Tämä mahdollisuus suunnitella ja organisoida suurikin järjestelmä pieninä osina on C++-kielen tarjoava huomattava etu. Tällaisten ohjelman osien ylläpidettävyys ja muunneltavuus on kohtuullisen helppoa.

C++-kielen vahvuutena Prologiin verrattuna on mahdollisuus poikkeusten käsittelyyn sekä mahdollisuus tietotyyppien oikeellisuuden tarkastamiseen jo kääntämisvaiheessa.

2.3. Prologin vahvuudet ja heikkoudet

Prolog-kielen tärkein vahvuus on kielen ilmaisuvoima. Prolog-kielellä ohjelmoituissa ohjelmissa ohjelmakoodin pituus on keskimäärin vain murtoosa proseduraalisiin ohjelmiin verrattuna [Cohen, 1985]. Muilla kielillä ohjelmoitujen ohjelmien pituus on yleensä noin 5-10 kertaa pidempiä kuin vastaavat Prolog-ohjelmat. Logiikkaohjelmoinnissa ohjelman kehittäminen ja suunnittelu ovat usein verrattoman helppoa [Henno, 1991].

Deklaratiivisessa ohjelmoinnissa on muistin käytön hallitseminen ongelmallista. Toisaalta muistin käyttämisen miettiminen Prolog-ohjelmissa ei ole ohjelmoijan ongelma, vaan se on yleensä tulkin ongelma. Joka tapauksessa Prolog-ohjelma saattaa varata suuria muistialueita käyttämättömille tietorakenteille [Mulkers *et.al*, 1994]. Prolog-ohjelmoinnissa ongelmallista on myös samanaikaisuuteen liittyvien ongelmien käsitteleminen [Sterling and

Shapiro, 1986]. Tämä samanaikaisuuden käsittelemisen vaikeus johtuu siitä, että Prolog ohjelmalla ei ole tilaa. Tähän ongelmaan ratkaisuna on kehitetty ohjelmointikieliä, jotka periytyvät Prologista, mutta kykenevät käsittelemään samanaikaisuutta paremmin. Alkuperäisessä Prologissa tämä ongelma kuitenkin pysyy.

Vertailtaessa Prologia ja C++-kieltä voidaan huomata, että Prologin puutteet ovat paljolti samoja, joita edellisessä kappaleessa mainittiin C++:n vahvuuksina. Prolog-kieli ei anna mahdollisuuksia käänösvaiheessa tapahtuvalle tietotyypin oikeellisuuden tarkastamiselle ja tämän kaltaisten ohjelmointivirheiden paljastuminen vasta ajon aikana voi huonontaa ohjelmien toimivuutta.

Prolog-kielen heikkoutena voidaan pitää myös sen tuntemattomuutta ja osaamisen puutetta ohjelmistoteollisuudessa [Milliken *et.al*,1988]. Ohjelmistotuotanto on käännettävien ja proseduraalisten ohjelmointikielten hallitsema. Uusien järjestelmien on kommunikoidava olemassa olevien ohjelmien ja tietokantojen kanssa. Ohjelmoijien on vaikea vaihtaa kokemuksen myötä tutuksi tullutta paradigmaa toiseen. Nämä ongelmat eivät ole varsinaisesti Prolog-kielen vikoja, mutta ovat selvästi Prolog-ohjelmien kehittämistä ja kehittymistä jarruttavia tekijöitä.

Tässä kappaleessa on tullut esille lähinnä Prolog-ohjelmoinnin heikkouksia ja ongelmia. Tämä ei johdu siitä, että Prolog-ohjelmoinnissa olisi vain heikkouksia. Prologin ilmaisuvoima, ja sen kyky antaa ohjelmoijan ratkaista ongelma korkealla tasolla, tekee siitä joidenkin ongelmien kohdalla ei ainoastaan parhaan, vaan käytännössä ainoan mahdollisen vaihtoehdon.

3. C++:n ja Prologin ohjelmointiparadigmojen yhdistäminen

Tässä kappaleessa käsitellään C++- ja Prolog-ohjelmointikielten käyttämistä yhdessä ja toisiaan täydentäen. Ensin on esitetty joitain huomioita siihen, että C++ on käännettävä kieli ja Prolog on yleensä tulkettava kieli. Sitten on tämän tutkimuksen tärkein tekstin osa, jossa esitetään tapoja C++ ja Prolog kielen käyttämiseen yhdessä.

3.1. Kääntämisen ja tulkkauksen ongelma

C++ -kieli on käännettävä eli toisin sanoen C++-ohjelma on käännettävä C++-kääntäjällä konekielelle [Stroustrup, 1986]. Näin saatu konekielinen tiedosto on sitten tietokoneen suoraan ajettavissa. Prolog on (yleensä) tulkettava kieli eli Prolog-tulkki tulkkaa ohjelman konekielelle sitä mukaa, kun ohjelmaa ajetaan. Prolog-ohjelmassa itse ohjelma ja data muodostavat yhden kokonaisuuden [Sterling and Shapiro, 1986]. Datat muuttaminen siis tarkoittaa samalla myös

itse ohjelman muuttamista. Tästä Prologin ominaisuudesta johtuu se, että Prolog ohjelmaa ei voida C++:n tapaan kääntää valmiiksi konekielelle, koska kääntäminen pitäisi tehdä aina, kun dataa on käsitelty.

3.2. Eri tavat C++ ja Prolog -kielten käyttämiseen yhdessä

Mycroft [1996] esittää kirjoituksessaan algebraalisen esitystavan kahden paradigman yhdistämiseen. Samalla hän pohtii tapoja kahden paradigmatilastaan erilaisen alijärjestelmän yhdistämiseen. Mycroft sanoo, että mitä tiiviimmin kaksi eri ohjelmointikieltä (tai kaksi eri ohjelmointikielillä ohjelmoitua alijärjestelmää) on yhdistetty toisiinsa, niin sitä suuremmaksi järjestelmän monimutkaisuus kasvaa.

Seuraavat menetelmät eri ohjelmointikielillä ohjelmoitujen järjestelmien yhdistämiseen sopivat oikeastaan kaikissa tapauksissa, riippumatta kyseessä olevista ohjelmointikielistä. Mahdollisten yhdistämistapojen jako seuraavalla tavalla on kuitenkin mielekäs vain kahdella eri paradigmatilalla ohjelmoitujen järjestelmien yhdistämisessä. Mielekkyyttä johtuu tarpeesta säilyttää kahden ohjelmointitavan ominaiset piirteet ja edut.

3.2.1. Ohjelmointikielten yhdistäminen ja uuden kielen rakentaminen

Kahden ohjelmointikielen yhdistäminen ja uuden moniparadigmaisen kielen toteuttaminen on yksi tapa kahden ohjelmointikielen ominaisuuksien käyttöön toisiaan täydentäen. Yhden esimerkin tällaisesta deklarativisen ja proseduraalisen kielen yhdistämisestä esittää Milliken ryhmineen [Milliken *et.al*, 1988]. Uuden kielen rakentaminen on lähes rajattomia mahdollisuuksia tarjoava ala, mutta toisaalta käytännössä uuden kielen suunnittelu ja tulkin tai kääntäjän rakentaminen on liian kallista ja aikaa vievää. Tässä tutkimuksessa tyydytään toteamaan tällaisen uuden hybridikielen luomisen olevan yksi olemassa oleva vaihtoehto.

3.2.2. Ohjelmointikielten upottaminen toisiinsa

Jan Wielemaker [2002] esittää C++-rajapinnan Prologiin. Wielemaker on rajapintaa esittäessään pitänyt lähtökohtana, että Prolog on 'päävastuussa' ohjelman toiminnasta ja C++:aa on käytetty toiminnallisuuden lisäämiseksi Prologiin. Toimintoja joiden lisäämiseen C++:aa on käytetty ovat esimerkiksi ulkopuolisten resurssien kanssa kommunikointi ja tehokkuuden lisääminen. Rajapinta toimii kuitenkin myös silloin, kun C++ ohjelma on päävastuussa toiminnasta. Wielemakerin rajapinta on toteutettu SWI-Prologissa.

Wielemakerin esittämässä esimerkissä (esimerkki 1) nähdään C++:n ja Prologin käyttöä toisiinsa upotettuina yksinkertaisimmillaan. PREDICATE-makroa käytetään predikaattien määrittelyyn. Esimerkissä määritellään

predikaatti **hello/1** ja muunnetaan Prologin argumentti C:n tietotyyppiä. Makroja A<n> käytetään luomaan yhteys argumentteihin kullakin paikalla (arity). Esimerkissä on lisäksi kokeiltu tätä predikaattia yksinkertaisella kyselyllä.

```
PREDICATE(hello, 1)
{
    cout << "Hello " << (char *)A1 << endl;
    return TRUE;
}
```

?- hello(world).

Hello world

Esimerkki 1. Predikaatin lisääminen ja kokeileminen.

Seuraavat luokat ovat tärkeimmät luokat C++-rajapinnassa ja ne riittävät saamaan kuvan Prologin käytöstä C++-rajapinnan kautta. Lukijan on kuitenkin hyvä tietää, että Wielemaker esittää useita PITerm-luokasta periyettyjä aliluokkia, joilla on hyödyllisiä piirteitä listojen ja merkkijonojen käsittelyyn. Näitä PITerm-luokasta periyettyjä luokkia ei esitellä tässä.

- **PITerm** on Prologin termiä vastaava luokka. Luokka sisältää erilaisia rakentimia. Lisäksi PITerm sisältää operaattoreita tyyppimuunnoksia ja tyyppin tarkastuksia varten. Tällä luokalla on keskeinen asema datan käsittelyssä ja Prologin tiedon käsittelyyn C++:lla.
- **PIQuery** on perusluokka kyselyjen tekemiseen ja kyselyjen tulosten lukemiseen ja läpikäymiseen.
- **PIEngine** luokka toimii Prolog-moottorina ja siis käytetään ohjelmissa, joissa toiminnasta vastaava ohjelma on C++-kielinen. Luokassa on rakennin- ja purkufunktiot Prolog-ympäristön luomiseen ja tuhoamiseen.

Näiden luokkien avulla kyetään siirtämään tietoja C++:n ja Prologin välillä. Tämä rajapinta tarjoaa oivan mahdollisuuden yhdistää eri kielillä ohjelmoituja alijärjestelmiä. Tällaisen mahdollisuuden olemassaolo mahdollistaa myös aina kunkin ongelman ratkaisemiseen soveltuvan ohjelmointikielen.

3.2.3. Yhteisen ulkopuolisen datan käsitteleminen ja ODBC

ODBC (Open Database Connectivity) antaa mahdollisuuden käsitellä relaatiotietokantaa yhteisen rajapinnan kautta [Wielemaker, 2003]. Prolog-kielillä on helppo käsitellä relaatiotietokantaa ODBC:n kautta, mutta tällöin yleensä menetetään Prologin erityisominaisuudet tietokantaohjelmoinnissa. Wielemaker on esittänyt myös Prologin ODBC-rajapinnan, joka on toteutettu SWI-Prologiin. Tällainen yhteisen tietokannan käyttö kahden alijärjestelmän yhdistävänä tekijänä on kuitenkin yksi ajateltavissa oleva menetelmä.

Etuna tällaisessa menetelmässä on, että järjestelmiä ei tarvitse yhdistää kovin tiiviisti. ODBC:n avulla käsitellyssä yhteisessä relaatiotietokannassa yhteistä on vain data ja sitä käsittelevää alijärjestelmää on kohtuullisen helppo muuttaa. Esimerkkinä tällaisen yhdistämistavan käyttämisestä voisi olla tilanne, jossa Prolog-ohjelma toimii päättelykoneena, joka tallentaa C++ ohjelmaa ohjaavia parametrejä ODBC:n läpi relaatiotietokantaan. C++-ohjelma voisi sitten myös ODBC:n avulla lukea ohjausparametrit tietokannasta.

C++:lla ODBC:n ja relaatiotietokannan käyttö yhdessä on usein luonnollinen menetelmä. C++:n kohdalla tässä ei ole samanlaista kielen ominaisuuksien menettämisen ongelmaa kuin Prologin kohdalla, koska C++-ohjelma on selkeästi erillään laskettavasta datasta.

3.3. Yhdistämisen hyödyt

Käsiteltäessä Prologin ja C++:n vahvuuksia ja heikkouksia oli huomattavissa näiden kahden kielen ominaisuuksien vastakkaisuus. Tällöin on ilmeistä, että saataessa molempien kielten vahvat puolet käyttöön, saavutetaan huomattavia etuja. Prologin ja C++:n vahvuuksia käsiteltiin tarkemmin luvuissa 2.2 ja 2.3. Tärkeimpinä ominaisuuksina ovat kuitenkin C++:n laiteläheisyys ja Prologin deklarativisuus.

4. Yhteenveto

Tässä tutkimuksessa on kerätty tietoa C++- ja Prolog-ohjelmointikielistä ja pohdittu tarvetta ja mahdollisuuksia näiden käyttämiseen toisiensa tukemiseen. Lisäksi koottiin tietoa näiden kahden ohjelmointikielen vahvuuksista ja heikkouksista. Aikaisempia tutkimuksia aiheesta ei löytynyt. Toisaalta tutkimuksia ohjelmointikielten ja paradigmojen ominaisuuksista on tehty paljon. Tämän tutkimuksen tärkein tulos suhteessa aiempiin tutkimuksiin

oli koota niistä tähän aiheeseen liittyviä näkemyksiä. Mahdollisina tapoina kielten käyttöön toisiinsa täydentämiseen esitettiin

- ohjelmointikielien yhdistäminen luomalla uusi kieli,
- ohjelmointikielien upottaminen toisiinsa,
- yhteisen datan käsitteleminen (esimerkiksi ODBC:n avulla).

Tässä aihepiirissä mahdollisena jatkotutkimuksena voisi olla näiden mainittujen tapojen koetteleminen käytännössä. Erityisesti tulisi tutkia Prolog- ja C++-ohjelmointikielten upottamista toisiinsa. Tutkimuksen hypoteesina oli aluksi se, että Prolog- ja C++-kieliä voi käyttää toisiaan täydentäen ja että näin saavutetaan jotain etuja. Tämän hypoteesin on tutkimuksen myötä todettu pitävän paikkansa.

Viiteluettelo

- [Cohen, 1985] Jacques Cohen, Describing Prolog by its interpretation and compilation. *Communications of the ACM*, **28** (12), 1985.
- [Henno, 1992] Jaak Henno, *Prolog ja Olympoksen Jumalat*. WSOY, 1992.
- [Milliken *et.al*, 1988] K.R. Milliken, A.J. Finkel, D.A. Klein, N.B. Waite, Adding Rule-based Techniques to Procedural Languages. *Proceedings of the First International Conference on Industrial and engineering applications of artificial intelligence and expert systems - Volume 1*, ACM Press, 1988, 185-194.
- [Mulkers *et.al*, 1994] Anne Mulkers, William Winsborough, Maurice Bruynooghe, Live-Structure Dataflow Analysis for Prolog. *ACM Transactions on Programming Languages and Systems*, **16** (2), 1994.
- [Mycroft, 1996] Alan Mycroft, On integration of Programming Paradigms. *ACM Computing Surveys*, **28** (2), 1996.
- [Sethi, 1990] Ravi Sethi, *Programming Languages: Concepts and Constructs*. Addison-Wesley, 1990.
- [Sterling and Shapiro, 1986] Leon Sterling and Ehud Shapiro, *The Art of Prolog: Advanced Programming Techniques*. The MIT Press, 1986.
- [Stroustrup, 1986] Bjarne Stroustrup, *the C++ Programming Language*. Addison-Wesley, 1986.
- [Wielemaker, 2002] Jan Wielemaker, A C++ interface to Prolog. October 2002. Available as <http://www.swi-prolog.org/packages/pl2cpp.html> (15.12.2003)
- [Wielemaker, 2003] Jan Wielemaker, SWI-Prolog ODBC Interface. June 2003. Available as <http://www.swi-prolog.org/packages/odbc.html> (15.12.2003)

Ratkaisuja roskienkeruun ongelmaan

Zoltán Varga

Tiivistelmä.

Tutkielmassa esitellään roskan käsite tietojenkäsittelytieteessä, roskienkeruun perusmenetelmät ja niiden optimoidut muunnelmat. Lisäksi katsastetaan ohjelmoijan käytettävissä olevat roskienkeruutietoiset välineet eräissä ohjelmointikielissä ja -ympäristöissä (Java, .Net, C++).

Avainsanat ja -sanonnat: roskienkeruu, algoritmit, ohjelmointikieliset CR-luokat: D.4.2, D.3.3

1. Johdanto

Roskienkeruun tarkoitus on löytää ja poistaa tarpeettomat alkiot tutkittavasta joukosta. Sen tunnetuin sovellus on ohjelmien automaattisen muistinhallinnan toteutus, joka on tämän katsauksen teemana. Ensimmäiset ohjelmointikielen ajoympäristöön sulautetut alijärjestelmät, joiden tehtävänä oli päätellä mitkä ohjelman varaamat muistisolut olivat tarpeellisia ja mitkä ei, kehitettiin jo 60-luvulla. Alkuperäiset toteutukset olivat varsin tehottomia, mutta automaattisen muistinhallinnan edut manuaaliseen nähden kannustivat menetelmien kehittämiseen. Nykyään roskienkeruu on valmiiksi integroitu ohjelmistotuotannon valtavirran kielisiin ja alustoihin (Java, .Net), tai ainakin se on sovellettavissa niihin ajoaikaisen kirjaston tai muun vastaavan mekanismin avulla (C/C++).

1.1. Roskienkeruun tarve ja merkitys

Vanhanmalliset imperatiiviset kielet jättävät ohjelmoijan vastuulle dynaamisesti varatun muistin vapauttamisen. Näissä kielissä puuttuva muistin vapautusoperaatio aiheuttaa muistivuodon, kun taas muistialueen ennenaikainen vapautus synnyttää roikkuvia osoittimia, jotka voivat aiheuttaa indeterminististä ohjelman käyttäytymistä. Manuaalinen muistinhallinta lienee ohjelmavirheiden merkittävin lähde.

Modulaarisessa ohjelmoinnissa eksplisiittinen muistinhallinta voi synnyttää ohjelmalogiikasta riippumattomia riippuvuussuhteita moduulien välissä – varatun muistin hallinta ja saatavuustietojen ylläpito voi muuttua ohjelmassa paikallisesta ongelmasta globaaliseksi, mikä vähentää uudelleenikäytön mahdollisuuksia sekä heikentää moduulien ylläpidettävyyttä ja laajennettavuutta. Ohjelmoija saattaa törmätä myös sellaiseen ongelmaan, että ohjelman rakenteesta johtuen ei löydy varsinaista hyvää paikkaa tietyn muistialueen

vapauttamiseen. Eksplisiitinen muistinhallinta ilmenee siis ongelmana monella ohjelmistokehityksen abstraktiotasolla.

Ohjelmia tuotetaan hyvin monenlaisiin tarkoituksiin, siksi nille asetetut suoritusvaatimukset vaihtelevat suuresti, samoin niiden muistinkäytön laatu. Koska roskienkeruusta voi muodostua merkittävä pullonkaula ohjelman suorituksessa, on syytä ottaa huomioon eri ohjelmointikielten ja ajoympäristöjen tarjoamat mahdollisuudet virittää ja tehostaa roskienkeruu. Esimerkiksi Java-alustalla voidaan valita sopiva algorimi monesta vaihtoehdosta, mutta kaikissa muissakin ympäristöissä on runsaasti optimoitavia tekijöitä. Roskienkeruumenetelmien toimintamekanismin tarkka tuntemus on tarpeen tehtäessä roskienkeruun hienosäätöä. Menetelmien vertailu taas edesauttaa kunkin menetelmän hyötyjen ja haittapuolten ymmärtämistä, mikä on välttämätöntä oikean menetelmän valinnassa.

1.2. Lukujen sisältö

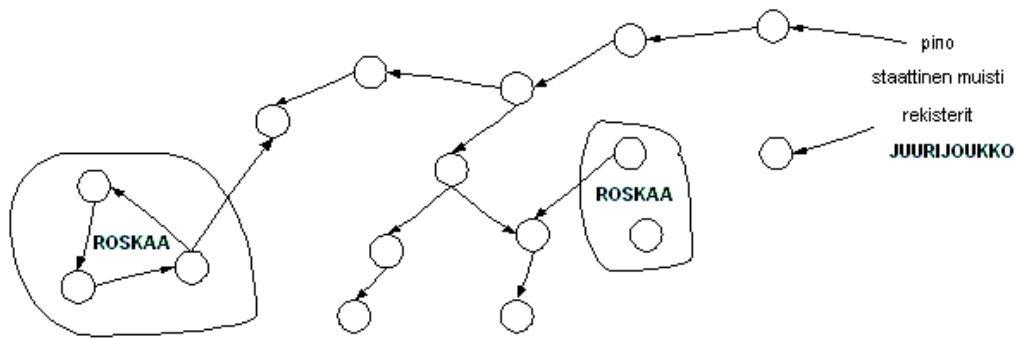
Seuraavissa luvuissa tutustutaan ensin roskienkeruun aihepiiriin yleisesti (luku 2), sen jälkeen käydään läpi kaikki klassiset roskienkeruumenetelmät (luku 3), jotka ovat kehittyneempien menetelmien, vähittäisten algoritmien (luku 4) ja ikäpohjaisen roskienkeruun (luku 5) perustana. Lopuksi tarkastellaan roskienkeruun ohjelmointitekniisiä näkökulmia eri ohjelmointikielten (Java, .Net, C++) kannalta (luku 6).

2. Roskienkeruun keskeiset käsitteet ja menetelmät

Ennen kuin tutustutaan algoritmien kirjoon, on syytä esitellä muutama roskienkeruun aihepiiriin kuuluva käsite. Samalla pohditaan roskienkeruun optimoitavuutta ja niitä tarpeita, jotka ovat johtaneet niin monenlaisen algoritmin ja muunnelman kehittelyyn.

2.1. Roskan käsite eri menetelmien kannalta

Roskienkeruu perustuu seuraavaan periaatteeseen: jos johonkin muistialkioon ei pääse ohjelmasta osoittamaan mitään viittauspolkua pitkin, alkio voidaan turvallisesti merkitä roskaksi (*garbage*). Kuten kuva 1 havainnollistaa, roskienkerääjä (*collector*) pitää tarpeellisina ohjelman ulottuvissa olevia oliota (*reachable data*) ja tarpeettomina (roskina) kaikkia irrallisia oliota (*unreachable data*). Kuva 1 mukailee Abdullahia ja kumppaneita [1992]. Alkiolla tarkoitetaan yleisesti ohjelman dynaamisesti varaamaa muistialuetta, esimerkiksi oliopohjaisten kielten oliota. Olio käsitteenä on tässä tapauksessa alkiota kuvaavampi, siksi tutkielmassa viitataan kaikkiin muistialkioihin olioina.



Kuva 1. Graafin solmut, joihin ei pääse juurijoukosta, ovat roskia.

Roskienkerääjän kaksi tehtävää ovat roskien löytäminen (*discovery*) ja siivoaminen (lakaisu), eli roskaksi todettujen olioiden varaaman muistin vapauttaminen (*reclamation*). Algoritmista riippuen tehtävät voidaan suorittaa peräkkäin tai lomittain, tai siivoaminen voidaan sivuuttaa väliaikaisesti otollisempaa hetkeä odottaen (*deferred reclamation*). Siivoaminen voi tapahtua myös implisiit- tisesti ilman erillisiä toimenpiteitä, kuten tullaan algoritmien esittelyn yhtey- dessä näkemään.

Roskien löytämiseen on tarjolla kaksi perustekniikkaa. Toinen niistä on viittauslaskureiden käyttö (*reference counting*) ja toinen olioverkon läpikäynti, jonka kaksi perusalgoritmia ovat kaksivaiheinen "merkitse ja lakaise" (*mark- and-sweep*) ja "pysäytä ja kopioi" (*stop-and-copy*).

Viittauslaskenta-algoritmi toimii laskennan lomassa. Jokaiselle oliolle ylläpidetään laskuria, jota kasvatetaan yhdellä aina kun olioon luodaan uusi viittaus ja vastaavasti vähennetään yhdellä kun viittaus poistetaan. Viittaus- laskuritekniikan roskamäärittely on yksinkertainen: olio on roskaa, jos sen vii- telaskurin arvo on nolla.

Viittauslaskentatekniikka ja graafin läpikäynti edustavat kahta eri roskienkeruupolitiikkaa. Viittauslakennan etu graafin läpikäyntiin nähden on sen vähittäisyys. Hyötylaskenta ja roskienkeruu lomittuvat synkronisesti. Jälkimmäinen tekniikka perusmuodossaan taas toimii niin sanotulla stop-and- go -periaatteella, eli hyötylaskenta pysähtyy, kunnes roskienkeruu on suo- ritettu alusta loppuun. Tekniikasta on kuitenkin olemassa myös vähittäisiä (*incremental*) ja rinnakkaisia (*concurrent*) muunnelmia. Rinnakkainen roskien- kerääjä toimii ohjelman hyötylaskentaa tekevän osan, eli muuttajan (*mutator* [Dijkstra *et al.*, 1978]) kanssa asynkronisesti.

Olioverkon läpikäyntiä käyttävä tekniikka perustuu graafiteorioihin. Ohjelman käyttämä muisti voidaan nähdä suunnattuna verkkona, jonka sol- muina ovat ohjelman luomat oliot, ja kaarina niitä yhdistävät osoittimet. Verkko käydään läpi lähtien viittausjuurijoukosta (*root set*), jonka muodostavat

staattisella muistialueella, kontrollipinossa ja rekistereissä olevat viittausmuuttujat, olioiden jäsenmuuttujat muukaanlukien. Tarpeellisuutta laajasti määritellen voidaan sanoa, että käytyään läpi kaikki mahdolliset viittauspolut, roskienkerääjä tuntee tarpeellisten olioiden joukon. Oliot, joita läpikäynti ei koskettanut, ovat roskia. Määrittely on siksi laaja, että vaikka ohjelmalla olisikin periaatteessa mahdollisuus käyttää tiettyä oliota, sellaista suorituspolkua, jossa oliota käytettäisiin, ei välttämättä enää ole. Edellä mainittu joukko saattaa siis sisältää myös kuollutta dataa (*dead data*), jota sanotaan tässä yhteydessä semanttiseksi roskaksi (*semantic garbage*). Roskienkeruussa datan elollisuuden (*liveness*) kriteeri on varsin konservatiivinen ja staattinen esimerkiksi optimoivan kääntäjän kriteereihin nähden.

Perinteisessä stop-and-go mallissa järjestelmän päätettäväksi jää roskienkeruun tahdistus. Yleensä toimenpide suoritetaan vasta silloin, kun muisti loppuu, eli muistinvarauspyyntö epäonnistuu (*fixed-space garbage collection*). Vaihtoehtona on prosentteina ilmoitettu muistinkäyttökynnys (*allocation-threshold*), jota ohjelma ei saa ylittää. Moni algoritmi toimii tehokkaasti vain, jos sillä on tarpeeksi vapaata työmuistia käytettävissään. Varhaisten kerääjien tärkein säädettävä ominaisuus olikin yllämainittu allokointikynnys, joka oli samalla merkittävin tapa optimoida roskienkeruuta. Kehittyneemmät algoritmit ovat paljon laajemmin optimoitavissa.

2.2. Menetelmien optimointi ja tehokkuusmittarit

Algoritmin optimointi tehdään aina jonkin ominaisuuden suhteen, toisen ominaisuuden kustannuksella. Yleisesti ottaen optimoitava tekijä on joko ohjelman läpäisykyky (*throughput*) tai sen muistinkäyttö (*footprint*) – roskienkeruussa keskeisiksi tekijöiksi nousevat myös katkojen pituus hyötylaskennassa (*pause-time*) ja muistin kierretyksen tehokkuus (*promptness*). Yleensä optimointi tehdään läpäisykyvyn suhteen, mutta esimerkiksi mobiileissa laitteissa optimoinnin kohteena on tyypillisesti muistinkäyttö. Ohjelman vasteaikoihin vaikuttavat roskienkeruun keskeytettävyyden tai rinnakkaisuuden. Keskeytettävyyden parantaminen vaatii uhrauksia sekä kellosykliä, että muistinkäytön osalta.

Optimoinnista aiheutuvat muistikustannukset voivat olla kolmenlaiset: kasvaneen (fyysisen) muistin tarpeen lisäksi vähemmän ilmeisiä ovat muistin pirstoutuminen pieniksi lohkoiksi (*fragmentation*) ja niin sanottu paikallisuuden ongelma (*locality*), joka tarkoittaa toisiinsa läheisesti liittyvien olioiden hajaantumista fyysisessä muistissa.

Muistin pirstoutuminen voi vaikeuttaa suuresti muistinvaraajan työtä, sillä varatessaan muistia se joutuu läpikäymään suuria tietorakenteita, jotka sisältävät tietoa vapaista muistilohkoista ja niiden koosta. Pirstoutuneessa muistissa

saattaa olla paljonkin vapaata tilaa jäljellä, silti muistinvarausoperaatiot voivat epäonnistua, koska sopivan kokoista lohkoa ei enää löydy. On olemassa muistinvarausalgoritmeja, jotka itsessään vähentävät pirstoutumisongelmaa, mutta ne vaativat myös veronsa joko kellosykleinä tai muistitavuina. Muistinvarausmekanismin tulee myötäillä roskienkeruun linjaa läpäisykyvyn ja muistinkäytön suhteen. Tosin muistin pirstoutuminen ei välttämättä ole jokaisen ohjelman ongelma. Esimerkiksi Johnstone [1997] osoitti, että on olemassa suuri joukko C ja C++ ohjelmia, joissa muistin pirstoutumista ei esiinny, tai siitä ei ole haittaa.

Paljon muistia käyttävissä sovelluksissa välimuistiosumat ja vähäinen sivustatarve ovat tehokkuuden avaintekijöitä. Kun tekniikkana ei ole viittauslaskenta, roskienkeruu koskettaa ohjelman muistia kokonaisuudessaan, sillä jokaisen muistisolun tarpeellisuus tutkitaan. Tällöin on tärkeitä, että toisiinsa liittyvät oliot jakautuvat ryhmänä mahdollisimman vähälle muistisivulle, muuten joudutaan jatkuvasti hakemaan uusia sivuja välimuistiin ja häätämään toisia takaisin keskusmuistiin.

Kirjallisuudessa mainitaan usein havainto olioiden lyhytikäisyydestä, eli olioiden suuresta lapsikuolleisuudesta (*infant mortality*). Yleisesti arvioidaan, että keskiverto-ohjelmassa noin 95% uusista olioista ei selviä ensimmäisestä roskienkeruusta. Havainnolla on merkitystä vähittäisten menetelmien tehokkuuden kannalta. Näiden algoritmien tehokkuutta voidaan nimittäin ilmaista yksinkertaisesti sillä, kuinka paljon muistissa on jäljellä roskia roskienkeruun päätyttyä, eli kuinka hyvin algoritmi selviää roskienkeruun aikana syntyneestä, niin sanotusta kelluvasta roskasta (*floating garbage*). Ikäpohjainen roskienkeruu (*generational garbage collection*), josta tarkemmin luvussa 5, on saanut alkunsa tästä lyhytikäisyshavainnosta.

3. Klassiset algoritmit

Roskienkeruun aihepiiristä on kirjoitettu monta katsausartikkelia kuluneiden vuosikymmenien aikana. Tunnetuimmat lienevät peräisin Cohenilta [1981] ja Wilsonilta [1992]. Tämän luvun klassisten algoritmien esittely pohjautuu lähinnä näiden artikkeleiden sisältöön.

3.1. Viittauslaskenta

Viittauslaskureita käyttävä tekniikka voi täyttää jopa reaaliaikaisuuden vaatimuksia, sillä se toimii hyötylaskennan lomassa, eikä yleensä kaappaa itselleen laskentaa pidentäen vasteaikoja. Tämä ei kuitenkaan tarkoita, että se ei vaatisi paljon järjestelmän voimavaroja. Jokainen viittausmuuttujaan tehty muutos aiheuttaa kahden ylimääräisen aritmeettisen operaation suorittamisen (vähän-

nys, lisäys). Samalla kun jonkin olion viitelaskuri tulee nolllaksi, täytyy vähentää yhdellä jokaisen olion laskuri, johon kyseinen olio viittaa. Jos olio on suuren puurakenteen juuri, hyötylaskenta voi keskeytyä merkittäväksi ajaksi. Ratkaisuna tähän on erillisen listan ylläpito suurista tietorakenteista, jotka ovat vapautettavissa. Lista tyhjennetään kun se ei haittaa hyötylaskentaa, tai viimeistään silloin kun muisti loppuu.

Viitelaskurin varaama tila voi olla myös varteen otettava tekijä. Vaikka bitti tai tavu riittäisi useimpiin käytännön tilanteisiin, joskus joudutaan varaamaan suurempi kokonaisuus arkkitehtuurisista syistä. Jos ohjelma käyttää todella paljon pienikokoisia dynaamisesti varattuja olioita (esim. kokonaislukuja), viitelaskureiden osuus ohjelman käyttämästä muistista voi lähentyä 50 prosenttia. Jos viitelaskurille varataan yksi ainoa tavu, ylivuodon vaara kasvaa. Ylivuodon sattuessa laskuria ei voi enää käyttää, vaan olio jää keräämättä.

Viittauslaskentatekniikkaa ei juurikaan käytetä niissä ohjelmointikielissä, jotka sallivat syklisiä rakenteita. Nimittäin algoritmin roskakriteeristä "jos olioon ei viitata, se on roskaa" ei seuraa, että "jos olioon viitataan, se ei ole roskaa". Niin kauan kuin kaksi oliota viittaavat toisiinsa, kummankaan viitelaskuri ei tule nolllaksi.

Tekniikkaa käytettäessä joudutaan ottamaan jokin sopiva olioverkkoa läpikäyvä menetelmä avuksi, jos halutaan

- eheyttää muistia, tai
- kerätä ne oliot, joiden viitelaskuri on vuotanut yli, tai
- poistaa sykliset rakenteet muistista.

Viittauslaskureita käyttävää tekniikkaa suositetaan puhtaasti funktionaalisissa ohjelmointikielissä, sillä ne eivät salli syklisiä rakenteita, ja voivat hyödyntää viitelaskuria "vain yhdestä paikasta viitattujen" (*unique reference*) ominaisuuksien tarkistamiseen.

3.2. Merkitse ja lakaise (mark-and-sweep)

Tekniikkana mark-and-sweep on yksinkertainen ja sen periaate on täysin virheetön. Verkon läpikäynti voidaan suorittaa leveyssuunnassa (horisontaalisesti), edeten olioiden etäisyyden mukaan juurijoukosta, tai pystysuunnassa (vertikaalisesti), edeten graafissa aina niin syväälle kuin mahdollista. Läpikäydyt oliot merkitään joko bittikarttaan, tai merkintä tehdään olion yhteyteen varattuun tavuun (*tag*). Bittikartan käyttö on huomattavasti välimuistiystävällisempi tapa, sillä se ei muuta välimuistin sisältöä, eikä siten aiheuta varsinaisen muistin turhaa päivitystä. Lopuksi lakaistaan muisti keräten kaikki merkitsemättömät oliot listaan, jota käytetään varattaessa muistia (*free-list*). Listan fyysisesti vierekkäisiä elementtejä voidaan yhdistää (*coalesce*).

Algoritmia voidaan tehostaa jättämällä lakaisuvaihe pois, ja suorittaa se vähittäisesti muistin varaamisen yhteydessä (*lazy sweeping* tai *mark-and-don't-sweep*). Toinen tapa (nimeltään *treadmill collection*) on ylläpitää linkitettyä listaa tarpeellisista olioista (aktiivinen lista) ja vapaista muistipaikoista (passiivinen lista). Kun vapaita muistipaikkoja ei ole enää saatavissa tai vapaa muisti on pirstoutunut liian pieniksi paloiksi, tehdään roskienkeruu. Jäljellä olevat vapaat muistipaikat liitetään aktiiviseen listaan, näin passiivinen lista tyhjenee. Samalla kun olioverkkoa käydään läpi, tarpeelliset oliot siirretään passiiviseen listaan. Roskienkeruun päätyttyä aktiivinen lista sisältää roskat ja varaamattomat muistialueet. Vaihtamalla listojen roolit (*flip*), päästään alkutilanteeseen. Tekniikka ei ota kantaa muistin pirstoutumiseen ja sen muistintarve on listojen toteutuksesta riippumatta suuri.

3.3. Tiivistävä merkitse ja lakaise (mark-compact)

Tiivistävä (*compressing*) muunnelma edellisestä algoritmista on kehitelty muistin pirstoutumisen välttämiseksi ja vähentämään paikallisuusongelmaa. Algoritmi kerää roskien sijasta tarpeelliset oliot. Merkitseminen suoritetaan tavalliseen tapaan, mutta siivousvaiheessa algoritmi siirtää merkityt oliot vierekkäin muistin alkupäähän. Tällä tavoin saavutetaan toisiinsa liittyvien olioiden läheinen fyysinen sijainti muistissa. Algoritmin todellinen teho piilee siihen liittyvässä muistinvarausmallissa: dynaamista muistia voidaan käyttää pinomaisesti "keko-osoitimen" (*heap pointer*) avulla, joka jakaa muistin käytettyyn ja vapaaseen osaan. Uusille olioille ei tarvitse etsiä sopivaa vapaata lohkoa, vaan ne luodaan peräkkäin käytetyn alueen jatkeeksi siirtäen osoitinta aina eteenpäin. Tiivistettäessä olioiden uuden sijainnin laskeminen, niiden varsinainen siirtely ja viittausmuuttujien päivitys ovat kuitenkin varsin kalliita operaatioita.

3.4. Tiivistäen kopioiva keruu (stop-and-copy)

Tiivistävän algoritmin ideaa edelleen kehittäen tiivistäen kopioiva (*copying-compacting*) algoritmi vähentää edellisen laskentakustannuksia suorittamalla tiivistämisen samalla kun käy olioverkkoa läpi. Ohjelman suorituksen aikana muisti on jaettu kahteen samankokoiseen osaan - puoliavaruuteen (*semispace*) - joista vain toinen on ohjelman käytössä. Kun tämä alue (lähdeavaruus) varataan täyteen, alkaa roskienkeruu, jossa tarpeelliset oliot kopioidaan tiivistäen toiseen muistipuoliskoon (kohdeavaruuteen). Roskienkeruun päätyttyä puoliskoiden roolit vaihtuvat (*flip*). Varsinaista lakaisua ei tarvita.

Cheney [1970] esitti tiivistäen kopioivan algoritmin, joka käyttää leveys-suuntaista läpikäyntiä. Kopiointi etenee aaltolina: kopioituaan ensin juurijoukosta saavutettavat oliot, algoritmi käy kopioiden viittausjäsenmuuttujat

läpi, mistä syntyy uusi kopiointialto. Heti kun viittausmuuttujan osoittama olio on kopioitu, muuttuja päivitetään osoittamaan olion uuteen paikkaan. Syklisten rakenteiden takia algoritmi käyttää viittausten uudelleenohjaustekniikkaa (*memory forwarding*). Jokaisen kopioidun olion tilalle jätetään vanhaan muistipuoliskoon osoitin olion uuteen paikkaan (*forwarding reference*). Mikäli vastakopioidusta oliosta viitataan jo aikaisemmin kopioituun olioon, viittausmuuttujan arvo päivitetään vastaamaan tätä osoitinta.

Tiivistäen kopioiva algoritmi ehkäisee olioiden hajaantumista ja säilyttää muistin eheyden huomattavasti pienemmin kustannuksin kuin tiivistävä algoritmi. On kuitenkin selvää, että tiivistäen kopioivaa algoritmia käyttävää roskienkeruuta joudutaan suorittamaan useammin, sillä muistia on käytettävissä puolet vähemmän.

3.5. Adaptiivinen roskienkeruu

Käytännön toteutuksissa harvoin toimitaan vain yhden algoritmin varassa. Ajoaikainen järjestelmä voi valita uuden algoritmin spekulatiivisesti heurististen sääntöjen avulla. Esimerkiksi joissakin ohjelmissa kopioiva algoritmi voi olla tehokas ohjelman alustusvaiheessa, jossa syntyy paljon uusia olioita, mutta ei enää myöhemmin, kun ohjelman muistinkäyttö on vakiintunut ja vanhoja olioita kopioidaan jatkuvasti paikasta toiseen. Tällöin on tarkoituksenmukaisempaa vaihtaa merkitse ja lakaise -tyyppiseen algoritmiin.

Eckelin [2000] mukaan JDK:ssa 1.1 virtuaalikone toimii tällä tavalla. Tämäntapaisia hybridiratkaisuja tutkitaan nykyisinkin aktiivisesti. Esimerkiksi Bacon *et al.* [2003] esittivät C++ kieleen tarkoitettun toteutuksen, joka käyttää tavallisesti ei-kopioivaa algoritmia, mutta vaihtaa automaattisesti kopioivaan muistin pirstoutuessa.

4. Vähittäiset algoritmit

Vähittäisen roskienkeruun ideana on ehkäistä pitkät tauot hyötylaskennassa tekemällä työtä lyhyitä aikoja kerrallaan. Tarkoituksena on ohjelman vasteaikojen parantaminen, haasteena taas keruun riittävän nopean edistymisen varmistaminen.

4.1. Vähittäin etenevä roskienkeruu

Ajettaessa reaaliaikaisuuden piirteitä omaavia sovelluksia, roskienkeruun rinnakkaisuus, vähittäisyys tai keskeytettävyyys nousee kynnyskysymykseksi. Hyötylaskennan lomassa toimivalla viittauslaskentatekniikalla on reaaliaikaisuuden vaatimat ominaisuudet, mutta muiden negatiivisten ominaisuuksiensa takia tekniikkaa sovelletaan harvoin. Jäljelle jäävät siis mark-and-sweep ja stop-and-copy -tekniikoiden optimoidut muunnelmat.

Perusmuodossaan edellä mainitut tekniikat ovat stop-and-go -tyyppisiä, joten ne vaativat muuttimen toiminnan jäädyttämistä roskienkeruun ajaksi. Algoritmien vähittäisissä muunnelmassa ongelmat ilmenevät merkintävaiheessa, sillä muutin saattaa muuttaa graafin kesken läpikäynnin. Roskienkerääjältä kuitenkin vaaditaan, että – huolimatta voimakkaasta häirinnästä – toimii turvallisesti (virheettä), haittaamatta muuttimen toimintaa. Tilanne on erityisen hankala tiivistävien ja kopioivien algoritmien osalta. Asetelmaa voidaan verrata monen lukijan ja monen kirjoittajan ongelmaan, sillä sekä muunnin, että roskienkerääjä voivat muuttaa graafia. Tämäntapainen toiminta vaatii raskasta koordinoitua.

Dijkstra *et al.* [1978] kolmivärimerkintä (*tricolor-marking*) on omiaan haainnollistamaan ongelmaa. Kolmella värillä – musta, harmaa, valkoinen – merkitystä graafista on nähtävissä kunakin hetkenä läpikäynnin tila. Läpikäynnin alussa graafin kaikki solmut ovat valkoisia. Edetessään graafissa roskienkerääjä värjää koskettamansa solmut harmaiksi. Joutuessaan umpisolmuun (solmuun, josta ei pääse enää eteenpäin), se värjää solmun mustaksi. Mustaksi muuttuvat myös ne solmut, joista pääsee vain mustiin solmuihin. Läpikäynnin päättyessä kaikki alkioit ovat joko mustia (tarpeellisia), tai valkoisia (roskia). Tässä muodossaan algoritmi ei juuri eroa edellä esitetyistä, joiden voidaan oikeastaan katsoa käyttävän kaksivärimerkintää. Harmaa väri on kuitenkin tarpeen, sillä se ilmaisee, että solmun edustama olio on tarpeellinen, mutta (kaikista) sen jälkeläisten tarpeellisuudesta ei ole vielä tietoa.

Keskeytyksen jälkeen roskienkerääjä jatkaa työtään harmaista solmuista eteenpäin, eikä palaa enää mustiin solmuihin, vaikka muuttaja olisikin tehnyt niistä tarpeettomia ennen merkintävaiheen päättymistä. Tämä ei kuitenkaan riko virheettömyysvaatimusta, sillä tällaiset oliot tulevat kerätyksi seuraavan roskienkeruun yhteydessä. Asialla on korkeintaan tehokkuusmielessä merkitystä.

Tilanne on toinen, jos muuttaja hävittää kaikki viittaukset valkoiseen solmuun, mutta samalla luo uuden viittauksen siihen mustasta solmusta. Tällöin valkoinen solmu tulisi kerätyksi tarpeellisuudestaan huolimatta. Kuten edellisestä esimerkistä on nähtävissä, virheettömyysvaatimus edellyttää, että mustasta solmusta ei voi päästä valkoiseen solmuun suoraan. Kyseessä on niin sanottu invarianttisääntö: musta solmu on invariantti, jos harmaiden solmujen vyöhyke erottaa sen valkoisista solmuista. Invarianttisäännön voimassapitoon on tarjolla kaksi mekanismia, lukumuuri (*read-barrier*) ja kirjoitusmuuri (*write-barrier*).

Lukumuurin käyttö tarkoittaa käytännössä, että muuttajan lukiessa valkoisen solmun osoitteen solmu värjätään harmaaksi. Näin ollen muuttaja ei voi

luoda viittauksia valkoisiin solmuihin. Kirjoitusmuuria voidaan käyttää kahdella tavalla, tarkkailemalla joko mustiin tai muunvärisiin solmuihin tehtyjä muutoksia. Molemmissa tapauksissa ohjelma suorittaa ylimääräisiä käskyjä kirjoitusoperaation yhteydessä, joiden tarkoitus on pitää roskienkerääjän tiedot ajan tasalla. Nämä käskyt voidaan lisätä ohjelmaan jo käännoaikana. Kirjoitusmuureista koituu yleensä vähemmän kustannuksia kuin lukumuureista, sillä kirjoitusoperaatioita suoritetaan vähemmän.

4.2. Kirjoitusmuurit

Klassinen tapa hyödyntää kirjoitusmuuria on roskienkeruun aikana muuttuneiden viittausten vanhojen arvojen tallentaminen (*snapshot-at-beginning* [Wilson, 1992]²⁰). Läpikäynnin päätyttyä roskienkerääjä aloittaa uuden läpikäynnin käyttäen tallennettuja osoittimia juurijoukkona. Tämä varmistaa sen, että mikään olio ei häviä roskienkerääjän näkyvistä, ja sen tiedot pysyvät näin ajan tasalla. Asia voidaan toki nähdä myös konkreettisenä haittana: algoritmin tehoaste jää alhaiseksi, sillä läpikäynnin aikana syntyneitä roskia ei lakaista.

Kelluvan roskan kannalta tehokkaampi tapa on tarkistaa onko viittausmuuttujan päivitys tehty mustaan solmuun. Jos näin on asianlaita, joko musta, tai viitattu solmu väritetään harmaaksi. Tällä kirjoitusmuurin käytötavalla (*incremental update*) on omana erikoisuutenaan se, että uudet oliot voidaan turvallisesti merkitä valkoisiksi, sillä vaikka uusi olio liitettäisiinkin mustaan solmuun, jompikumpi solmuista muuttuu harmaaksi [Dijkstra *et al.*, 1978]. Muussa tapauksessa solmu jää valkoiseksi, kunnes läpikäynti (mahdollisesti) koskettaa sitä. Asialla on merkitystä olioiden lyhytikäisyshavaintoa ajatellen, sillä keruun ajan syntyneet roskat voidaan siivota.

Kirjoitusmuuria voidaan käyttää varmistamaan myös kopioivien algoritmien virheettömyys. Muurin tehtävänä on toistaa lähdeavaruuden olioihin tehdyt muutokset kohdeavaruudessa. Koska kirjoitusoperaatiot ovat kalliita, tätä menetelmää käytetään vain erikoistapauksissa.

4.3. Lukumuurit

Cheneyn [1970] tiivistäen kopioiva algoritmi on helposti muutettavissa vähittäiseksi lukumuurin avulla Bakerin [1978] esittämällä tavalla: Lukumuurin tehtävänä on varmistaa, että muuttaja työskentelee aina olion ”oikean” version kanssa. Jos olio on jo kopioitu toiseen muistipuoliskoon, lukumuuri kääntää kaikki viittaukset sinne. Muussa tapauksessa kopiointi tehdään saman tien. Tämä vastaa solmun värjäämistä harmaaksi. Uudet oliot luodaan valmiiksi kopiointin kohdeavaruuteen (mustiksi solmuiksi), mikä johtaa siihen, että ne

²⁰ Nimitys ”snapshot-at-beginning” on peräisin Wilsonilta.

selviävät roskienkeruusta huolimatta mahdollisesta tarpeettomuudestaan roskienkeruun päätyttyä. Tosin roskienkeruun aikana roskaksi muuttuneet vanhat oliot siivotaan.

Lukumuuria voidaan käyttää myös mark-and-sweep -tekniikan kanssa, erityisesti linkitettyjä listoja käyttävän muunnoksen (treadmill) yhteydessä. Syklisesti yhteen linkitetty lista sisältää järjestyksessä neljä joukkoa: passiivinen lista (vanhat oliot), aktiivinen lista (tyhjä), uudet oliot (tyhjä), vapaat muistialueet. Joukkojen erottamiseksi on käytössä neljä osoitinta. Läpikäynnin aikana passiivisen listan solmut siirretään aktiiviseen listaan harmaaksi väritettyinä. Uudet oliot syntyvät mustina solmuina, eli ne selviävät roskienkeruusta. Lukumuurin aktivoituessa luettu olio siirretään aktiiviseen listaan. Kun läpikäynti on päättynyt, vierekkäiset joukot sulautetaan yhteen: uusien olioiden lista liitetään aktiiviseen listaan ja vanhojen olioiden lista (sisältää roskat) liitetään vapaiden muistialueiden listaan. Roskienkeruu päättyy joukkojen roolinvaihtoon.

4.4. Tehokkuuskysymykset

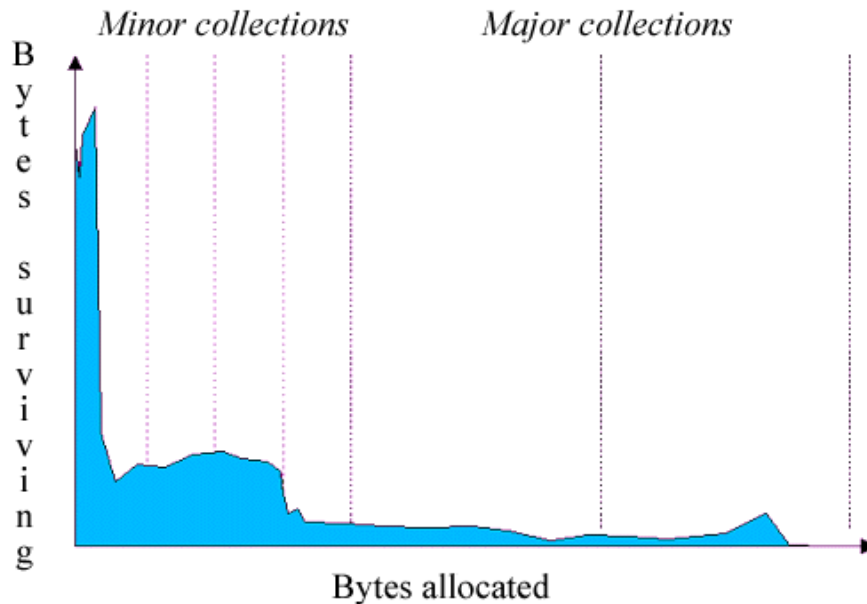
Kuten luvussa kaksi todettiin, optimoinnissa on aina (vähintään) kaksi vastakkaista näkökulmaa. Kun puhutaan vähittäisten algoritmien tehokkuudesta, nämä näkökulmat ovat tarvittava laskenta (*overhead*) ja syntyneiden ja lakaistujen roskien määrän mittasuhte. Optimaalisuudesta voidaan puhua vasta kun olemme määritelleet kahden näkökulman tärkeyssuhteen, joka on kuitenkin sovelluskohtainen asia. Yleistäen voidaan todeta, että jos sovellus luo jatkuvasti olioita, jotka tulevat lyhyessä ajassa tarpeettomiksi, saattaa olla parempi käyttää "huolellista" algoritmia, joka tarkistaa myös uusien olioiden tarpeellisuuden. Yleistys ei aina kuitenkaan päde, sillä "huolimattomat" algoritmit vaativat vähemmän laskentaa, siksi niitä voidaan suorittaa nopeammin ja useammin. Jos sovellus synnyttää pitkäikäisiä olioita, tiukat kooridinoitointimenpiteet ovat vain haitaksi, ja huolimattoman algoritmin käyttö on kannattavaa.

5. Ikäpohjainen roskienkeruu

Sukupolvittaisen tai ikäpohjaisen roskienkeruun idea perustuu jo aiemmin mainittuun havaintoon siitä, että suurin osa ohjelman luomista olioista ovat väliaikaisia, eli ne tulevat lyhyessä ajassa tarpeettomiksi, kun taas pieni osa olioista on pitempään aikaan käytössä. Ilmiö näkyy myös kuvasta 2 [Sun, 2003].

Ikäpohjaiset menetelmät ryhmittävät oliot sukupolviin, ja huolehtivat roskienkeruusta sukupolvikohtaisesti. Mitä nuorempi sukupolvi, sitä useammin siitä kerätään roskat. Etenkin kopioivat algoritmit hyötyvät tällaisesta

olioiden kategorisoinnista, sillä pitkäikäisten olioiden jatkuva kopiointi on kallista. Uudet oliot jäävät suurella todennäköisyydellä lyhytikäisiksi, niitä ei siksi tarvitse moneen kertaan kopioida, jos taas ne saavuttavat tietyn iän, ne siirtyvät seuraavaksi vanhempaan sukupolveen. Tekniikkaa voidaan verrata kullanhuuhtontaan. Tässä luvussa tarkastellaan tarkemmin miten ikäpohjaista roskienkeruuta voi tehostaa.



Kuva 2. Tyypillinen olioiden ikäjakauma [Sun, 2003].

5.1. Olioiden ikä ja sukupolvien lukumäärä

Sukupolvittainen roskienkeruu tarjoaa mahdollisuuden valita sopiva algoritmi jokaista ikäluokkaa varten. Ajatellaan kuitenkin yksinkertaisempaa esimerkkiä, joka käyttää kopioivaa algoritmia. Muisti jaetaan esimerkiksi kolmeen osaan, joista jokainen vielä puolitetaan tiivistäen kopioivaa algoritmia varten. Kun nuorimman sukupolven aktiivinen muistipuolisko täyttyy, aloitetaan roskienkeruu tavalliseen tapaan, mutta kynnyksikää saavuttaneet oliot kopioidaan keskimmäisen sukupolven aktiiviseen puoliskoon. Keskimmäinen sukupolvi siivotaan samalla tavalla, mutta harvemmin. Vanhimpaan sukupolveen päässeet oliot ovat de facto pitkäikäisiä.

Edellisessä esimerkissä on kuitenkin monta sellaista yksityiskohtaa, jota kannattaa miettiä tarkemmin. Eräs tällainen yksityiskohta on sukupolvien lukumäärä ja nille varattujen muistialueiden kokojen suhde. Eri sukupolville

voidaan valita myös sopiva kynnyksikä, lisäksi joudutaan miettimään keinoja olioiden iän tallentamiseen.

Sukupolvien lukumäärä ja kynnyksikä liittyvät läheisesti yhteen: mitä suurempi määrä sukupolvia on käytössä, sitä nopeammin olioita kannattaa ylentää. Ääritapauksessa roskienkeruusta selviävät oliot ylenevät suoraan seuraavaan sukupolveen. Edellisessä esimerkissä tämä järjestely johtaisi siihen, että vain vanhimman sukupolven muistia pitäisi puolittaa – nuorempien sukupolvien kohdeavaruus olisi seuraavan sukupolven lähdeavaruus. Tämä voi olla aika tehokas ratkaisu, mutta täytyy muistaa, että sukupolvia täytyy olla melkoinen määrä ennen kuin nuorimman ja vanhimman sukupolven oliot eroavat iältään merkittävästi.

Kuten edellä todettiin, algoritmi voidaan valita sukupolvikohtaisesti. Edellisessä esimerkissä voi tilan säästämiseksi käyttää tiivistävää algoritmia vanhimmassa tai vanhimmissa sukupolvissa. Roskienkeruu kannattaa tehdä kuitenkin vähittäin, koska roskien etsiminen vanhoista sukupolvista koskettaa muistia kokonaisvaltaisesti. Nuorissa sukupolvissa stop-and-go -tyyppinen keruu voi olla tarkoituksenmukaisempaa, koska se joudutaan tekemään usein ja nopeasti.

5.2. Viittaukset sukupolvesta toiseen

Tähän asti emme kiinnittäneet mitään huomiota ikäpohjaisen roskienkeruun suurimpaan tekniseen haasteeseen, sukupolvien rajoja ylittäviin viittauksiin, joihin törmätään väistämättä. Roskia ei voi kerätä mistään sukupolvesta tuntematta kaikkia niitä viittauksia, jotka kohdistuvat muista sukupolvista kyseiseen sukupolveen. Yleensä voidaan tehdä sellainen olettaus, että vanhoista sukupolvista viitataan vähemmän nuorempiin sukupolviin kuin toisin päin. Viittauksista vanhempiin sukupolviin ei kannata pitää rekisteriä, vaan roskien etsintä vanhoista sukupolvista lähtee yleensä liikkeelle nuorimmasta sukupolvesta, sen sijaan joissakin toteutuksissa viittauksista nuorempiin sukupolviin pidetään kirjaa jollakin tavalla.

Yksi tällainen tapa on uudelleenohjaustaulukon käyttö. Sukupolven olioihin viitataan muista sukupolvista välillisesti, ja taulukon alkiot lisätään roskienkeruussa juurijoukkoon. Ratkaisu ei ole kuitenkaan kovin tehokas, koska muistin lisäksi se vaatii paljon myös ylimääräisiä kellosyklejä, sillä osoittimen purku tai päivitys laukaisee aina luku- tai kirjoitusmuurin.

Pelkällä kirjoitusmuurilla pärjätään, jos tallennetaan jokaisen muuttuneen viittauksen osoite ja tarkistetaan vasta roskienkeruun yhteydessä onko kyseessä sukupolvien rajoja ylittävä viittaus. Uudelleenohjaustaulukko on edelleen tarpeen, mutta se ei enää ole käytössä muuttajan toiminnan aikana, vaan yksinomaan roskienkerääjän toimiessa.

Kaikissa tapauksissa ei yllämainittu oletus viittauksista vanhoista sukupolvista nuorempiin päde. Ajatellaan esimerkiksi suurta säiliöoliota, esimerkiksi listaa, vektoria tai suurta taulukkoa, jonka ohjelma luo käynnistyessään ja käyttää loppuun asti, mutta vaihtaa sen sisältöä jatkuvasti varaamalla siihen uusia olioita. Tällöin vanhasta sukupolvesta viitataan jatkuvasti nuorimpaan, siksi kirjoitusmuuri aiheuttaa suuria kustannuksia kellokykleinä.

Sama pätee tietenkin sukupolvittaisen roskienkeruun perusolettamukseen, eli siihen, että olioiden ikä ja tarpeellisuus ovat rinnastettavissa. Jos jostakin syystä tämä oletamus ei toteudu, tuhlataan sukupolvien hallinnointiin todella paljon järjestelmän voimavaroja. Tämä on erityisesti totta kun vähittäinen algoritmi on käytössä.

6. Roskienkeruu ohjelmointikielissä

Roskienkeruusta sanotaan usein, että se vapauttaa ohjelmoijaa muistinhallinnan vastuusta kokonaan, mutta se ei sellaisenaan pidä paikkaansa. Roskienkeruumekanismien tarkka tuntemus on tarpeen kun tehdään muistinhallintaa koskevia ratkaisuja sekä suunnittelu että ohjelmointivaiheessa. Kerääjä on vastuussa tarpeettomien olioiden siivoamisesta, mutta sille on tehtävä selväksi mikä on tarpeellista ja mikä ei. On tärkeää ymmärtää olioiden elinkaari: oliot syntyvät, ovat käytössä, tulevat näkymättömiksi, minkä jälkeen ovat mahdollisesti heikosti, pehmeästi tai (Java-kielessä) haamuviihättäviä, viimein tulevat juurijoukosta täysin irrallisiksi, viimeistellyiksi ja lopulta lakaistuiksi. Tämä pätee tässä luvussa esitettyihin kieliin (.Net, Java, C++).

6.1. Roskienkeruutietoinen ohjelmointityyli

Oliot saattavat jäädä näkymättömyyden tilaan todella pitkiksi ajoiksi. Niin voi käydä esimerkiksi ohjelmasilmukassa luoduille olioille, jotka tulevat käytännössä kerääjän näkyviin vasta aliohjelman suorituksen päätyttyä. Kaikissa toteuksissa ei ole tällaista vaaraa, mutta tarpeettomaksi tulleet oliot on aina syytä myös merkitä tarpeettomiksi poistamalla niihin kohdistuvat viittaukset.

Myös erityyppiset heikot viittaukset (*weak reference*) kertovat kerääjälle olioiden tarpeellisuudesta. Roskienkerääjä ei ota heikkoja viittauksia huomioon päättäessään olioiden tarpeellisuudesta, eli pelkästään heikosti viitattut oliot ovat kerättävissä. Heikojen viittaustan kaksi merkittävintä sovellusta ovat kätkömuistit tai heikot hakurakenteet (*cache* ja *weak dictionary*) ja syklisistä rakenteista johtuvien kustannusten vähentäminen [Drake, 2001]. Kätkömuistien avulla voidaan pitää kirjaa tietyistä olioista estämättä roskienkerääjää tekemästä työtään. Syklisten rakenteiden purku on usein kallis toimenpide. Ajatellaan esimerkiksi DOM-oliopuita, jotka ovat kahteen suuntaan linkitettyjä puura-

kenteita, joista oksan poistaminen on raskas iteroiva operaatio. Jos taas kaikki lehdistä juureen päin osoittavat viittaukset toteutetaan heikkoin viittauksin, `unlink`-operaatio tulee koostumaan yhden osoittimen nollauksesta.

Pehmeästi viitattuja (*softly referenced*) olioita kerätään vain jos niiden käyttämää muistia tarvitaan muuhun. Kannatta siis viitata pehmeästi niitä olioita, joiden luonti ja viimeistely ovat kalliita operaatioita. Java-kielessä [Sun, 2002b] voidaan käyttää myös haamuviittausta (*phantom reference*), josta ei voi koskaan saada enää vahvaa viittausta viitattuun olioon, koska olio on jo viimeistely ja lakaistavissa. Tieto siitä, että näin on tapahtunut, voi silti olla arvokasta.

Olioiden viimeistely suoritetaan asynkronisesti ennalta-arvaamattomana hetkenä. Viimeistelijä ei siis ole sama kuin olion lopetusoperaatio (tuhoajametodi). Viimeistelijää tarvitaan todella harvoin. Sen ainoana tehtävänä on vapauttaa olion ohjelman ulkopuolelta (esimerkiksi käyttöjärjestelmästä) varaamia voimavaroja. Toisaalta mikään ei takaa, että olio tulee keräytyksi ja viimeistellyksi ennen kuin ohjelman suoritus päättyy [Eckel, 2000]. Siksi ohjelmoijan on syytä huolehtia siitä, että heti kun oliota ei tarvita enää, se siivoaa jälkensä. Viimeistelijät hidastavat muistin kierrätystä, siksi niitä ei pidä määrittellä tarpeettomasti. Viimeistelijämetodissa olio voi "herätä kuolleista", tai herättää muita. Boehm [2003] määritteli viimeistelyyn liittyviä sääntöjä, jotka varmistavat roskienkeruun turvallisuuden. Huomattavaa on, että olemassaolevat toteutukset eivät täytä kaikilta osin Boehmin vaatimuksia, mutta asiaan ei syvennyttä tässä yhteydessä tämän enempää.

6.2. .Net

Microsoftin .Net-kielen ajoaikainen järjestelmä (CLR, Common Language Runtime) kerää roskat kolmesta sukupolvesta käyttäen tiivistävää (mark-compact) algoritmia [Kukreja and Nair, 2001]. Hyötylaskenta pysähtyy kokonaan roskienkeruun ajaksi. Microsoft ei paljasta mitä kriteerejä CLR käyttää päättäessään koska roskienkeruu suoritetaan.

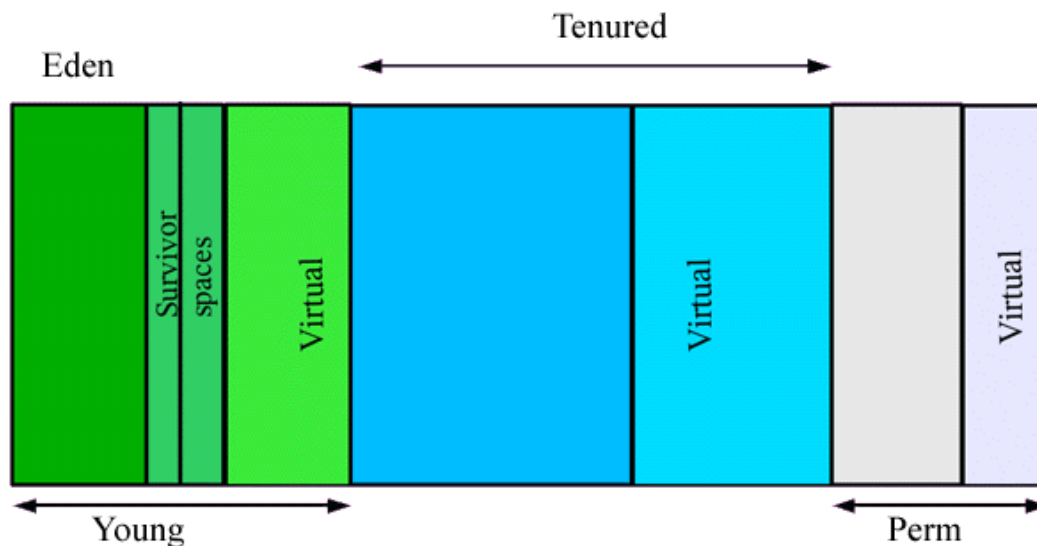
Viimeistely .Net-kielessä tapahtuu erikoisella tavalla. Kaikki viimeistelyä vaativat oliot, eli niiden luokkien instanssit, jotka ylimäärittelevät metodia `Object.Finalize`, rekisteröidään syntyessään CLR:n sisäiseen tietorakenteeseen, viimeisteltävien olioiden jonoon (*Finalization queue*). Roskienkeruun merkintävaiheen aikana kerääjä tarkistaa löytyykö viimeistelyjonosta sellaisia olioita, joita se on todennut roskiksi. Kaikki tällaiset oliot siirretään viimeisteltävien olioiden jonosta varsinaiseen viimeistelyjonoon (*Reachable queue*) eikä kerääjä pidä niitä enää tässä vaiheessa roskina, vaan ne "nousevat kuolleista" (*resurrect*). Tämän jälkeen muisti tiivistetään ja viimeistelyjono tyhjennetään suorittamalla jonossa olevien olioiden viimeistelijät. Viimeistelyjen olioiden varaama tila vapautetaan vasta seuraavan roskienkeruun yhteydessä.

Ympäristö tukee myös pehmeitä viittauksia luokan `WeakReference` muodossa. Viittaustyyppi on nimestään huolimatta nimenomaan ”pehmeä” eikä ”heikko”, koska CLR kerää ei-vahvasti viitattut oliot vasta silloin kun muisti loppuu. Ohjelmoija voi päättää siivotaanko heikosti viitattu olio vasta viimeisteltynä tai heti roskaksi toteamisen jälkeen. Luokan `WeakReference` oliot ovat ajoympäristön sisäisiä tietorakenteita, eikä niitä luoda varaamalla niille muistia dynaamisesti. Heikko viittaus on itse asiassa kahva sisäisen taulukon alkioon. Viimeistelytarpeensa mukaan heikot viittaukset tallennetaan yhteen kahdesta taulukosta (*long weak reference table* ja *short weak reference table*).

Omasta mielestäni Microsoftin salailu roskienkeruun toteutuksen suhteen on haitallista ja varsin tarpeetonta, sillä sen teknologia ei edusta alan huipputasoa.

6.3. Java

Roskienkeruu on tullut Java-kielen mukana yleiseen tietoisuuteen. Java virtuaalikoneen määrittelyn kirjoittajat Lindholm ja Yellin [1999] tottavat, että Java-kielessä on käytössä roskienkeruu, eikä olioita vapauteta koskaan eksplisiittisesti. Lisäksi virtuaalikoneen toteuttaja on vapaa valitsemaan mieleisensä automaattisen muistinhallintamekanismin. Toteutukset eroavatkin toisistaan huomasti. Tässä yhteydessä keskitytään kuitenkin Sun microsystems:n virtuaalikonetoteutukseen (jdk 1.4.1 ja 1.4.2) [Sun, 2002a].



Kuva 3. Muistin rakenne Java-kielessä (JDK 1.4.2) [Sun, 2003].

Roskienkerääjää voidaan valita Java-tulkin käynnistyksen yhteydessä neljästä eri vaihtoehdosta. Kaikki neljä kerääjää perustuvat samaan ikäpohjaiseen muistinhallintamalliin. Muisti on jaettu kahteen sukupolveen (ks. kuva

3), joista molemmat ovat tarvittaessa laajennettavissa. Nuoren sukupolven (*young generation*) muistialue on jaettu kolmeen osaan. Kaksi osaa (*survival spaces*) ovat tiivistäen kopioivaa roskienkeruuta varten, kolmas alue on nimetty Eedeniksi (*Eden*), ja se on varattu uusille olioille. Kuten edellisen luvun kuvasta 2 voidaan päätellä, suurin osa Eedenin olioista ei selviä ensimmäisestä roskienkeruusta. Vanha sukupolvi (*tenured generation*) on tarkoitettu siivottavaksi merkitse ja lakaise -menetelmällä. Muistin ja kaikkien eri muisti-alueiden kokoa ja niiden mittasuhteita voidaan säätää.

Jos oletuskerääjä ei ole riittävän tehokas sovelluksen tarpeisiin, voidaan ottaa optimoitu kerääjä tilalle. Optimoinnin kohteena voi olla läpäisykyky tai taukojen pituus hyötylaskennassa. Läpäisykykyä voidaan monisuoritin-laitteissa lisätä *throughput collector* -kerääjän avulla, joka tukee myös adaptiivisia säätöjä: kerääjä voi hyödyntää toiminnassaan pitkältä ajalta automaattisesti keräämänsä tilastollista aineistoa sovelluksen käyttäytymisestä. Katkojen pituutta voidaan lyhentää rinnakkaisen kerääjän (*concurrent collector*) avulla. Kerääjä ajetaan omassa säikeessään ja se keskittyy siivoamaan vanhaa sukupolvea. Ohjelman muut säikeet pysähtyvät keruun aikana vain kahdesti ja silloinkin vain lyhyeksi ajaksi. Toinen vaihtoehto katkojen lyhentämiseksi on vähittäinen kerääjä (*incremental tai train collector*). Tämä kerääjä sopii koneisiin, joissa on vain yksi suoritin. Vanhan sukupolven siivouksen vaatima aika jaetaan pienempiin osiin siivoamalla vanhaa sukupolvea vähittäin nuoren sukupolven siivouskertojen yhteydessä. Menetelmä vaatii niin paljon koordinoitua, että ohjelman suorituskyky heikkenee väistämättä. Jos tästä ei ole suurempaa haittaa, vähittäisen kerääjän käyttö on tehokas tapa parantaa ohjelman vasteaikoja.

Roskienkeruusta Java-kielessä on saatavissa todella runsaasti tietoa. Kiinnostuneelle lukijalle suosittelen tässä luvussa käytettyjä lähteitä.

6.4. C++

Perinteisten imperatiivisten kielten tavoin C ja sen olioperustainen laajennos C++ käyttävät eksplisiittistä muistinhallintaa. Viimeisen vuosikymmenen aikana on esitetty useita kokonaisvaltaisia ratkaisuja ja ehdotuksia roskienkeruun integroimiseksi näihin kieliin, erityisesti jälkimmäiseen. Ellis ja Detlefs [1993] esittivät C++-kieleen roskienkeruujärjestelmän, joka lienee tunnetuin ja kattavin ratkaisu, josta on myös otettu paljon mallia (yhtenä esimerkkinä Sun microsystemsin libgc-kirjasto). Esitys pohjautuu Bartlettin (1989) ja Boehmin (1991) tekemään esityöhön ja Zornin tehokkuusmittauksiin (1992).

Ellis ja Detlefs lähestyivät asiaa holistisesti järjestelmätasolla ja määrittivät järjestelmälle neljä osa-alueita, jotka ovat

- rajapinta ohjelmointikieleen (*language interface*),

- turvallisten ohjelmarakenteiden alijoukko (*safe subset*),
- käytettävät algoritmit (*collection algorithms*) ja
- turvallisen koodin generointi (*code generation safety*).

Esityksen kulmakivinä ovat järjestelmälle asetetut turvallisuusvaatimukset, lisäksi vaatimukset mahdollisimman pienistä muutoksista kieleen, sen toteutukseen ja ohjelmointityyliin sekä perinteistä muistinhallintaa käyttävien ja roskienkeruun alaisten moduulien yhteensopivuuden takaaminen. Myös toteutuksen kannettavuuteen ja tehokkuuteen kiinnitetään huomiota.

Turvallisen koodin generointi taataan kahden käännösaikaisen rajoitteen, kymmenen käännösaikaisen ja kuuden ajoaikaisen tarkistuksen avulla. Kääntäjien ja ajoaikaisten järjestelmien täytyy toteuttaa nämä rajoitteet ja tarkistukset, jotta ne olisivat roskienkeruuturvallisia (*GC-safe*). Kaikki tarvittavat toimenpiteet voidaan tehdä välikoodin generoinnin yhteydessä (*front-end compilation*).

Turvallisten ohjelmarakenteiden alijoukko on C++-kielen syntaksin alijoukko, jota käyttämällä roskienkeruun virheettömyys voidaan taata. Ohjelmoija merkitsee erityisellä `pragma` määreellä ne lähdekoodin osat, jotka on tarkoitettu roskienkeruuturvallisiksi.

```
#pragma safe

//turvallinen koodi

#pragma unsafe
```

Kääntäjä huolehtii siitä, että kaikki ohjelmarakenteet ovat turvallisia ja turvallisuussäännöt pysyvät voimassa myös ajoaikana. Tämä tarkoittaa käytännössä suuren joukon ylimääräisiä ajoaikaisia tarkistuksia, esimerkiksi osoitinmuuttujien alkuarvon ja arvomuutosten validiteetin osalta.

Roskienkeruun alaiset oliot erotetaan muista olioista (*collected / non-collected objects*) omaan (loogiseen) kekkoon kieleen lisätyn `gc`-määreen avulla. Määrettä käytetään luokan tai muuttujan esittelyn yhteydessä, siten että osoitin T^* on identtinen osoittimen `gc T*:n` kanssa. Toisin sanoen määre ei vaikuta tyyppitykseen tai tyyppinmuunnoksiin, mikä takaa yhteensopivuuden keräämättömien moduulien kanssa.

Muista moduuleista tuoduista luokista voi luoda kerättäviä instansseja `new`-operaattorilla, tai yleisesti `typedef`-operaattorin avulla:

```
new gc T;

typedef gc T TGC;

new TGC;
```

Seuraava asetelma on myös mahdollinen:

```
gc class D: C {...};
```

Luokka C on tuotu moduulista, jossa ei ole valmiutta käyttää roskienkeruuta tai toimia monisäikeisessä ympäristössä. Vaarana on, että luokan C lopetusoperaatio suoritetaan asynkronisesti roskienkeruun yhteydessä tuntemattomin seurauksin. Tällöin voidaan määritellä luokalle D jono, johon viimeisteltävät oliot kerätään. Ohjelma voi tyhjentää jonon turvallisessa suorituksen vaiheessa.

Kerättäviä olioita voi vapauttaa `delete`-operaattorilla, mutta varsinaisen viimeistelytyön tekee kerääjä. Lopetusoperaattori (tuhoajametodi) toimii oletusarvoisena viimeistelijänä, mutta viimeistelijämetodi voidaan määritellä kerättäville luokille erikseen. Roskienkeruuta tukevat luokat toteuttavat automaattisesti yleisen (template) rajapinnan, joka sisältää muun muassa kaikki yllämainitut viimeistelyyn liittyvät operaatiot. Myös heikot viittaukset toteutetaan vastaavalla tavalla.

Roskienkeruun toimivuus Javassa on antanut puhtia pyrkimyksille lisätä roskienkeruu C++-kieleen ja joitakin hyviä toteutuksia on jo nyt saatavilla. Valitettavasti ANSI-standardointia joudumme luultavasti odottamaan vielä todella pitkään.

7. Yhteenveto

Automaattisen muistinhallinnan käyttö ohjelmistotuotannossa on yleistymässä. Koska roskienkeruu on usein järjestelmän toimivuuden pullonkaulana, algoritmin valinnalla ja hienosäädöllä on ratkaiseva vaikutus siihen, pystyykö ohjelma täyttämään sille asetetut vaatimukset. Vaatimuksia voidaan asettaa läpäisykyvyn, muistinkäytön ja vasteaikojen suhteen. Vasta kun näiden tekijöiden tärkeysuhde on määritelty voidaan etsiä tilanteeseen sopiva algoritmi. Oleellisen tärkeää on myös roskienkeruumekanismien huomioon ottaminen muistinhallinnan suunnittelu- ja toteutusvaiheessa.

Algoritmien esittelyn yhteydessä esiteltiin myös niiden käyttöön liittyviä ongelmia. Esimerkiksi merkiste ja lakaise -tyyppisten algoritmien ongelmana on muistin pirstoutuminen ja olioiden hajaantuminen muistissa. Ongelmaan tuovat helpotusta algoritmien tiivistävät muunnelmät, jotka vaativat kuitenkin enemmän laskentaa. Laskentakustannuksia voidaan vähentää käyttämällä tiivistäen kopioivaa menetelmää, jolloin maksettava hinta on pahimmillaan kaksinkertaistuva muistintarve.

Vähittäisen roskienkeruumenetelmän valinnassa huomioitavat seikat ovat

- menetelmän sallimat vasteajat,
- algoritmin huolellisuus, eli roskienkeruun päättyessä jäljelle jääneen kelluvan roskan määrä,
- tarpeettomaksi muuttuneen muistin vapauttamiseen kuluva aika
- ja vähittäisyyden tuoma ylimääräinen laskentakuorma.

Kelluvan roskan kannalta tehokkain menetelmä on kirjoitusmuurin käyttö (incremental update -muunnelma). Bakerin kopioiva algoritmi toimii myös melko tehokkaasti, mutta on ennalta-arvaamattomampi vasteaikojen suhteen. Tässä yhteydessä on kuitenkin vaikea tehdä yleistyksiä algoritmien soveltuvuudesta, koska kevyitä ja huolettomia algoritmeja käyttävä roskienkeruu voidaan suorittaa useammin, toisaalta raskasta koordinointia käyttäviä algoritmeja on helpompi keskeyttää.

Ikäpohjaisten menetelmien käyttö on yleensä kannattavaa, koska oletamus olioiden lyhytikäisyydestä pätee lähes aina. Menetelmien tehokkuuteen voivat vaikuttaa oleellisesti säädettävät ominaisuudet ja optimoitavat tekijät, muiden muassa

- sukupolvien lukumäärä ja
- yksittäisten sukupolvien osuus käytettävissä olevasta muistista,
- kynnyksiä,
- eri sukupolvissa käytetyt algoritmit,
- sukupolvien rajoja ylittävien viittausten rekisteröinti,
- roskienkeruun eskaloituminen sukupolvesta toiseen sekä
- menetelmän tai menetelmien vähittäisyys.

Alan kirjallisuudessa mainitaan joitakin nyrkkisääntöjä roskienkeruu-algoritmien soveluvuudesta tietyntyyppisiin sovelluksiin, mutta kattavaa soveltuvuus- ja sovellusluokittelua ei kuitenkaan ole vielä kukaan esittänyt. Lisätutkimuksen arvoista olisi myös adaptiivisen roskienkeruun kehittäminen tai roskienkeruun entistä parempi integroiminen C++-kieleen.

Viiteluettelo

- [Abdullahi *et al.*, 1992] S. Abdullahi, E. Miranda, and G. Ringwood, Collection schemes for distributed garbage. *Proceedings of the International Workshop on Memory Management (IWMM 92)* - LNCS 637. Springer-Verlag, September 1992, 43-81.
- [Bacon *et al.*, 2003] D. F. Bacon, P. Cheng, and V. T. Rajan, A real-time garbage collector with low overhead and consistent utilization. *Proceedings of the 30th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, ACM Press, 2003, 285-298.
- [Baker, 1978] H. G. Baker, List processing in real time on a serial computer, *Communications of the ACM*, **21** (4), Apr. 1978, 280-294.
- [Boehm, 2003] H-J. Boehm, Destructors, finalizers, and synchronization. *Proceedings of the 30th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, ACM Press, 2003, 262-272.

- [Cheney, 1970] C. J. Cheney, A nonrecursive list compacting algorithm, *Communications of the ACM*, **13** (11), Nov. 1970, 677-678.
- [Cohen, 1981] J. Cohen, Garbage collection of linked data structures. *Computing Surveys*, **13** (3), September 1981, 341-367.
- [Dijkstra *et al.*, 1978] E. W. Dijkstra, L. Lamport, A. J. Martin, E. F. M. Steffens, On-the-fly garbage collection: An exercise in cooperation, *Communications of the ACM*, **21** (11), 966-975.
- [Drake, 2001] Fred L. Drake, Jr., Weak References (Python Enhancement Proposal). Available as <http://www.python.org/peps/pep-0205.txt> [11.1.2001].
- [Eckel, 2000] Bruce Eckel, Thinking in Java. Available as http://www.codeguru.com/java/tij/tij_c.shtml [15.12.2003].
- [Ellis and Detlefs, 1993] J. R. Ellis and D. L. Detlefs, Safe, Efficient Garbage Collection for C++. Available at <http://gatekeeper.dec.com/pub/DEC/SRC/research-reports/abstracts/src-rr-102.html> [10.6.1993].
- [Johnstone, 1997] M. S. Johnstone, Non-Compacting Memory Allocation and Real-Time Garbage Collection. Ph.D. dissertation, The University of Texas at Austin, December 1997.
- [Kukreja and Nair, 2001] A. Kukreja and R. Nair, Garbage Collection in .Net. <http://www.dotnetextreme.com/articles/garbageCollector.asp> [31.10.2003].
- [Lindholm and Yellin, 1999] Tim Lindholm and Frank Yellin, *The Java Virtual Machine Specification, Second Edition*, luku 3.5.3. Addison-Wesley, 1999.
- [Sun, 2003] Sun Microsystems Inc, Tuning Garbage Collection with the 1.4.2 Java™ Virtual Machine, (performance documentation for the java hotspot virtual machine). <http://java.sun.com/docs/hotspot/gc1.4.2/index.html>, [18.09.2003].
- [Sun, 2002a] Sun Microsystems Inc, Java 1.4.1 HotSpot Virtual Machine White Paper, (Technical White Paper september 2002). Available as http://java.sun.com/products/hotspot/docs/whitepaper/Java_Hotspot_v1.4.1/Java_HSpot_WP_v1.4.1_1002_1.html [12.4.2003].
- [Sun, 2002b] Sun Microsystems Inc, Java™ 2 Platform, Standard Edition, version 1.4.1 API Specification, (package java.lang.ref). Available as <http://java.sun.com/j2se/1.4.1/docs/api/java/lang/ref/package-summary.html> [31.10.2003].
- [Wilson, 1992] P. R. Wilson. Uniprocessor garbage collection techniques. *International Workshop on Memory Management*, St. Malo, France, September 1992. (The Proceedings has been published as Springer-Verlag Lecture Notes in Computer Science no. 637).

Teknologian vaikutus opetukseen

Matti Voutilainen

Tiivistelmä

Tutkielmassa tarkastellaan onko suomalaisessa koululaitoksen opetusmenetelmissä tapahtunut mitään todellista muutosta 1900-luvun alusta tähän päivään, jos tieto- ja muun tekniikan vaikutus karsitaan siitä pois. Tarkastellaan miten terminologia on muuttunut, jos on muuttunut ja miten mahdollinen sanojen muutos tarkoittaa kuitenkin samoja asioita.

Opetusmenetelmiä lähestytään tarkastelemalla kunakin aikana opettajien koulutuksessa käytettyä oppimateriaalia. Tämän suhteen on selvä epäsuhta, sadan vuoden takaisia oppikirjoja ei ole kovinkaan paljon, mutta ne vähät mitä on ovat olleet aikanaan ratkaisevassa asemassa. Tämän päivän oppimateriaalia on paljon, mutta mikään ei ole erityisasemassa.

Tutkielma rajataan koskemaan koulutusta, jossa opiskelijat ovat korkeintaan noin 20 vuoden ikäisiä. Perusteena on ajatus, että kunakin aikana on opettajakoulutuksessa käytetty oppimateriaali jotenkin kuvaa oman aikansa opetusmenetelmiä ja tapoja. Opettajakoulutus oli 1900-luvun alussa keskittynyt hyvin vahvasti seminaareihin joista valmistui opettajia lähinnä sen aikaisen kansakoululaitoksen tarpeisiin.

Lisäksi käydään lyhyesti läpi suomalaisen koululaitoksen kehitystä, koska koululaitoksen yleinen tila kuvaa arvopohjaa ja oppimiskäsitystä, jolle opetusmenetelmät pohjautuvat.

Avainsanat ja -sanonnat: arvot, kasvatus, opetus, koulu, opetusmenetelmä

CR-luokat: K.3.1

1. Johdanto

Oppimisella käsitetään yleensä olion kykyä ottaa informaatiota vastaan ympäröivästä maailmasta ja muuttaa käyttäytymistään niin, että olion hyöty maksimoituu. Tämä oppiminen voi olla tietoista, kuten ihmisellä tai ns. vaistonvaraista toimintaa kuten eläimillä. Ihminen voi haluta tietoa ja opiskella tiedon itsensä takia ilman, että siitä on ainakaan heti välitöntä hyötyä.

Oppimistavat ja kyvyt ovat ihmiselle kehittyneet vuosimiljoonien aikana pienissä yhteisöissä välittömässä vuorovaikutuksessa toisten ihmisten kanssa. Opittavia asioita ovat olleet kaikki luontoon ja sitä kautta ravintoon liittyvät asiat. Toinen tärkeä oppimisen alue on ollut yhteisön jäsenenä toimiminen,

koska hyvin toimivalla ja yksimielisellä heimolla on ollut paremmat mahdollisuudet turvata yksilönkin toimeentulo.

Heimoyhteiskunnan aikaan kaikkalainen oppiminen tapahtui yhteisössä toimimisen yhteydessä. Nykyaikaan sanottaisiin, että se oli työssä oppimista. Opittiin monia asioita yhtä aikaa eli oppiminen oli integroitua eri oppiaineiden suhteen. Lisäksi koko yhteisö joitakin asioita samaan aikaan, kyseessä oli tällöin oppiva organisaatio tai yhteisöllinen oppiminen.

Yhteiskunnan tullessa yhä monimutkaisemmaksi yksilöiden toiminnat eriytyivät toisistaan, muodostui eri ammatteja ja toimintoja joiden perusteita oli hankala oppia ilman etukäteistä valmennusta. Ongelma ratkaistiin tekemällä erilaisia instituutioita, joiden tehtäväksi tämä valmentaminen annettiin.

Näihin instituutioihin syntyi uusi ammattikunta, opettajat, joiden tehtävänä oli jakaa tietoa oppilaille ja saada aikaan heissä yhteiskunnan toivomia muutoksia.

Tässä tilanteessa opettajille alkoi muodostus tapoja ja ajatuksia, joiden avulla he voisivat saada parhaiten opetettavat asiat perille, alkoi syntyä oppimiskäsityksiä ja opetusmenetelmiä.

Yhteiskunnan edelleen kehittyessä muodostui instituutio, jonka tarkoituksena oli opettajien valmentaminen tulevaan ammattiinsa. Niiden perustaminen sai kasvatukseen, opettamiseen ja oppimiseen liittyvän järjestelmällisen ja määrätietoisen tutkimuksen ja kehittämistoiminnan käyntiin. Pääpiirteissään voidaan sanoa, toimintojen painotus oli edellä mainitussa järjestyksessä. Ensin pohdittiin kasvatukseen liittyviä tekijöitä, sitten opetusmenetelmiin ja varsinaisia oppimiseen liittyviä seikkoja alettiin tutkia suuremmissa määrin viimeiseksi. Järjestys kuvaa yhteiskunnan arvomaailman muutoksia laajemminkin. Yhteiskunnan kiristäessä otetaan kansalaistaan 1700-luvulta alkaen oli tärkeää kasvattaa heitä yhteiseen muottiin ja ns. yhteiskunta kelpoisiksi kansalaisiksi. Tähän kasvatustyöhön pyrittiin sitten kehittämään sopivia menetelmiä. Yksilön oikeudet yhteiskunnan ohi alkoivat varsinaisesti nousta esille vasta toisen maailmansodan jälkeen. Silloin alettiin pohtia miten varsinainen oppiminen tapahtuu, mitä tapahtuu aivojen hermostossa ja miten ihminen muodostaa käsityksiä erilaisista havainnoistaan.

2. Suomen koululaitoksen kehitystä

2.1. Perusopetus

Turussa on ollut ensimmäinen Suomen koulu. Varhaisimmat maininnat siitä ovat 1300-luvulta, mutta on arveltu, että se aloittanut toimintansa jo aikaisemmin [Iisalo,1991]. Koulua hallinnoi roomalaiskatolinen kirkko. Keskiajan loppupuolelle maahamme perustettiin ns. kaupunkikouluja, joita

1500-luvun lopulla oli Viipurissa, Raumalla ja Porvoossa. Nämäkin koulut olivat tiukassa kirkon ohjauksessa. Tärkeitä opiskeltavia asioita tuohon aikaan oli latina, teologia ja antiikin kirjallisuus. Johtavia kasvatuksellisia tavoitteita oli totteleminen, ehdoton usko Jumalalliseen totuuteen lisäksi pohdittiin millaiset ihmiset voivat pelastua.

Humanismi alkoi saada sijaa yhteiskunnassa 1500-luvulla ja tämä heijastui tietenkin myös koululaitoksen toimintaan. Tavoitteena oli monipuolisesti sivistynyt ihminen, jolle mikään, mitä hän ei oman ajattelunsa kautta voinut hyväksyä, ei ollut totta. Latina säilyi edelleen tärkeimpänä oppiaineena, koulussa saattoi 51 viikkotunnista olla 32 tuntia latinaa. Muutos oli kuitenkin siinä, että opettaja pyrki saamaan selville oliko oppilaat myös ymmärtäneet lukemansa. Toinen periaatteellinen muutos kasvatuksellisiin tavoitteisiin oli, että kun aikaisemmin piti olla ehdottoman kuuliainen katoliselle kirkolle niin uskonpuhdistuksen jälkeen kuuliaisuutta piti osoittaa kuninkaalle. Tämä maallisti koulujen tavoitteita ja niihin lisättiin sivistyksen lisäksi yhteiskunnan hyötyä tavoittelevia sisältöjä. Pyrittiin kouluttamaan virkamiehiä kuninkaan ja valtion tarpeisiin papiston lisäksi. Päätäntävalta kouluissa siirtyi kirkolta valtiolle.

Ruotsi-Suomen koululaitos oli 1600-luvulla kolmiportainen. Alimpana oli pedagogiot eli lastenkoulut seuraavana oli triviaalikoulut ja ylimpänä kouluasteena ennen yliopistoja olivat lukiot. Latina oli edelleen tärkein oppiaine.

Vasta 1700-luvun lopulla ja 1800-luvun alkupuolella alettiin laajemmin ja vakavammin ajatella, että koulutuksella voitaisi tavoitella yhteiskunnalle ja sen jäsenille taloudellista hyötyä ja että koulutus olisi oltava kaikkien saatavilla. Kurjuuden ja köyhyyden syynä pidettiin tietämättömyyttä. Toiseksi tavoitteeksi tuli itsenäisesti ajattelevien ihmisten kasvattaminen. Aikansa valistuneet ihmiset alkoivat käyttää, jopa omia varojaan koulujen perustamiseen. Tamperelaisena yksityiskohtana voidaan mainita kruununvouti Gabriel Ahlman, joka testamenttasi omaisuutensa vuonna 1799 koulujen perustamista varten [Iisalo, 1991]

Tosin tällöinkin ilmeni ajatuksia, että tavalliselle kansalle ei saa antaa liiallista opetusta, koska liiallisen opin pelättiin vieraannuttavan rahvaan poistyö teosta ja pahimmillaan kapinallisten aatteiden leviämisen kansan pariin.

Keisari Aleksanteri II antoi vuonna 1858 asetuksen kansanopetuksesta. [Vahtola, 2002]. Tämä asetus aloitti kansakoulujen perustamisen pitäjiin. Koulut olivat tällöin täydellisesti irti kirkosta. Koulujen valvonta uskottiin pitäjäläisten valitsemalle johtokunnalle. Pitäjille ei kuitenkaan tässä vaiheessa määrätty velvollisuutta perustaa kouluja.

Pakolliseksi koulujen perustaminen tuli 15.4.1921, kun maahamme säädettiin oppivelvollisuuslaki. Se velvoitti kunnat perustamaan kuusiluokkaisia kansakouluja niin, että jokaisella lapsella on koulu enintään viiden kilometrin etäisyydellä kodistaan. Kansakoulu toimi tämän lain mukaan pääpiirteissään niin kauan, kun se oli olemassa eli 1970-luvulle saakka.

2.2. Oppikoulut

Oppikoulut ovat toimineet pohjakouluina yliopisto-opintoja varten. Suomessa oppikoulujen kehitys varsinaisesti alkoi 1800-luvun alkupuolella. Koulujen opetussisältöjä suunniteltiin lähes yksinomaan humanismin hengessä.

Hallinnollisesti koulut olivat valtion ohjauksessa. Oppikoulu oli tavallaan jaettu kolmeen eritasoon. Alimpana oli ala-alkeiskoulu, joka oli eräänlainen kansakoulun ja ylempien oppikoulujen välimuoto. Seuraavana oli neliluokkainen yläalkeiskoulu. Kolmantena tasona oli lukio, josta sitten saattoi siirtyä yliopistoon.

Keisari Aleksanteri II antoi vuonna 1873 asetuksen, jolloin kolmiportainen oppikoulu korvattiin kahdeksanluokkaisella lyseolla. Lyseoon ei ollut varsinaista pohjakoulutus vaatimusta, mutta tulijoilla oli oltava riittävät perustaidot lukemisessa. Lyseot oli tarkoitettu ainoastaan pojille. Vasta vuonna 1905 asetuksella säädettiin, että oli käytävä osa kansakoulusta ennen lyseoon siirtymistä. Asetusta tulkittiin niin, että lyseon pohjavaatimukseksi tuli neljä luokkaa kansakoulua. Tämä tulkinta säilyi niin kauan kuin rinnakkaiskoulujärjestelmä oli olemassa. Oppikoulut olivat aluksi enimmäkseen ruotsinkielisiä, mutta jo 1880-luvulla oli suomenkielisissä oppikouluissa enemmän opiskelijoita kuin ruotsinkielisissä.

Seuraava oppikoulua koskeva muutos tapahtui 1914, jolloin lyseo jaettiin keskikouluun ja lukioon. Aikaa myöten tuli tyttökouluja ja yhteiskouluja, mutta koulutuksen rakenne säilyi samana lyseon kanssa peruskoulun tulon saakka.

2.3. Ammatillinen opetus

Suomessa oli laki ammattikuntalaitoksesta voimassa vuoteen 1868 saakka. Kaupungeissa oli saman käsityöammatin harjoittajille virallinen yhteenliittymä, ammattikunta. Nämä ammattikunnat valvoivat opetusta omalla alallaan sekä määrällisesti, että laadullisesti.

Oppipojaksi pääsi 14-vuotiaana ja opiskeluaika oli noin viisi vuotta. Sen päätteeksi hänen oli suoritettava kisällintutkinto. Kisällin oli lähdettävä muutaman vuoden kierrokselle opiskellakseen ammattia lisää muissa yrityksissä ja muilla paikkakunnilla. Joissakin ammateissa kisälli veloitettiin

olemaan määrätty aika ulkomailla ja hän ei saanut kisällikierroksen aikana tulla esim: 50 kilometriä lähemmäksi kaupunkia, jossa oli ollut oppipoikana. Hän oli eräänlainen vaihto-oppilas, kansainvälistyi.

Mestariksi pääsi noin viisi vuotta kestävän kisälliajan jälkeen, kun osoitti kunnollisuutensa ja suoritti työnäytteen. Ainoastaan mestari saattoi toimia itsenäisenä ammatinharjoittajana.

Vuonna 1842 säädettiin asetus, jolla kisällit ja oppipojat velvoitettiin hankkimaan itselleen luku- ja kirjoitustaito sekä perustiedot neljästä laskutavasta. Tällöin alettiin pitää ns. sunnuntaikouluja, joissa noita taitoja opetettiin.

Ensimmäiset koulumuotoiset ammattikoulut, joita sanottiin käsityökouluiksi, perustettiin vuonna 1886 Maarianhaminaan ja Turkuun. Varsinainen ammattikoulu, jonka pääsyvaatimuksena oli kansakoulun käyminen perustettiin Helsinkiin vuonna 1898. Ammatillinen koulutus muuttui 1900-luvun aikana hyvinkin monella, koska se sitten koulun ylläpitäjää tai tavoitteita. Syntyi erilaisilla tavoitteilla varustettuja kouluja: ammatteihin valmistava koulu, yleinen ammattikoulu, ammattioppilaskoulu, erikoisammattikoulu, konepajakoulu jne... Lisäksi oppisopimusjärjestelmä oli toiminnassa ja on edelleenkin vaikkakin sen painoarvo on vaihdellut huomattavasti eri aikoina.

Tällä hetkellä ammatillinen koulutus on toisen asteen koulutusta ja siellä suoritetaan ammatillisia perustutkintoja. Koulujen toiminta perustuu vuonna 1998 säädettyyn lakiin toisen asteen ammatillisesta koulutuksesta. Koulut voivat lain mukaan valita nimensä melko vapaasti ja sen takia on syntynyt erilaisia instituutteja ja ammattiopistoja vaikka kaikki toimivat samojen tavoitteiden mukaisesti.

Ammatillisen koulutuksen tavoite on ollut kaikkina aikoina hyvin selkeästi ammatin oppiminen. Yleisiä oppiaineita oli ja on edelleen: äidinkieli, vieras kieli, matematiikka, fysiikka, kemia, mutta kaikkien näiden sisällöt suuntautuvat opiskeltavan ammatin mukaisesti. Yleisiä kansalaisuuteen kasvattavia aineita on yhteiskuntaoppi ja taidekasvatus.

2.3. Opettajakoulutus

Joskus 1700-luvulla opettajan kelpoisuusehdot oli määritelty, että opettajan pitää olla köyhä ja muuhun työhön kykenemätön. Sitten kuitenkin havaittiin, että opettajalta vaadittiin muitakin avuja. Vuonna 1783 H. G. Porthan alkoi pitää kasvatusopillisia luentoja kotiopettajiksi aikoville.

Varsinaisen opettajakoulutuksen voidaan katsoa maassamme alkaneen 1809, kun Turkuun perustettiin kasvatusopillinen seminaari. Tavoitteena

yhteiskunnan kannalta oli hyödyn tavoittelu, alettiin uskoa, että hyvällä koulutuksella voidaan parantaa yhteiskunnallisia oloja. Seminaari toiminta kuitenkin loppui opiskelijapulaan 1820-luvulla.

Jyväskylään perustettiin vuonna 1863 ensimmäinen kansakoulunopettajia valmistava seminaari. Se oli eurooppalaisista esikuvistaan poikkeava sikäli, että sinne otettiin miehiä ja naisia. Tärkeimpiä pääsyvaatimuksia oli todistus moitteettomasta käytöksestä. Seminaarin ensimmäinen johtaja oli Uno Cygneaus. Seminaareja perustettiin myöhemmin lisää ja ne keskittyivät yksinomaan kansakoulunopettajien kouluttamiseen. Oppikoulujen opettajilta vaatimukset olivat 1900-luvun alussa koskivat lähinnä opetettavan aineen osaamista. Samoin oli tilanne ammatillisella puolella, sielläkin kiinnitettiin sodan jälkeiseen aikaan saakka huomiota ainoastaan opetettavan ammatin osaamiseen. Tällä hetkellä peruskoulun, lukion ja ammatillisen oppilaitoksen opettajalta vaaditaan 35 ov pedagogiset opinnot ja opetusharjoittelu.

3. Kasvatuksellisten- ja opetuksellisten tavoitteiden muutoksia

Sata vuotta sitten Suomi, kansakuntana haki omaa identtiteettiään ja myös koulut valjastettiin tähän tehtävään. Luokkaerot olivat suuret. Sivistyneistö oli hyvinkin tietoinen maailman muutoksista ja Suomen mahdollisuudesta itsenäistyä ja muodostaa oma kansakunta. Tavallisesta kansasta suurin osa osasi ainakin jollakin tavoin lukea ja kirjoittaa, koska näiden taitojen hallinta oli välttämätöntä ripille pääsemiseksi, ripillä käynti oli puolestaan ehto avioliiton solmimiselle. Kansan elintason niukkuus rajoitti sen mahdollisuuksia osallistua laajemmin yhteiskunnalliseen toimintaan ja itsensä sivistämiseen. Tavat olivat joissakin asioissa kehittymättömät ja joissakin asioissa hyvinkin hienotunteiset. Saatettiin sylkeä lattialle ja istua hattu päässä sisällä, mutta vieraampia ihmisiä teititeltiin samoin kuin lapset vanhempiaan ja ruokaan suhtauduttiin hyvin kunnioittavasti.

3.1. Tavoitteet tuhatyhdeksänsataa luvun alussa

Perusopetuksen tavoitteena 1800-luvun lopussa ja pitkälle 1900-luvulle oli ennen kaikkea kasvatuksellinen. Lukutaidon opettaminen nähtiin osana tätä tavoitetta. Tuohon aikaan oli suhteellisen helppo kontrolloida mitä kansalla annettiin luettavaksi. Kaiken lukemisen piti olla joko jollakin tapaa hyödyllistä tai sivistävää ja kansallista tietoisuutta lisäävää. Lisäksi kansakouluopetuksella piti pyrkiä vaikuttamaan myös oppilaiden vanhempien asenteeseen ja ajattelumaailmaan. Tapakasvatus joutui lähtemään joissakin asioissa hyvinkin alkeista. Esimerkkinä voidaan mainita kirjassa Täällä Pohjantähden Alla, kuvaus kylän ensimmäisestä kansakoulun opettajasta, joka joutui opettamaan

nenäliinan käyttöä. [Soininen 1901] "Johtakaamme ensiksi mieleen, että kasvatuksen päämääräksi on esitetty lujan siveellisen luonteen kehittäminen kasvatettavassa, ja sen tärkeimmäksi välikappaleeksi opetus, johon kuitenkin liittyy välttämättömänä valmistuksena hallinta ja yhtä välttämättömänä täydennyksenä ohjaus." Edellä olevassa lainauksessa on tiivistetty hyvin tuon ajan lapsille ja nuorille tarkoitetun koulun tavoitteet.

Sama kasvatuksellinen tavoite koski myös ammatillista opetusta. Kisälleille tarkoitettujen sunnuntaikoulujen tarpeellisuutta perusteltiin 1800-luvun lopulla, ammatin oppimisen lisäksi myös seuraavalla tavalla [Klemelä, 1999] "Moni nuorukainen joutuessaan käyttämään hyvin vapaa-aikansa siten tulee varjelluksi harhateiltä ja onnelliseksi kasvatetuksi kelpo kansalaiseksi, rehelliseksi ja taitavaksi työläiseksi sekä säädylliseksi ja uskolliseksi alamaiseksi."

Opetuksellisiakin tavoitteita oli kouluille asetettu tuohonkin aikaan. Kuopion tuomiokapituli määritteli vuonna 1892 kirkon alkeisopetukselle seuraavat tavoitteet. [Klemelä 1999] *Alemmalla osastolla* : tavuu, suoraluku, raamatunhistoria, katekismus ulkoa, helppojen sanojen kirjoittaminen ja virsilaulua. *Ylemmällä osastolla*: suoraluku, raamatunhistoria, katekismus kokonaisuudessaan, kauno- ja oikeinkirjoitusta ja virsilaulua. Tavoitteista suuri osa siirtyi sellaisenaan kansakoulun tavoitteisiin.

Opetussuunnitelmat olivat väljempää kuin nykyisin. Alla on opetussuunnitelma vuodelta 1901 [Soininen, 1901]. Nykyään tämä tämän tasoista suunnitelmaa sanotaan lukusuunnitelmaksi.

Opetussuunnitelma			
A Henkistä sivistystä tarkoittavat aineet			B Ruumin harjoitus
Reaaliaineet		Taideaineet	Muodolliset aineet
Ihmiselämää koskeva ryhmä	Luonnonelämää ja ihmisen suhdetta luontoon ryhmä		
Uskonto	Maantieto	Laulu	Voimistelu
Historia	Luonnontieto	Piirustus	
Kirjallisuus		Käsityöt	
			Kielenopetus lukeminen kirjoitus
			Matematiikka

Mielenkiintoisena yksityiskohtana voidaan mainita, että käsitöiden tavoite oli ensisijaisesti kasvattaa oppilaissa esteettisiä taitoja ja vasta toissijaisesti

tavoiteltiin käsitöillä hyötyä. Tämäkin kuvastaa merkittäväällä tavalla tuon ajan kasvatuksellisia ihanteita ja tavoitteita.

Tuohonkin aikaan opetusalan ihmiset pohtivat myös tavoitteiden yksilöllisyyttä ja niiden huomioimista opetuksessa. [Soininen, 1901] "Onhan sanottu kouluopetuksen suureksi varjopuoleksi juuri sitä, että se laiminlyöpi yksilölliset taipumukset, pakottaa etevät kyvyt tavalliseen tasaan ja estää niiden vapaata kehittämistä." Tämä ongelma pyrittiin ratkaisemaan, nykykielellä ilmaistuna, eriyttämällä, mutta tähän palataan myöhemmin.

3.2. Tavoitteet nykyään

Perusopetuksen tavoitteet määrittelee eduskunta yleisellä tasolla. Opetushallitus laatii valtakunnan tason opetussuunnitelma, jonka perusteella koulut tekevät omat tarkat opetussuunnitelmansa.

Yleisenä asiana voidaan todeta, että koulu on muuttunut teoriapainotteiseksi ja siellä korostetaan tiedon määrää kasvatuksellisten tavoitteiden kustannuksella. Tähän varmaankin monia syitä. Ei ole eikä voida olettaa olevankaan niin yhtenäistä arvopohjaa, jolle kasvatuksen perustaa, kuin sata vuotta sitten. Yhteiskunta muuttuu yhä enemmän sellaiseksi, että siinä selviytyminen vaatii yksilöiltä parempia tiedollisia valmiuksia. Koululaitos pyrkii vastaamaan tähän tarpeeseen, antamalla opiskelijoille mahdollisimman hyvät ja monipuoliset taidot.

Nykyinen yhteiskunta arvostaa tiedollista osaamista huomattavasti enemmän kuin taidollista osaamista, vaikka juhlapuheissa toisin väitetään. Osaltaan tämä johtuu ehkä siitä, että koulutuksen sisällöstä ja tavoitteista päättävät suurimmalta osalta ihmiset joilla ei ole syvällistä käsitystä johonkin ammattiin kasvamisesta ja sen vaatimuksista. Ammatilliset tavoitteetkin pyritään jakamaan keinotekoisesti kursseihin, kuten teoreettisetkin aineet. Talon rakentaminen ei ole erillisiä kursseja vaikka niiden sisältö kokonaisuutena kattaisikin kaikki talonrakentamiseen liittyvät asiat.

Peruskoulun virallisten tavoitteiden sisään on kirjoitettu ajatus, että kaikki menevät lopulta yliopistoon. Tämä ääneen lausumaton tavoite on hallinnut peruskoulun opetussuunnitelman muotoutumista ja sen takia siitä on jäänyt vähälle taito- ja taideaineet. Yksi todiste tästä lukioon menon, piilo-opetussuunnitelmasta on luettavissa Helsingin Sanomista 14.12.2003 artikkelista, jossa käsitellään poikien huonoa koulumenestystä peruskoulun viimeisillä luokilla.

Toinen asia, joka heikentää taide- ja taitoaineiden asemaa on yhteiskunnan muuttuminen kaikissa suhteissa taloudellista hyötyä tavoittelevaksi. Oppilaista

pyritään kasvattamaan vain tuottajia ja kuluttajia ei ihmisiä. Tätäkään tavoitetta ei lausuta julki, mutta se on piilotavoitteena koululaitoksen sisällä.

Virallisissa opetussuunnitelmissa puhutaan kodin ja koulun yhteistyöstä ehkä enemmän kuin sata vuotta sitten. Käytännössä tämä on hankalampaa kuin aikaisemmin. Sata vuotta sitten Suomessa oli huomattavasti enemmän yhtenäiskulttuuri kuin nykyään. Erot kotien ja yhteiskunnan tavoitteiden välillä olivat pienet. Tällöin oppilaalle tarjottiin samaa käyttäytymismallia kodista ja koulusta.

Kokonaisuutena voisi sanoa, että sata vuotta sitten kasvatuksen tavoitteet oli määritelty tarkasti ja määrätietoisesti, millaisia ihmisiä koulusta tulee ulos ja millaiset ihanteet, ajatusmallit ja tavat heillä on. Nykyään on määritelty tarkasti tiedolliset tavoitteet, millaisen tietovaraston kanssa oppilaat päättävät koulunsa.

4. Oppimiskäsityksen muuttuminen

Tässä osassa käydään lyhyesti läpi, mitä erilaisia oppimiskäsityksiä on sadan vuoden aikana ollut. Tämän ei ole tarkoituskaan olla tyhjentävä kuvaus asiasta. Oppimiskäsityksistä eniten pyritään esittelemään 1900-luvun alun käsityksiä ja tämän päivän käsityksiä.

Herbartilaisuus. Saksalainen professori Johan Friedrich Herbart (1776 - 1841) on ensimmäisiä, joka pyrki kehittämään käsitystä oppimisesta ja opettamisesta. Mikael Soininen omaksui Herbartin ajatukset ja toi ne suomalaiseen koululaitokseen.

Herbartin ajatukset perustuivat kolmelle keinolle: hallinta, kuri ja opetus. *Hallinnalla* hän tarkoitti koulun yleistä kuria ja järjestystä. *Kuri* tarkoitti opettajan persoonallisuuden merkitystä luonteenkehitykselle. Tämä kohta asetti suuria vaatimuksia opettajan persoonalle ja käytökselle. Varsinaisen *opetuksen* tuli olla kasvattavaa ja monipuoliseen harrastuneisuuteen ja siveellisyyteen pyrkivää.

Soininen muokkasi, kuten oppi-isänsäkin opetustapahtuman kulkuun liittyviä käsitteitä ja tapahtumia. Soinisen mukaan [Iisalo, 1991] opetuksessa tuli olla kolme vaihetta.

1. Havainnollisuuden aste
Opettajan tuli aluksi kiinnittää uusi asia oppilaiden kokemusmaailmaan
2. Käsitteellisyden aste
Uusi pyrittiin saamaan kokonaisuudeksi ja hallittuun muotoon
3. Harjoituksen aste
Uutta tietoa pyrittiin soveltamaan käytännössä

Herbartilaisuudessa opiskelija käsitettiin kuitenkin aika passiivisena, lähinnä astiana, johon opettaja kaatoi tietoa tai muokkasi oppilasta kuin saventalajaa muokkaa savea. Oppilaan omaa roolia ei juurikaan pohdittu.

Behaviorismi. Tässäkin mallissa opiskelija käsitetään melkoisen passiiviseksi. Oppimisen tuloksia mitataan ulkoisen käyttäytymisen perusteella. Oppimistilanne suunnitellaan hyvin ja on tarkasti määritelty loppukäyttäytyminen johon pyritään. Jos loppukäyttäytyminen on ennalta toivottu katsotaan, että oppimista on tapahtunut. Oppilaan omaa roolia pohdittiin kuitenkin selvästi enemmän kuin aikaisemmissa malleissa. Mallissa ei pohdita onko opittava asia sisäistetty. Mallia voisi lyhyesti kuvailla keppi ja porkkana.

Humanistinen ja kriittinen käsitys. Näissä malleissa oppilaan oma osuus on tärkeä. Oppilaan oletetaan olevan kriittinen kaikkien asioiden suhteen ja pohtivan kaikkea uutta opittavaa asiaa. Oppilas ei saa hyväksyä mitään asiaa ylhäältä annettuna, vaan hänen täytyy myös ymmärtämisen kautta hyväksyä oppimansa. Tämä hyväksyminen ja ymmärtäminen koskee erityisesti sellaisia käsitteitä kuin: arvot, uskomukset, normit ja erilaiset oletukset.

Kognitiivinen malli. Tämä perustuu jossakin määrin havainto psykologiaan. Mallin keskiössä on oppilas. Pyritään selvittämään miten hän havaitsee, luokittelee tietoa, käsittelee tietoa, oppii ja muistaa. Näiden tietojen perusteella pyritään sitten rakentamaan oppimistilanne ja ympäristö sellaiseksi, että se palvelisi mahdollisimman hyvin kunkin oppilaan tarpeita ja taipumuksia. Tämän mallin käytöllä voidaan välttää edellä mainittua ongelmaa joka, "pakoittaa etevät kyvyt tavalliseen tasaan."

Konstruktivinen malli. Tämä malli laajentaa kognitiivista mallia, että siinä pyritään selvittämään miten tieto rakentuu oppilaan mielessä. Kuinka hän muokkaa ja prosessoi oppimaansa ja yhdistelee uutta asiaa aikaisemmin oppimaansa. Vuorovaikutus ympäristöön ja opettajaan jää vähemmälle huomiolle. Tästä puutteesta huolimatta malli soveltuu ehkä parhaiten käytettäväksi, kun suunnitellaan verkko-opetusta. Verkko-opetuksessahan oppilas on aika paljon yksin ja joutuu rakentamaan opittavan asian mieleensä aika paljon omin avuin.

4.1. Mallien käyttö eri aikoina

Herbartilaisuus oli tietenkin vallitseva tai voidaan sanoa lähes ainoa tapa opettaa 1900-luvun alkupuolelle. Tämä johtui osaltaan siitä, että Soinin kirjat ja opit olivat merkittävässä asemassa tuohon aikaan. Muutos alkoi tapahtua 1920-luvulla [Iisalo, 1991] ns. reformipedagogiikan myötä. Silloin alettiin

korostaa oppilaan huomioimista opetustilanteessa. Samalla pyrittiin painottamaan, että oppiminen ei rajoitu pelkästään kouluun.

Toinen uudistus, mikä tuli reformipedagogiikan myötä oli menetelmä, jota sanotaan nykyään ongelmakeskeiseksi oppimiseksi. Siinä oppilaille annetaan tehtävä, josta suoriutuminen vaatii monenlaisia tietoja ja taitoja. Tarkoitus on, että oppilasryhmä selviytyy tehtävästä mahdollisimman itsenäisesti, opettaja johdattaa oppilaat vain tiedon lähteille.

Behaviorismi oli vallitseva käsitys 1900-luvun puolen välin seutuvilla. Malli oli selkeä, koska loppukäyttäytyminen oli oppimisen mitta, sillä ei ollut kovinkaan suurta merkitystä miten tieto meni oppilaan tietoisuuteen ja miten hän sitä järjesteli mielessään. Yksilön omaa toimintaa ja yksilöllisyyttä alettiin korostaa voimakkaasti 1970-luvun alusta alkaen jolloin sijaa saivat ns. kehittyneemmät oppimismallit.

Tällä hetkellä on vaikea erottaa jotakin selkeää yksittäistä mallia, joka olisi todettu selkeästi muita paremmaksi. Opettajien mielessä ehkä sekoittuu yhdeksi kokonaisuudeksi Humanistinen ja kriittinen käsitys, Kognitiivinen malli ja Konstruktiivinen malli yhdeksi kokonaisuudeksi, jonka mukaan he suunnittelevat opetustaan ja oppimistilanteita. Behaviorismi on kuitenkin jollakin tapaa taustalla, koska se antaa selkeän palautteen oppimistuloksista sekä oppilaalle, että opettajalle.

5. Opetusmenetelmien vertailu

5.1. Mikä on opetusmenetelmä

Käsitettä opetusmenetelmä voidaan lähestyä monesta eri näkökulmasta. Puhutaan luokkaopetuksesta/työpajaopetuksesta tällöin määrävänä tekijänä on paikka missä opetus tapahtuu. Jakoperusteena voi olla etäisyys oppilaan ja opettajan välillä, lähiopetus/etäopetus. Määritelmä perusteena voi olla väline, puhutaan liitu ja taulu/tietokoneavusteisesta opetuksesta. Opettavan ryhmän koko voi olla myös peruste, yksilöopetus/ryhmäopetus. Jakoperusteena voidaan käyttää myös opettajan roolia, itsenäinen /opettajajohtoinen opiskelu.

Tässä yhteydessä ei niinkään kiinnitetä huomiota johonkin jakoperusteeseen. Mukaan on otettu toimintoja joissa käytetään nykyään apuna tietotekniikkaa. Sitten on verrattu miten opetusta on hoidettu noin sata vuotta sitten ja miten tietotekniikka on muuttanut tai voi muuttaa vastaavaa toimintoa. Samalla pyritään arvioimaan miten asioita on ilmaistu aikaisemmin ja miten nykyään.

Perinteisessä luokkaopetuksessa käytetään nykyään aika paljon apuna tietotekniikkaa, mutta en sitä pidä tässä yhteydessä varsinaisena opetusmenetelmän muutoksena. Mielestäni se ei ole mitään uutta, jos opettaja

on aikaisemmin näyttänyt pahvitaululta leijonan kuvaa ja nyt hän tekee saman videotykillä Power-Point esityksenä.

5.2. Vuorovaikutteinen opetus

Käsitteenä yleisempään tietoisuuteen vuorovaikutteinen opetus on tullut vasta tietokoneiden myötä. Kuitenkin kaikki ihmisten välinen kanssakäyminen on vuorovaikutusta tavalla tai toisella. Verkko-opetuksen yhteydessä on tullut esille myös käsite interaktiivisuus. Tällekin käsitteelle on monia määritelmiä. Hyvä määritelmä on mielestäni seuraava [Lehtinen, 1997] Todellista *interaktiota* on, kun henkilö kommentoi viestiä, joka on kommentti hänen alkuperäiseen viestiinsä. Ts. vuorovaikutusta tapahtuu vasta sitten, kun kommenttia kommentoidaan.

Vastaavaa asiaa pohti [Soininen, 1906] määritellessään käsitettä vuoropuheinen opetusmuoto-kysymys. "Keskustelun sielutieteellinen ja metodinenmerkitys on se, että toiselta puolen kysely vaikuttaa liikkeellepanevana sysäyksenä oppilaan ajatustoimintaan, ja toiselta puolen vastausten antaminen vaatii häntä tekemään selkoa tämän ajatustyön suorituksesta ja tuloksista." Lisäksi Soininen ohjeistaa opettajia toimimaan niin, että kysymyksiin ei vastata kyllä tai ei vaan vaativat aina opettajan kommenttia oppilaan vastauksen jälkeen.

Näin ajateltuna vuorovaikutteinen tai interaktiivinen opetus ei ole mikään uusi asia. Sitä on tehty aina kouluissa enemmän tai vähemmän tietoisesti. Tietotekniikka on tuonut siihen kuitenkin uusia ulottuvuuksia ja sen takia pedagogit ovat siitä kiinnostuneet. Suurin etu on ehkä se, että verkko keskustelut eivät häviä samalla tavoin kuin puhe. Opettaja voi johdatella verkkokeskustelua ja saada sen lopputuloksena syntymään, ainakin periaatteessa, vaikka hyvää oppimateriaalia joka on kaikkien käytössä. Etuna on myös välimatkojen aiheuttamien esteiden häviäminen. Suurimpana haittana näen verkko-opetuksessa varsinkin, jos sen määrä on suuri, sanattoman viestinnän puutteen. Sanaton viestintä voi olla joissakin tapauksissa merkittävämpää kuin puhuttu sana.

5.3. Eriyttäminen

Normaalissa perusopetusryhmässä on yleensä 20 - 35 oppilasta. Ryhmän sisällä on normaali lahjakkuusjakautuma, mutta opetus pitäisi hoitaa keskitason mukaan, jolloin lahjakkuuden molemmat ääripäät jäävät vaille huomiota. Ongelmaa on pohdittu ja pyritty ratkaisemaan jo sata vuotta sitten, kuten kappaleessa 3.1. todetaan. Pääasiallinen ohje oli tuolloin, kuten nykyäänkin, lisätehtävien antaminen. Muita tavoitteita lisätehtävillä tuolloin oli pitää

oppilaat vapaa-aikanaanakin hyvien harrastusten parissa ja taata työrauha luokassa.

Tekniset keinot olivat siihen aikaan varsin rajalliset. Mahdollista oli antaa lisää luettavaa tai tehtäväksi tutkia omatoimisesti ympäristöä mm. luontoa. Saattaa sitten luettu ja/tai luonnontutkimus lyhyeen kirjalliseen muotoon, jota opettaja voi sitten käyttää vaikka lisäopetusmateriaalina muille oppilaille.

Eriyttämiseen tietotekniikka tarjoaa todella hyvät mahdollisuudet. Ainoat edellytykset on, että jokaisella oppilaalla käytössään tietokone ja sen riittävät käyttötaidot esillä olevaa opiskelua varten. Käyttötaitoihin on ennen tämän menetelmän käyttöä kiinnitettävä erityistä huomiota, ettei välineestä tule pääasiaa opittavan sisällön kustannuksella.

Opettaja voi tehdä koneelle hyvinkin eritasoisia tehtäviä niin, että kaikki oppilaat joutuvat työskentelemään osaamisensa ylärajoilla. Tällöin hitaammatkin saavat onnistumisen elämyksiä ja lahjakkaat eivät pitkästy. Kone on väsymätön tehtävien tarkastaja ja oppilas voi harjoitella yksilöllisesti niin kauan kuin on tarpeellista. Nopeasti eteneville on verkon kautta tie auki hyvin monille tiedon lähteille, jolloin he voivat edetä kykyjensä mukaan.

Eriyttäminen on opetuksen osa-alue, jolle tietotekniikka soveltuu erittäin hyvin. Tällä hetkellä ei vain tätä mahdollisuutta vielä hyödynnetä läheskään kaikessa siinä laajuudessa missä olisi mahdollista. Nykyiset kuvankäsittelyn mahdollisuudet antavat erittäin hyvät mahdollisuudet monien asioiden havainnollistamiseen, mikä auttaa erityisesti hitaasti oppivia oppilaita. Suurimpana esteenä on ehkä sopien verkkomateriaalien puute.

5.4. Ongelmalähtöinen opetus

Reformipedagogit kehittivät 1900-luvun alkupuolella ns. projektiopetusta. Perusajatuksena oli, että kaikki opetus muutetaan erilaisiksi projekteiksi, joissa oppilaat toimivat ryhmässä. [Iisalo, 1991]. Projektina voisi olla vaikka yliopiston synty Tampereelle. Sen yhteydessä opiskeltaisi taloustietoa, historiaa, maantiedettä jne., kaikkia niitä asioita, jotka ovat vaikuttaneet yliopiston syntyyn Tampereelle. Opettajan osuus on toimia lähinnä matkaoppaana tiedon lähteille ja arvioida kerättyjen tietojen oikeellisuutta.

Ongelma voidaan antaa myös käytännöllisestä asiasta esim: rakentakaa pieni sähkömoottori. Tällöin oppilaat joutuvat selvittämään sähkömoottorin toimintaperiaatteen, raaka-aineet, valmistustavan ja -menetelmän, jne... ja lopuksi valmistamaan kyseisen moottorin.

Nykyään tätä menetelmää kutsutaan ongelmalähtöiseksi oppimiseksi ja sitä perustellaan sillä, että kehittää oppilaiden omatoimisuutta ja pakottaa heitä ottamaan vastuuta omasta oppimisestaan. Ongelmalähtöistä menetelmää

suositellaan käytettäväksi varsinkin niiden opiskelijoiden kanssa joilla on motivaatio ongelmia normaalissa opetuksessa.

Tähän opetusmenetelmään tietotekniikka soveltuu erittäin hyvin. Opettaja voi valmistellessaan tehtävää huomioida erittäin hyvin kunkin ryhmän valmiudet, kuinka paljon esitietoja hän antaa varsinaisen tehtävän lisäksi. Ennen työn aloitusta ryhmän on jollakin tapaa organisoiduttava. Ryhmän kootessa sitten tietoja vaikka jollekin oppimisalustalle ovat kaikkien tuotokset heti kaikkien nähtävillä. Tämä tietojen nopea esiintulo auttaa käsitteen muodostusta kaikkien mukana olijoiden osalta, kaikille muodostuu heti alusta alkaen mieleen oikeanlainen malli käsiteltävästä asiasta. Toinen etu, jonka tietotekniikka tarjoaa, on tietojen tekninen käsittely. Kun tieto on löytynyt verkosta, se voidaan siirtää ryhmän omaan käyttöön suhteellisen helposti. Lisäksi ryhmän kokoamaa tietokokonaisuutta helppo muokata.

5.5. Oppiaineiden integrointi

Koulujen opetussuunnitelmat koostuvat erillisistä oppiaineista, vaikka maailmassa toimitaan kokonaisuudessa. Opiskeltavan tiedon määrää on kaikkina aikoina pidetty liian suurena. Asia sisällön määrä ei saisi ylittää oppilaan kykyä oppia, [Soininen, 1901] "Oppiainesta ei saa olla enempää eikä vähempää kuin minkä ehtii hyvin ehtii läpikäydä". Lisäksi hän pohtii millaisen jäljen oppiaines oppilaaseen jättää, onko se vain ulkoaopittua, vai onko siitä ehditty käydä periaatteellista keskustelua. [Soininen, 1901] piti tärkeänä mm. lukukirjan sisältöä, kuinka hyvin se palveli muiden aineiden opetusta. " Vielä tärkeämpi on kuitenkin lukukirjassa sisällyksen likeinen yhteys muun opetuksen ja varsinkin realiaineiden opetuksen kanssa. Sillä tämän yhteyden kautta joutuvat lukukirjan lukukappaleet jäseniksi vielä paljon suurempiin ajatuskokonaisuuksiin." Soininen piti oppiaineiden rinnastusta niin tärkeänä, että hän kirjoitti siitä oppikirjaansa kokonaisen luvun.

Tähän ongelmaan tarjotaan tuolloin yhtenä ratkaisuna opetusaineiden rinnastusperiaatetta, nykyään sanottaisi oppiaineiden integrointia. [Soininen, 1901] toteaa " Muutoin ovat jääneet käyttämättä ja huomaamattakin kaikki rinnastuksen parhaimmat apuneuvot opetus- ja oppimistyön helpottamiseksi sekä hyvien tulosten saavuttamiseksi."

Soinisen ajan koulussa oppiaineiden rinnastuksen toteuttaminen oli kokonaan opettajan innostuksen ja osaamisen varassa. Reformipedagogit kehittivät asiaa pidemmälle. He ehdottivat [Iisalo, 1991], että poistettaisi perinteiset oppiainerajat tai ainakin niitä madellettaisi. Heidän mielestään näin tulisi menetellä varsinkin alaluokkien opetuksessa. Ammatillisessa opetuksessa

on aina tapahtunut jossakin määrin erioppiaineiden integrointia johtuen siitä, että ammattiaineiden opettajat ovat opettaneet mm. matematiikkaa ja fysiikkaa.

Nykyään myönnetään oppiaineiden integroinnin hyöty, mutta suurimpana esteenä siinä on ehkä opettajien asenne. Meillä valmistuu erikseen, matematiikan-, maantiedon-, kielten-, liikunnan opettajia. Syntyy varsin raadollinen tilanne, kun kukaan ei halua luopua eduistaan, vaan pitää oman oppiaineensa puolta.

Tietotekniikan avulla olisi mahdollista toteuttaa integrointia ehkä suhteellisen helpostikin. Esimerkkinä matematiikka ja maantiede. Osa maantiedon kurssista tehtäisi verkossa. Kurssiin otettaisi mukaan geometriaa ja trigonometriaa. Nämä osiot sitten opettaisi matematiikan opettaja, mutta laskujen sisältö tulisi maantiedosta. Verkossa kumpikin opettaja olisi koko ajan tietoinen toistensa ja oppilaiden tekemisistä. Samalla tavoin voitaisi yhdistää myös monia muita oppiaineita.

Tälle integroinnin alueelle tullaan todennäköisesti tulevaisuudessa tuottamaan oppimateriaalia, kun opetussuunnitelmien perusteet tuovat tällaiselle materiaalille käyttöä. On syytä muistaa, että oppiaineiden rajat ovat osittain keinotekoisia ja näin ollen muutettavissa tai poistettavissa.

Internet on sellaisenaan hyvin integroitunut eri oppiaineiden suhteen. Elävässä elämässä yksilö joutuu sieltä hyvinkin hajanaisista tiedon sirpaleista kokoamaan kuhunkin tilanteeseen tarvitsemansa tiedon kokonaisuuden.

5.6. Ryhmätyöt

Ryhmätyötä sivuttiin, kun käsiteltiin ongelma-aperusteista oppimista. Siinä menetelmässä on kuitenkin etusijalla ongelman ratkaisu ja toisella sijalla ryhmän toiminta. Opetusmenetelmänä ryhmätyössä on tavoitteena ryhmänä oppia, kulloinkin esillä oleva asia ja ennen kaikkea ryhmässä toimimista. Varsinkin lapsiryhmässä on opettajan tehtävä huolehtia ryhmän toiminnasta.

Sata vuotta sitten ryhmätyön käyttöä ei juurikaan opetettu opettajakoulutuksessa, mutta sen käyttö yleistyi reformipedagogiikan myötä. Matti Koskeniemen Kansakoulun Opetusopissa vuodelta 1944 ryhmätyö saa suuren painon. Kirjassa annetaan yksityiskohtaisia ohjeita ryhmän koosta ja muodosta ja millaisilla perusteilla ryhmät tulisi muodostaa. Muutoin ryhmätyön kulku esitellään, kuten ryhmätyöt toimivat tälläkin hetkellä missä hyvänsä seminaarissa. Kuitenkin ehkä tuotostakin tärkeämpänä kirjassa pidetään ryhmän sosiaalista toimintaa.

Ryhmätyöt voidaan ainakin osittain organisoida uudella tavalla tietotekniikan avulla. Normaalisti ryhmässä joku valitaan sihteeriksi ja yrittää

sitten saada tuotoksen paperille tai kalvolle, joka sitten esitellään palautuksen yhteydessä Lapsijoukossa (monasti aikuisillakin) äänekkäimmän mielipide tulee helposti ryhmän mielipiteeksi, jolloin hyvinkin arvokkaat ajatukset voivat jäädä vaille huomiota.

Verkossa jokainen voi kertoa oman mielipiteensä ja tietonsa esillä olevasta asiasta ja se jää kaikkien käyttöön. Tästä materiaalista saadaan selkeämpi mielikuva ryhmän mielipiteestä. Kirjoitettuna tekstinä kaikkien mielipiteet ovat paremmin samanarvoisia kuin huutoäänestyksessä. Myös sanallisesti hiljaisten ja arkojen mielipiteet tulevat esille. Opettaja voi varovasti ohjata tässä vaiheessa ryhmän mielipiteen muodostumista, kun kaikkien mielipiteet ovat ruudussa näkyvissä.

Viiteluettelo

- [Bould & Feletti 1999] David Boud & Grahame I. Feletti, *Ongelmalähtöinen Oppiminen Uusi Tapa Oppia*, Hakapaino Helsinki 1999.
- [Engeström 1994] Yrjö Engeström, *Perustietoa Opetuksesta*, Painatuskeskus, Helsinki 1994.
- [Hein & Larna 1992] Irene Hein & Riitta Larna, *Lähelle, Kaukana, Yksin, Yhdessä*, Hakapaino 1992.
- [Iisalo 1991] Taimo Iisalo, *Kouluopetuksen Vaiheita*, Otava, Keuruu 1991.
- [Jansson 1927] Uno Jansson, *Kansakoulun Laskuopin Opetusoppi*, Gummerus, Jyväskylä 1927.
- [Klemelä 1999] Kirsi Klemelä, *Ammattikunnista Ammatillisiin Oppilaitoksiin*. Painosalama, Turku 1999.
- [Lehtinen 1997] Erno Lehtinen, *Verkkopedagogiikka*, Edita, Helsinki 1997.
- [Lonka & Lonka 1991] Kirsti Lonka ja Irma Lonka, *Aktivoiva Opetus*, Kirjayhtymä, Helsinki 1991.
- [Kivinen 1988] Osmo Kivinen, *Koulutuksen Järjestelmän Kehitys*, Turun Yliopisto, Turku 1988.
- [Meisalo & Sutinen & Tarhio 2003] Veijo Meisalo, Erkki Sutinen ja Jorma Tarhio, *Modernit oppimisympäristöt*, Tietosanoma, Helsinki 2003
- [Miettinen 1990] Reijo Miettinen, *Koulun Muuttamisen Mahdollisuudesta*, Painokaari, Helsinki 1990.
- [Nummenmaa & Virtanen 2002], Anna Raija Nummenmaa & Jorma Virtanen, *Ongelmasta Oivallukseen*, Yliopistopaino, Tampere 2002.
- [Paakkola 1991] Esko Paakkola, *Johdatus Monimuoto Opetukseen*, Valtion painatuskeskus, Helsinki 1992.

- [Sahlberg & Leppilampi 1994] Pasi Sahlberg ja Asko Leppilampi, *Yksinään vai Yhteisvoimin*, Yliopistopaino 1994
- [Soininen 1901] Mikael Soininen, *Opetusoppi*, Päivälehdin Kirjapaino, Helsinki 1901.
- [Soininen 1895] Mikael Soininen, *Kasvatusopillisia Luennoita*, Otava, Helsinki 1895.
- [Susitaival 1950] Paavo Susitaival, *Suomen Puusepänteollisuuden Vaiheita*, Esan Kirjapaino, Lahti 1950.
- [Vahtola 2003] Jouko Vahtola, *Suomen Historia*, Otava, Keuruu 2003.
- [Koskenniemi 1946] Matti Koskenniemi, *Kansakoulun opetusoppi*, Otava, Helsinki 1946.

Tietojärjestelmät perusterveydenhuollossa

Suvi Vuorela

Tiivistelmä.

Sosiaali- ja terveydenhuollon ympäristössä on käytetty tietojärjestelmiä ja teknologiaa hyväksi jo monien vuosikymmenien ajan. Viime vuosiin asti alalla on ollut tyypillistä monet erilliset järjestelmät, joiden yhteistoiminta ei ole ollut mahdollista. Yhteiskunnassa ja alalla esiintyneet ongelmat ovat pakottaneet kehittämään toimintamalleja. Ratkaisuna nähdään yhtenäinen tietojärjestelmä, joka tukisi mm. saumatonta palveluketjua ja yhtenäistä toimintaa. Uusien järjestelmien toiminnasta ei ole vielä oikeastaan tutkimustietoa, mutta joitakin etuja ja ongelmia voidaan jo hahmottaa. Tutkimukseni tarkoitus on olla yleiskatsaus perusterveydenhuollon tietojärjestelmistä ennen, nyt ja tulevaisuudessa. Tuon myös hiukan esille taustaa alasta, alan toiminnasta ja vaatimuksia ja rajoitteita uudelle järjestelmätyypille.

Avainsanat: Perusterveydenhuolto, tietojärjestelmä, saumaton palveluketju

1. Johdanto

Sosiaali- ja terveydenhuollon toimintaympäristö on ollut viimeisen kymmenen vuoden aikana kovien muutosten alla. Yhtenä muutoksen syynä on 1990-luvulla Suomessa ollut lama, joka pakotti alan suuriin säästötoimiin. Valtionosuuksia leikattiin, henkilökuntaa vähennettiin ja samalla toimintaa yritettiin tehostaa. Vaikka pahin lama maassa on jo ohi, niin edelleenkin terveydenhuollon ympäristö elää jonkinasteisessa lamassa.

Tietojärjestelmiä ja teknologiaa on käytetty hyväksi alalla jo monien vuosikymmenien ajan. Käyttö ei kuitenkaan ole ollut kovin tehokasta. Ongelmana ovat olleet monet erilliset järjestelmät, joiden yhteensovittaminen on ollut mahdotonta. Nyt Suomessa yritetään lopettaa jatkuva uusien järjestelmien kehittäminen, ja kehittää kaikkia palveleva yhdenmukainen järjestelmä, joka tukisi yhteistoimintaa. Esimerkkinä uudesta yhdenmukaisesta järjestelmäkokonaisuudesta on Novo Group Oyj:n kehittämä Pegasos-nimisen sosiaali- ja terveystietojärjestelmän, joka on jo otettu useissa kaupungeissa käyttöön. Järjestelmän tarkoitus on palvella kaikkia hoitotyötä tekeviä vähentämällä rutiinitöitä ja samalla nopeuttaa tietojen saatavuutta. Potilas hyötyy uudesta järjestelmästä siten, että hänen palveleminen helpottuu ja

nopeutuu, kun hänen tiedot ovat aina henkilökunnan käytössä potilaan hakeutuessa hoitoon.

Hyvinvointipalveluiden toimivuudella on suuri merkitys Suomessa. Aihetta on tutkittu paljon viimeisen kymmenen vuoden aikana eri näkökulmista. Tällä hetkellä ollaan menossa alalla kohti yhtenäisiä tietojärjestelmiä, jotka mahdollistaisivat mm. saumattoman palveluketjun. Tutkimustuloksia yhtenäisten järjestelmien käytöstä ei vielä oikeastaan ole. Tätä tutkimusta varten olen tutkinut kirjallisuutta, jonka perusteella olen kirjoittanut tekstin koskien perusterveydenhuollon tietojärjestelmiä; taustaa toimintaympäristöstä, miten asiat ovat ennen olleet ja mitä uudelta järjestelmältä on odotettavissa.

2. Sosiaali- ja terveydenhuollon toimintaympäristö

Sosiaali- ja terveydenhuollon toimintaympäristön tehtävät ovat moninaiset. Ympäristön vaikutus ihmisen elämään on loppujen lopuksi suurempi kuin moni osaa kuvitellakaan. Toimiala jakautuu useaan sektoriin, joiden toimintaa ohjaa valtio ja joiden rahoitus tulee etupäässä verovaroista. Palveluiden toteutuksesta vastaa kunnat joko yksin tai yhteistyössä toisten kuntien kanssa. [STM, 1998:b] Tässä luvussa tuon esille toimialaan vaikuttavia tekijöitä, tulevaisuuden näkymiä ja lähtökohtia toimintaympäristön kehittämiseksi. Aluksi kuitenkin kerron hiukan perusterveydenhuollosta Suomessa.

2.1. Perusterveydenhuolto Suomessa

Tampereen kaupungin Perusterveydenhuollon strategia vuoteen 2012 (2002) tuo kattavasti esille tietoa perusterveydenhuollosta. Suomessa on kaikilla samanlaiset oikeudet terveydenhuoltoon sosiaaliryhmästä, tuloista ja asuinpaikasta riippumatta. Julkisten sosiaali- ja terveydenhuollon tarjoamien palvelujen lisäksi Suomessa palveluita tarjoavat kunnalliset osakeyhtiöt, liikelaitokset ja yksityissektori, kolmas sektori mukaan lukien. Suomessa perusterveydenhuolto muodostaa kuitenkin pohjan, johon muu terveydenhuolto rakentuu. Perusterveydenhuollon tehtävä on antaa ihmisille ensisijainen hoito ja ohjata tarvittaessa erikoissairaanhoidon. Sen tavoitteena on väestön terveydentilan parantaminen ja toimintaa leimaa koko väestöön ja riskiryhmiin kohdistuva ennaltaehkäisevä työ. Toivottavaa olisi, että perusterveydenhuolto voisi seurata potilaan kulkua hoitoyksiköissä ja mahdollisesti puuttua siihen tarpeen vaatiessa.

Perusterveydenhuollon merkittävydestä kertoo myös käyntien määrä, joita sosiaali- ja terveystieteiden tutkimuksen (2002:b) mukaan vuonna 2002 oli 25,13 miljoonaa. On siis ymmärrettävää, että asioiden on sujuttava tällä sektorilla. Kertomuksen mukaan terveydenhuollon palvelua on yritetty parantaa

omalääkäri- ja omahoitajärjestelmällä, jonka piirissä oli vuoden 2000 lopussa 66 % väestöstä. Järjestelmän tarkoitus on mahdollistaa parempaa palveluiden saatavuutta ja toimivuutta ja samalla antaa mahdollisuuden työntekijöiden kehittää omaa työtänsä. Valitettavaa kuitenkin on huomata, että ihmiset eivät ole tyytyväisiä toimintaan, sillä ainakin Tampereella on omalääkäreille niin pitkät jonot, etteivät ihmiset pääse sinne tarpeen tullen. Järjestelmän tavoiteltuja etuja ei siis vielä ole taidettu saavuttaa.

2.2. Toimintaympäristöön vaikuttavia tekijöitä

Sosiaali- ja terveysministeriön työryhmämuistio (1998:b) käsittelee toimintaympäristön merkittäviä muutoksia viime vuosien ajalta. Muutokset ovat johtuneet taloudellisen laman lisäksi erilaisten hallintoelinten päätösvallan muutoksista. Myös väestörakenteen muutokset, teknologian ja tietojärjestelmien kehittyminen sekä kansalaisten sosiaalisten ja muiden perusoikeuksien merkityksen korostuminen perusoikeusuudistuksen myötä ovat vaikuttaneet toimintaympäristön muutoksiin.

Valtionosuuksien leikkaaminen taloudellisen laman myötä johti 1990 -luvulla voimakkaisiin säästö- ja toiminnan tehostamispaineisiin sosiaali- ja terveydenhuollossa. Muun muassa henkilöstön määrään alalla supistettiin tuntuvasti 1990-luvulla, joskin lamavuosien jälkeen määrä on jälleen kääntynyt hiukan nousuun. Ongelmana pitää kuitenkin nähdä vielä tälläkin hetkellä alalla vallitsevat henkilökunnan määräämääräiset työsuhteet. Jatkuvat sijaisuudet eivät voi olla vaikuttamatta palvelun laatuun. [STM, 1998:b]

Suomessa on edelleen korkea työttömyysaste, mikä vaikuttaa heikentävästi verotuloihin ja tätä kautta sosiaali- ja terveydenhuollon rahoitukseen [STM, 1998:b]. Pitkäaikaiset työttömyysjaksot vaikuttavat myös ihmisten henkiseen ja fyysiseen hyvinvointiin. Erityisesti sosiaalipalveluiden, mutta myös terveyspalveluiden tarve kasvaa koko ajan ihmisten arkipäivän ongelmien myötä. Toisilla on liikaa töitä, joka aiheuttaa ongelmia kun taas joillakin ei ole työtä ollenkaan ja tämä aiheuttaa ongelmia. Työllisyystilanne vaikuttaa siis monta kautta toimintaympäristöön.

2.3. Tulevaisuudennäkymät

Väestöennusteen mukaan Suomen väkiluku kasvaa nykyisestä 5,1 miljoonasta asukkaasta 2,1 % vuoteen 2010 mennessä. Ongelmana ei ole kuitenkaan väestön lisääntyminen vaan ikärakenteen muuttuminen. Yli 75-vuotiaiden osuus väestöstä tulee kasvamaan lähes kolmanneksella samalla kun alle 15-vuotiaiden määrä vähenee noin 10 %. Väestön vanheneminen lisää palvelujen kysyntää ja hoitokustannuksia, sillä vanhimpien ikäluokkien hoitokustannusten on laskettu olevan yli kymmenkertaiset työikäisiin verrattuna. [STM, 1998:b]

Terveyskertomuksen (2002:b) mukaan ongelmana ikärakenteen muuttumisessa on myös elatussuhteen kehittyminen. 1990-luvulla kansantaloudessa yhtä työllistä kohti oli yksi työelämän ulkopuolinen. Tulevaisuudessa suhteen odotetaan kääntyvän epäedulliseksi eli työelämän ulkopuolella on enemmän ihmisiä kuin työntekijöitä.

Sosiaali- ja terveysministeriön projektiraportissa (2002:a) nähdään ongelmana myös lääkärien toimiminen sekä yksityisellä että julkisella sektorilla. Noin 80 % yksityisellä sektorilla toimivista lääkäreistä hoitaa päätoimenaan kunnan virkaa. Tämä aiheuttaa mielestäni hieman moraalisen ongelman, sillä julkisen sektorin ongelmat ajavat potilaita yksityisen lääkärin vastaanotolle, jossa kuitenkin työskentelevät samat lääkärit kuin terveyskeskuksessa. Tällöin potilaan pitää kuitenkin maksaa saamastaan hoidosta. Tulevaisuudessa tähän asiaan olisi syytä kiinnittää huomiota ja kunnallisia virkoja hoitavan lääkärin palkka tulisi asettaa siten, ettei hänen tarvitsisi hakea lisäansiota yksityisen puolelta. Toisaalta lääkärit voivat hakeutua yksityiselle puolelle töihin myös "helpomman" työn takia.

Vuoden 2002 sosiaali- ja terveyskertomuksen mukaan yksi alan tulevaisuuteen vaikuttava merkittävä tekijä on globalisaatio, jolla on sekä suoria ja välillisiä että hyviä ja huonoja vaikutuksia. Elinkeinorakenteen muutoksen vaikutuksia voidaan jo nyt erottaa ihmisten elämästä. Erot ihmisten omaisuuden, henkisen tietämyksen ja fyysisen hyvinvoinnin välillä Suomessa kasvavat ja tämä vaikuttaa myös merkittävästi toimialan toimintaa tulevaisuudessa. Tulevaisuutta ajatellen positiivisena asiana kuitenkin pitää nähdä valtion talouden asettuminen kohtuulliseen tasapainoon 2000-luvun alussa. Janne Lehenkari tuo esille artikkelissaan (2003) positiivisena asiana myös sen, ettei teknologian hyväksikäyttö alalla ole jäänyt vain sosiaali- ja terveyspolitiikan tavoitteeksi vaan esim. vuoden 2001 valtion talousarviossa on osoitettu vajaa 3,5 miljoonaa euroa hankekokonaisuudelle, jonka tavoitteena on tietoteknologian hyödyntäminen vanhusten ja vammaisten itsenäisessä työskentelyssä.

Päivi Röppäsen opinnäytetyössä (2003) kerrotaan, että Suomessa on tavoitteena saada vuoteen 2007 mennessä potilas-, tutkimus- ja hoitotiedot yhtenäiseen muotoon. Tietojen tulee siis olla valtakunnallisesti eri organisaatioiden ja järjestelmien käytettävissä. Luonnollisesti tietosuojan asettamat rajat on otettava huomioon. Tämä tavoite ei ole helppo ja se aiheuttaaakin terveydenhuollon sektorille vaativan haasteen seuraavien vuosien ajalle.

2.4. Lähtökohtia kehitykselle

Suomessa sosiaali- ja terveyspalvelujärjestelmän lähtökohtana on perusterveydenhuollon, erikoissairaanhoidon ja sosiaalitoimen välinen yhteistyö. Valitettavasti käytännön yhteistyössä on suuria puutteita, sillä palveluissa on päällekkäisyyttä terveydenhuollon eri sektoreiden (erikoissairaanhoidon, perusterveydenhuolto, yksityissektori, työterveyshuolto) kesken ja myös esimerkiksi sairaanhoitopiirien sisällä erikoissairaanhoidon eri palvelujen tuottajien (keskussairaalat, aluesairaalat, erikoislääkärijohtoiset terveyskeskukset) kesken. Myös tukipalveluiden tuottamisessa on päällekkäisyyttä. Esimerkiksi potilaille on todettu tehtävän päällekkäisiä röntgen- ja laboratoriotutkimuksia muun muassa koska tutkimustulokset eivät välity hoitopaikasta toiseen. [STM, 1998:b]

Terveydenhuollon toimivuutta koskevan selvitysmiesraportin (1998:b) mukaan erikoissairaanhoidon ja perusterveydenhuollon välillä on puutteita erityisesti läheteiden, konsultaatioiden, epikriisien ja jatkohoitojärjestelyjen osalta. Potilaat kärsivät usein hoitoyksiköiden välisestä heikosta tiedon siirrosta. Heidät saatetaan lähettää turhaan toiseen hoitoyksikköön, jossa heille tehdään turhia testejä. Myös lääkärit kärsivät tilanteesta, sillä raportin mukaan lääkäri saa vain 30 - 60 %:ssa tapauksista hoitopalauteen erikoissairaanhoidon lähettämistään potilaista, poliklinikkakäynneistä vieläkin harvemmin. Potilaan hoidon kokonaisprosessista ei valitettavasti kukaan kannan vastuuta tai toisaalta siihen ei varmaan ole mahdollisuutta käytettävillä olevia resursseja käyttäen.

Niin kauan kun sosiaali- ja terveydenhuolto alan rahoitus valtion puolelta pysyy alhaisena, on alan toimintaa kehitettävä niin, että palvelu pystytään tuottamaan pienemmällä rahoituksella. Röppänen näkee (2003) tärkeimpänä keinona terveydenhuollon kustannuksien rajoittamisen. Kustannusten vähentäminen ei kuitenkaan saa vaikuttaa potilaan hoitoon vaan palvelutason on säilyttävä. Kustannuksia onkin syytä lähteä laskemaan toimintaprosesseja järkeistämällä. Toiminnassa tulisi pyrkiä Röppäsen mukaan sairaus- ja diagnosikeskeisestä ajattelutavasta saumattomiin hoitoketjuihin.

Työolot ovat Sosiaali- ja terveystieteen (2002:b) mukaan yksi merkittävä asia, joka pitää ottaa huomioon alan kehittämistä suunniteltaessa. Vuosina 1997–2000 sosiaali- ja terveydenhuollossa työterveyslaitoksen tutkimuksen mukaan kiire työssä ja työn koettu henkinen rasitus lisääntyivät vähän sosiaali- ja terveydenhuollon alalla, vaikka ne vähenivät hiukan muilla toimialoilla. Sairaslomien määrä ja pituus vaihtelevat kuitenkin huomattavasti alalla ammatin mukaan eivätkä ongelmaa siis voi yleistää koko toimialaan. 1990-luvun loppupuolella sairauslomat lisääntyivät alalla ja työterveyslaitoksen mukaan samansuuntainen kehitys jatkuu edelleen. Sosiaali- ja

terveydenhuollon tavoite- ja toimintaohjelman päätavoitteisiin 2000–2003 sisältyy henkilöstön osaamisen, riittävyden, hyvinvoinnin ja jaksamisen edistäminen. Alalla työn määrän ja vaativuuden kasvun lisäksi myös eettinen kuormitus ja väkivallan pelko ovat lisääntyneet. Työmäärän kasvu ei ole kuitenkaan lisännyt henkilökunnan määrää ja siksi vajaakuntoisten työntekijöiden selviytymien työssä on edelleen vaikeutunut. Työolojen vaikutusta terveydenhuollossa henkilökunnan työhön ei siis pidä vähätellä. Asia on otettava vakavasti, sillä ilman toimivaa henkilökuntaa ei ole toimivaa terveydenhuollon palvelua.

Lehenkari (2003) näkee terveydenhuollon teknologiat yhteiskunnallisesti merkittävänä. Niiden avulla on mahdollisuus ratkaista hyvinvointiyhteiskunnan kasvuvaiheen jälkeisen ajan terveydenhuollon ongelmia. Hän uskoo uusien teknologioiden voivan tehostaa hoitoa ja parantaa palveluiden käyttömahdollisuuksia, laatua, saatavuutta ja seurantaa. Terveydenhuollon teknologian tuottajille ja käyttäjille haasteen muodostaa yhteistoiminnan saavuttaminen ja käytäntöjen opettaminen, sillä vakiintuneita yhteistyötapoja tai -malleja ei ole olemassa. Haaste on myös terveydenhuollon muuttuva, monimutkainen ja hajautettu asiantuntijaorganisaatio. Lehenkari muistuttaa, että tulevaisuudessa tulee ottaa huomioon terveydenhuollon ja hyvinvointiteollisuuden tuotteiden tulo yhä lähemmäksi kuluttajan arkipäivää, kun omahoidon ja itsenäisen suoriutumisen tukeminen yleistyvät. Teknologian käyttö tulee siis olla mahdollista myös normaalille kansalaiselle.

3. Tiedon ja tietojärjestelmien yhteys

Tieto ja tietojärjestelmät liittyvät luonnollisesti toisiinsa. Tietokantaan kerätään tietoa, joita yhden tai useamman tietojärjestelmän on mahdollista käyttää ja päivittää. Tieto ja tietojärjestelmä -käsitteet ovat hiukan monimuotoisempia kuin useat tulevat ajatelleeksikaan, siksi käsitelenkin tässä luvussa hiukan niiden teoreettista merkitystäkin. Kerron myös molempien merkityksestä terveydenhuollossa.

3.1. Tietojärjestelmä

Jarkko Ansamäen (2002) määrittelee tietojärjestelmän johonkin tarkoitukseen suunnitelluksi tietojenkäsittelykokonaisuudeksi. Kokonaisuus muodostuu tietojenkäsittelylaitteista (esim. tietokoneet, laskimet) ja tiedonsiirtolaitteista (esim. verkot, modeemit). Myös ihmiset, jotka käyttävät järjestelmään laitteita on huomioitava osaksi kokonaisuutta. Ihmiset hoitavat tehtävänsä toimintaohjeiden mukaisesti eli myös nämä on käsitettävä osaksi järjestelmää. Luulen, että monelle näin laaja käsite on hiukan yllätys. Useat ihmiset

käsittävät vain järjestelmän teknisen osan. Ripatti (1999) tähdentää, että tietojärjestelmän tarkoitus on tukea tiedon tarkoituksenmukaista kulkua, saatavuutta ja käsittelyä.

3.1.1. Tietojärjestelmä terveydenhuollossa

Sinikka Ripatin artikkeli (1999) käsittelee tietojärjestelmien käyttöönottoa terveydenhuollon alalla. Tietojärjestelmät ovat hankittu terveydenhuoltoon kuten muillekin aloille toiminnan apuvälineeksi. Tietojärjestelmän hankinta toimintayksikköön pitää tukea toimintaa ja samalla myös kehittää sitä eli lopulta parantaa terveydenhuollossa potilaalle tarjottavan palvelun laatua. Suuret tietojärjestelmähankinnat vaativat pitkäaikaisen sitoumuksen järjestelmän toimittajaan. Hankinnat tulee siis suunnitella hyvin ja sen onnistumiseksi on kaikkien osapuolien tehtävä töitä, myös koko hoitohenkilökunnan. Heidänkin pitää olla valmiita oppimaan uutta ja muuttamaan vanhoja käytäntöjä uusiksi.

Röppänen jakaa opinnäytetyössään (2003) terveydenhuollon tietojärjestelmä hankinnat neljään ryhmään käyttötarkoituksen mukaan. Järjestelmä voidaan hankkia **tukemaan toiminnan suunnittelua**. Tällöin hoitohenkilökunnalla ja muilla yksikön työntekijöillä on paremmat mahdollisuuden jäsentää työnsä osaksi laajempaa kokonaisuutta. Tarkoituksena voi olla myös potilaan **hoidon jäsentäminen**. Järjestelmään kirjataan potilaalla oleva ongelma ja etsitään siihen ratkaisuja. Myös hoidon suunnittelun voi hoitaa järjestelmän avulla ja käyttää totutuksessakin apuna.

Hoitohenkilökunnan **kommunikoinnilla** on suuri merkitys potilaan hoidon toteuttamisessa. Tämän vuoksi myös tätä osa-aluetta on tärkeää tukea järjestelmän avulla. Henkilökunnan on kommunikoitava ongelman ratkaisusta ja sen toteuttamisesta. Lisäksi työtovereiden tulee keskustella asioista ja kysyä toisilta tärkeitä tietoja hoidon kannalta. **Toimistotehtävien hoito** vie terveydenhuoltoalalla henkilökunnalta paljon aikaa. Kaikki tiedon on kirjattava, mahdollisesti siirrettävä ja säilytettävä. Näiden toimintojen hoitoon on tarpeen olla myös oma järjestelmätyyppi. [Röppänen, 2003] Mielestäni tämän osa-alueen toteutuksella on tärkein merkitys, sillä silloin henkilökunnan rutiininomaiset tietojen kirjaamiset ja tulosteiden tuottamiset helpottuisivat ja näin henkilökunnalle jäisi enemmän aikaa potilaiden palveluun.

3.2. Tieto - käsitteenä

Filosofiasta tuttu klassisen tiedon määritelmän mukaan tieto on hyvinperusteltu tosi uskomus. Filosofian kannalta en kuitenkaan asiaa enempää käsitellä. Minna Koskinen lähestyy tiedon käsitettä hieman enemmän tietojärjestelmien kannalta. Tiedon tasot voidaan jaotella dataan, informaatioon

ja tietämykseen. Tieto on dataa silloin kun se on koodattu fyysiselle välineelle merkkietona. Informaatioksi Koskinen määrittelee siirtymän datan ja tietämyksen välimaastossa. Tietämys on ihmisen päässä sijaitseva tieto- tai taitokokonaisuus. Tieto on informaatiota vastaanottajalle silloin kun se on hänelle uutta.

3.2.1. Tiedon eri muodot terveydenhuollossa

Tiedolla voidaan nähdä olevan myös terveydenhuollossa erilaisia muotoja. Häkkisen *et al.* (2002) mukaan terveydenhuollon palvelutarjoajat tarvitsevat toiminnassaan yleistä teoreettista tietoa ja empiiristä tietoa. Teoreettinen tieto käsittää esimerkiksi tietoa elimistön toiminnasta, mikä on yleistettävissä teoriassa kaikkiin ihmisiin. Empiirinen tietoa vastaavasti käsittää tiedot kunkin potilaan terveydentilasta. Näitä tietoja ei voi etukäteen opiskella vaan ne ovat tapauskohtaisia ja niitä voidaan tarvittaessa tulkita teoreettista tietoa apuna käyttäen.

Terveydenhuollossa tietoa kerätään monella eri tavalla ja monissa eri tilanteissa. Kerätyn tiedon perusteella tehdään johtopäätöksiä ja hoitoratkaisuja. Häkkinen *et al.* (2002) jakavat terveydenhuollon tiedon seuraaviin muotoihin. Työntekijä ottaa vastaan tietoa (dataa), josta hän koostaa uutta merkityksellistä tietoa (informaatiota). Ajan ja kokemuksen myötä saatu informaatio muuttuu tietämykseksi (knowledge), jota hoitohenkilökunta käyttää työssään apuna. On myös hyvä muistaa, että osa tietämyksestä voi olla tiedostamatonta.

Häkkinen *et al.* (2002) ottavat esille myös informaation vaihdannan, jolla tarkoitetaan kommunikaatiota eli viestintää. Viestintä on informatiivista vain silloin kun se vähentää vastaanottajan epätietoisuutta. Onnistuneeseen viestintään tarvitaan tarve ja kyky viestiä, yhteinen kieli sekä sopiva kanava. Terveydenhuollossa on ollut ongelmana eri merkitsemistapojen käyttö. Tämä vaikeuttaa tiedon siirtoa ja ymmärtämistä. Häkkisen *et al.* mukaan kokonaisuuksien hahmottaminen on sitä vaikeamaa, mitä enemmän tietoa on ja mitä enemmän ne vaikuttavat toisiinsa. Tehtävien hoitaminen vaikeutuu, jollei kaikkea tarvittavaa tietoa ole saatavilla tai jos tieto on väärää tai tulkitaan väärin. Mielestäni on huomioitava myös se, että vastaanottajalle ei välttämättä ole kykyä ymmärtää tietoa.

3.2.2. Tiedon merkitys terveydenhuollossa

Häkkinen *et al.* (2002) näkevät tiedolla olevan terveydenhuollossa tärkeä asema. Tieto on työn edellytys, mutta ei kuitenkaan kohde. Henkilökunta on koulutettu käyttämään ja tulkitsemaan tietoa oikein, mutta ei esimerkiksi muokkaamaan tai siirtämään sitä. Häkkinen *et al.* tähdentävät, että toimiva ja

tehokas sosiaali- ja terveydenhuolto edellyttää sujuvaa tiedonkulkua hoito- ja palveluketjussa. Henkilökunnalla on oltava kyky myös kerätä tietoa tehokkaasti. Tieto on saatava esille tarvittaessa.

Häyrinen ja Saranto (2003) pohtivat artikkelissaan tietohallinnon merkitystä terveydenhuollossa. Tietohallinto tarkoittaa organisaation tietoresurssien hyväksikäytön suunnittelua, johtamista, toteutusta ja valvontaa. Tietoresurssit ovat tietojärjestelmät laajemmassa merkityksessä. Ne koostuvat tietovarastoista, ohjelmistoista, laitteista, tietoliikennejärjestelyistä sekä ihmisistä tietojen lähteinä ja hyväksikäyttäjinä. Häyrisen ja Saranton mielestä tietohallintoa tulisi kehittämään siten, että se edistäisi eri toimijoiden välistä tiedon kulkua, palveluketjujen saumatonta toimintaa ja laadun seurantaa.

Tiedonkulun tärkeän merkityksen vuoksi sen toimivuutta olisi tärkeää pystyä myös tutkimaan. Päivi Röppänen kuitenkin tuo esille, ettei tiedonkulun arviointiin ei ole vielä valmiita välineitä. Häkkinen *et al.* (2002) näkevät, että toimivan hoitoketjun edellytys on toimiva tietoketju. Hoitoketjun pitäisi olla saumaton eli rajapintojen ei pitäisi näkyä potilaalle. Häkkisen *et al.* mukaan tiedon pitää terveydenhuollossa olla kattavaa, oikea-aikaista ja tarkoituksen mukaista. Hoitotyössä tiedolla on myös dokumentoinnin vaatimus. Tiedot pitää dokumentoida ja aiempaa tietoa pitää voida hyväksikäyttöö.

4. Tietojärjestelmien kehittyminen terveydenhuollossa

Kleemolan (1998) mukaan tietotekniikan käytön uskotaan avaavan uusia mahdollisuuksia terveydenhuollon laadun ja taloudellisuuden parantamiseen. Tieto- ja teletekniikoita käyttäen potilastietojen siirtäminen sähköisessä muodossa toimintayksiköiden/rekisterinpitäjän välillä on mahdollista. Tämä verkottuvan tietojärjestelmän käytön tarkoitus on tukea potilaan saumattoman hoidon toteutumista myös silloin kun potilaan hoito jatkuu toisessa terveydenhuollon toimintayksikössä tai toimintayksikkö hankkii hoitoon liittyviä palveluita muilta terveydenhuollon toimintayksiköiltä. Tässä luvussa kerron aluksi tietojärjestelmien kehittymisen historiasta ja matkasta kohti yhtenäistä järjestelmää. Kerron myös saumattomasta palveluketjusta ja sähköisestä potilaskertomuksesta, jotka ovat kehityksen tavoitteita.

4.1. Historiaa

Päivi Röppäsen (2003) mukaan teknologia hyväksikäyttö terveydenhuollossa ole täysin uusi asia, sillä jo 1960-luvulla on puhuttu mahdollisuudesta käsitellä tietokoneella sairaskertomustietoja. Ensimmäiset eräkäyttöiset ja keskitetyt atk-järjestelmät kehitettiin yliopistollisille keskussairaaloille - 70-luvulla, joista sitten 80-luvulla siirryttiin päätekäyttöisiin järjestelmiin. Monet sairaalat ottivat

käyttöön tällöin merkkipohjaisen Musti sovelluksia. Vanha järjestelmä tyyppi säilyi 1990-luvun puolivälille asti, jolloin alettiin suunnitella graafista käyttöliittymää M-pohjaisiin perinnejärjestelmiin. Vähitellen sovellusten kehittämistyö on siirtynyt sairaaloilta kokonaan alan yrityksille.

Myös Antero Ensio kertoo raportissaan (1999) terveydenhuollon ohjelmistojen kehityksestä kuitenkin hieman enemmän ohjelmistoyritysten näkökulmasta. Aluksi ohjelmistoyritykset tekivät lähes kaikki ohjelmistot kullekin asiakkaalle erikseen, hänen tahtojensa mukaan. Näitä ohjelmistoja kutsutaan nimellä räätälöity ohjelmisto. Seuraavassa kehitysvaiheessa syntyi hyvästä räätälöidystä ohjelmistosta uusi asiakaskohtaisesti muunneltu versio uutta asiakasta varten. Näitä kutsutaan tuoteräätälöidyksi ohjelmistoiksi. Näiden valmistaminen ja ylläpitäminen oli asiakasta kohden halvempaa kuin asiakaskohtaisten ohjelmistojen. Kun ohjelmistojen ominaisuudet vakiintuivat, toivat toimittajat markkinoille ohjelmistoja, sama ohjelmisto kaikille, joskin käyttäjän tarpeiden huomioon ottaminen oli mahdollista erilaisilla parametreilla tai asetuksilla. Ohjelmistotuotteet eivät jääneet kansallisiksi vaan nykyään useimmat tuotteet ovat kansainvälisiä.

Hautsalon *et al.* artikkelissa (2002) käsitellään terveydenhuollon tietojärjestelmien historiaa yksikön kannalta. Terveystieteiden tutkimuksessa on ollut jo vuosia käytössä useita erilaisia tietojärjestelmiä, jotka on hankittu mikä mihinkin tarkoitukseen eri aikoina. Etenkin erikoissairaanhoidossa vanhimmat järjestelmät on hankittu yli kymmenen vuotta sitten. Normaalilla arkeilla on ollut, että lääkärin vastaanotolla potilaan asioiden hoitamiseen ja tiedonhallintaan on käytetty jopa neljää eri sovellusta, joita pidetään samanaikaisesti näytöllä auki. Sama tilanne on myös hoitohenkilökunnan tallettaessa hoito- ja laskutustietoja osastolla. Jokainen järjestelmä vaatii aina oman käyttäjätunnuksen ja salasanan. Tiedot eivät kulje järjestelmien välillä vaan jokaisesta järjestelmästä on haettava erikseen esille käsiteltävän potilaan tiedot ja jokaiseen talletetaan usein myös samoja hoitotietoja. Todellisuutta siis ovat järjestelmät, jotka alun perin on hankittu helpottamaan työtä, vaikuttavatkin päinvastoin. Järjestelmien yhteensovittamista ei ole osattu ajatella saatikka vaatia aikaisemmin.

4.2. Kohti yhtenäistä järjestelmää

Kehitys on siis menossa kohti yhtenäisiä järjestelmiä joiden uskotaan tehostavan ja helpottavan terveydenhuollon työtä. Heidi Häkkinen (2003) mielestä muutosten tulisi lähteä jo järjestelmän suunnittelusta. Hänen mielestään tietojärjestelmiä suunniteltaessa pitäisi keskittyä toiminnan mallintamiseen yksittäisten toimintojen sijasta. Toiminto on tavallisesti yhden henkilön suorittama kun taas toimintaan osallistuu joukko ihmisiä, jotka toimivat yhteistyössä toistensa kanssa. Toimintateoreettinen suunnittelumalli

mahdollistaa erilaisten roolien työtoimintojen kuvaamisen. Ihmisen toimintaa tulee tarkastella eri tasoilta; yksilö, ympäristö ja sosiaalinen toiminta. Menestyksellisen tietojärjestelmän integraation vaikutukset voivat olla laajemmat kuin aluksi on odotettukaan. Kehitys voi ulottua työtoiminta ja toimintamallien kehittämiseen laajemminkin.

Hautsalo *et al.* (2002) tuovat esille myös muita ongelmia. Yhteisen tiedotuskanavan puute heikentää siis perusterveydenhuollon ja erikoissairaanhoidon yhteistyötä. Potilaan tiedot ovat monessa eri järjestelmässä ja organisaation osassa niin, että niiden siirtäminen tietojärjestelmälle on usein ontuvaa. Monesti puhelin onkin käytetyin tapa asioiden järjestämiseksi ja hoitamiseksi. Järjestelmien ongelmana on myös yhteisen tiedon esittämismuodon puuttuminen. Tiedot esitetäänkin usein vapaamuotoisena tekstinä, jonka tulkitseminen ja eteenpäin siirtäminen voi olla ongelmallista. Nämä tietojen moninaiset esitystavat saavatkin lääkärit määräämään usein uusinta kokeita, kun toisen toiminta yksikön muistiinpanoja ei osata tulkita.

Sairaskertomuksen standardisoinnista ja digitaalista lähete- ja hoitopalautejärjestelmää on yritetty kehittää jo monien vuosien ajan. Monia alueellisia ja muita kehittämisprojekteja on toteutettu, missä tarkoitus on ollut kehittää tiedon kulkemista yksiköiden välillä, mutta käytännöt ei ole juurtunut yleiseen käyttöön. Yhtenä ongelmana on ollut, että projektit aloitetaan aina alusta eikä vanhoja tietoja osata/ haluta käyttää hyödyksi. Toisena ongelmana on eri ohjelmistojen kirjo sekä perusterveydenhuollossa että erikoissairaanhoidossa. Jatkuvasti markkinoille tulevat uudet ohjelmistot eivät helpota yhtään tavoitetta yhtenäisestä järjestelmästä. Yhteensopivat tietojärjestelmät mahdollistaisivat niiden samanaikaisen käytön tai toiminnan niin, että tieto päivittyy samanaikaisesti myös toisiin ohjelmiin. Tietojen siirtyminen sähköisessä muodossa järjestelmien välillä pitäisi saada mahdolliseksi. [Hautsalo, Häyrinen ja Korhonen, 2002]

4.2.1. Tietojärjestelmien integrointi

Röppänen (2003) näkee, että tietojärjestelmien integrointitarvetta on lisännyt uusien tietojärjestelmien käyttöönotto. Aikaisemmin käytössä ollut Mustiperinnejärjestelmä oli periaatteessa yhtenäinen, mutta se ei enää tukenut uusien tekniikoiden käyttöä. Uudet järjestelmätyypit eivät ole yhdistettävissä Mustijärjestelmiin. Myöskään kaikkien uusien sovellusten, jotka ovat toteutettu erilaisilla tekniikoilla, yhteensovittaminen ei ole mahdollista. Röppänen kuitenkin tähdentää, että yhteensovittaminen on välttämätöntä, jotta potilaan hoidon kannalta keskeiset tiedot olisivat nopeasti ja helposti hoitohenkilökunnan saatavilla.

Röppänen (2003) määrittelee, että tietojärjestelmien integroinnin toiminnaksi, jossa pyritään yhteen toimivuuteen kahden tai useamman järjestelmän välillä. Yhteen toimivuus on järjestelmien kykyä siirtää tietoa nopeasti toistensa välillä ja käyttää siirrettyä tietoa. Röppäsen mukaan integroinnin ongelma on tiedon siirron turvallisuus, luotettavuus, muuttumattomuus ja nopeus.

4.3. Saumaton palveluketju

Sosiaali- ja terveysministeriön työryhmän tuottama muistio (1998:a) käsittelee mm. saumatonta palveluketjua, jonka nähdään olevan merkittävä sosiaali- ja terveydenhuollon kehittämiskeino. Palveluketju on laajempi käsite kuin hoitoketju, sillä se kattaa julkisen ja yksityisen sektorin sekä järjestöjen tuottamat palvelut samoin kuin toimeentulo- ja sosiaaliturvaetuisuudet. Jotta asiakkaan hoitokokonaisuuden kattava ja organisaatorajat ylittävän saumattoman palveluketjun suunnittelu ja hallinta olisi mahdollista, tulee sosiaali- ja terveyspalvelujen tuotossa ottaa se huomioon. Asiakkaiden monimuotoiset ja verkottuvat palveluketjut vaativat toimiakseen hyvää integrointia ja palvelukokonaisuuden hallintaa.

Muistiossa määritellään (1998:a) asiakaskeskeisen palveluketjun tavoitteeksi, että asiakas on palvelujen keskipisteessä ja että hän saa tarpeellisen palvelun, hoidon, ohjeet ja muun informaation. Asiakkaan näkökulmasta palveluketjun tulisi muodostaa jatkuva kokonaisuus ilman tarpeettomia jonotusaikoja ja viiveitä. Vastaavasti palvelun antajalla tulee olla käytettävissä asiakasta ja hänen ongelmaansa koskevat tarpeelliset tiedot viivytyksettä, jotta asiakkaan odotusajat, pompottelu ja palvelun saannin viiveet voidaan minimoida. Saumattomuuden esteenä muistiossa tuodaan esille organisaatiokeskeiset toimintatavat. Alalla on paljon yksiköitä, joiden toimintatavat eroavat toisistaan ja tämä tekee hankalaksi yhtenäisen saumattoman palveluketjun sovittamisen kaikkiin yksiköihin. Tärkeää olisi muodostaa yhtenäiset toimintatavat, jolloin tieto kulkisi paremmin. Muutenkin yhteistyön merkitys on tärkeä.

Muistion mukaan (1998) saumaton palveluketju edellyttää saumatonta informaatioketjua. Tämä mahdollistuu parhaiten, kun tarpeellinen tieto on tallennettu digitaaliseen esitysmuotoon ja tiedon tarvitsijalla on käytettävissä ohjelmistot digitaalisen tiedon lukemiseen sekä tietoverkot tiedon siirtämiseen. Suomalaiselle sosiaali- ja terveydenhuollon tietojärjestelmille on tyypillistä, että asiakastiedot ovat tallennettu organisaatio- ja palveluntuottajakohtaisiin paikallisiin tietojärjestelmiin (ns. perinnejärjestelmiin). Nykyisin käytössä olevat asiakas- ja potilastietojärjestelmät on rakennettu tukemaan

organisaatiokeskeistä toimintatapaa ja siksi ne tukevat heikosti palveluketjumallia.

Muistiassa (1998) tuodaan esille asioita, jotka mahdollistavat tietojärjestelmien yhteistoiminnallisuuden. Yhteistoiminnallisuus vaatii standardeja, käytössä olevat yhteiset käsitteet, luokitukset ja ohjelmistorajapinnat. Nykyiset suomalaiset sosiaali- ja terveydenhuollon tietojärjestelmät käyttävät erilaisia ja yhtenäistämättömiä käsitteitä ja osin myös luokituksia. Myös terveydenhuollon tietojärjestelmästandardit ovat kehittymättömiä ja olemassa olevatkin standardit on otettu heikosti käyttöön. Vaikka maamme tietoverkkostruktuuri on varsin kehittynyttä, ei tietoverkko-operaattorien välinen yhteistyö tue riittävästi paikallisten ja alueellisten tietoverkkojen yhteistoiminnallisuutta.

4.4. Potilaskertomus sähköiseen muotoon

Häyrisen ja Saranton artikkelissa (2003) kerrotaan potilaskertomuksen historiasta. Terveydenhuollossa kerätään kustakin potilasta potilaskansio, joka sisältää kaiken tiedon potilaan saaduista hoidoista ja käynneistä kyseisessä hoitolaitoksessa. Perusterveydenhuollossa sairaskertomus otettiin käyttöön vuonna 1982. Sairaskertomuksen tarkoitus oli potilaan tietojen dokumentointi paperimuotoisena niin, että niiden tutkiminen myöhemmin oli mahdollista. Ongelmana paperimuotoisissa kertomuksissa on koettu papereiden etsiminen, katoaminen ja siirtäminen. Lisäksi tiedon on voitu kirjata ja niitä vain tutkia siellä missä kansio fyysisesti sijaitsee.

Kiviaho tuo artikkelissaan (2002) esille terveydenhuollossa aikaisemmin käytössä olleen paperisen potilaskertomuksen ongelmia. Hänen mukaan manuaalisen potilaskertomuksen kerääminen on työlästä ja aikaa vievää. Kertomuksen tuottamisprosessi koostuu saneluista, niiden purusta, konekirjoituksesta ja arkistotulostuksista. Häyrinen ja Saranto (2003) tuovat vastaavasti esille elektronisen potilaskertomuksen hyviä ja huonoja puolia. Elektronisen potilaskertomuksen tavoitteena on poistaa tämä tehoton ja työläs työmuoto ja siirtää kaikki tähän asti paperimuodossa olleet potilaskertomukset sähköiseen muotoon. Näin tietojen tallettaminen, säilyttäminen, eteenpäin välittäminen ja käyttäminen olisivat mahdollista tietotekniikan avulla. Ongelmana elektronisen kertomuksen kehittämisessä on ollut tähän asti organisaatiokohtaiset räätälöidyt järjestelmät, joihin yhtenäisen elektronisen kertomuksen sovittaminen on ollut mahdotonta.

Tulevaisuudessa elektroninen potilaskertomus ei ole vain hoitohenkilökunnan käytettävissä. Siirryttäessä kohti asiakaslähtöisempää toimintaa tulee myös potilaalla olla mahdollisuus tarkastella omia tietoja verkon kautta ja mahdollisesti lisätä itse jotakin tietoja. Tämä mahdollistaa

lisäämään tulevaisuudessa myös kotihoitoa, jos potilaan on mahdollista päästä tietokantaan kotoa. Elektroninen potilaskertomus mahdollistaa siis sekä tietojen käytön monessa paikassa samanaikaisesti, että tietojen generoimisen tarpeen mukaisesti. [Häyrinen ja Saranto, 2003] Röppäsen mukaan (2003) elektronisin potilaskertomusjärjestelmän etuihin voidaan lukea myös erimuotoisten tietojen käsittely mahdollisuus. Sillä voidaan käsitellä jatkuvaa kertomustekstiä, strukturoitua ja määrämuotoista tietoa (lomakkeet, laboratoriotulokset), tallettaa digitaalista kuvaa sekä ääntä. Kertomustietojen kirjaamiseen voidaan käyttää tavallisten PC-työasemien lisäksi myös kannettavia mikro- ja kämmenkoneita.

5. Yhtenäisen järjestelmän vaikutus alaan

Tavoitteena oleva yhtenäinen tietojärjestelmä terveydenhuoltoon ei ole helposti toteutettavissa. Asian kanssa on tehtävä vielä paljon töitä ennen kun toimiva ja palveleva järjestelmä saadaan käyttöön kaikkialle. Luonnollisesti uudella järjestelmällä toivotaan olevan vain positiivisia vaikutuksia, mutta näin ei kuitenkaan ole. Vaikka järjestelmällä on etuja potilaille, henkilökunnalle ja koko yhteiskunnalle, aiheuttaa se myös ongelmia heille kaikille. Järjestelmän hankinnassa on tulevaisuudessa tarkastella vaikutuksia monelta eri kannalta. Jo aikaisemmissa luvuissa on tullut esille yhtenäisen järjestelmän etuja ja mahdollisia ongelmia. Esitän kuitenkin tässä luvussa vielä joitakin etuja ja ongelmia.

5.1. Etuja

Järjestelmän käytöstä seuranneita etuja pitäisi kaiken järjen mukaan olla paljon, muutenhan järjestelmän hankintaan ei edes kannattaisi lähteä. Perusterveydenhuollolla on yhteiskunnallisesti merkittävä merkitys ja siksi järjestelmän hankintaprosessilla on suuret vaatimukset. Toimivan uuden järjestelmän toivotaan tuovan välillisesti etuja myös koko yhteiskunnalle.

Häkkisen mukaan (2003) hoitovirheitä voidaan vähentää terveydenhuollossa erilaisia tieto- ja viestintätekniikoita käyttämällä. Tästä on ymmärrettävästi etua potilaille ja henkilökunnalle, mutta tärkeä edunsaaja on mielestäni myös palvelun tarjoaja eli usein kunta/kaupunki. Kunta/kaupunki on vastuussa alan toiminnasta ja silloin sille on hyvä asia olla tunnettu vähäisistä hoitovirheistä. Miettisen *et al.* (2003) mukaan informaatiotekniikan käyttö mahdollistaa hoitoketjujen lyhentämisen. He uskovat myös, että yhteistyön henkilökunnan ja hoitoyksiköiden välillä paranee. Toiminnan tehostuessa voidaan odottaa kustannustenkin vähenevän eli tällöin tulisi

säästöjä. Toinen asia sitten on mihin nämä säästöt menevät. Palkataanko niillä esimerkiksi uutta henkilökuntaa vähentämään kiirettä? – Enpä usko

Tampereen perusterveydenhuollon strategian (2002) mukaan hoitohenkilökunnan kouluttaminen helpottuu, kun se on mahdollista omassa työpisteessä. Erilaiset oppaat ja käsikirjat on helpommin saatavilla ja niiden tiedot ovat ajan tasalla päivityksen helpotuttua. Tärkeää mielestäni on kuitenkin muistaa, että päivittyminen ei tapahdu itsestään vaan jonkun on se tehtävä, jotta ajankohtaista tietoa saataisiin. Lisäksi on otettava huomioon että liian useasti tapahtuva muutokset ja niitä seuraavat koulutukset eivät henkilökuntaa varmastikaan innosta.

Monet asiat tulee nähdä etuna sekä henkilökunnalle että potilaille. Tampereen perusterveydenhuollon strategian (2002) mukaan tietojen siirtymisessä sähköisessä muodossa on seuraavia hyötyjä: tiedonsiirto mm. lyhentää hoitoon pääsyaikaa, tutkimustuloksen käyttöön saanti nopeutuu, päällekkäisten tutkimusten tekeminen eliminoituu, asiantuntijan konsultointi nopeutuu jne. Toisessa toimintayksikössä syntyneitä aikaisempia hoitotietoja voidaan käyttää hyväksi myös potilaan hakeutuessa itse hoitoon toiseen hoitoyksikköön. Papereiden arkistotyö vähenee. Myös mahdollisuuksia siirtyä osittaiseen internet-ajanvaraukseen ja omalääkärikonsultaatioon selvitetään ja kokeillaan ainakin Tampereella Pegasos-järjestelmän käyttöönoton yhteydessä.

5.2. Mahdollisia ongelmia

Koivisto ja Aaltonen (2003) näkevät ongelmana työkäytäntöjen epäyhtenäisyyden. Vaikka käyttöönotossa asiaa kiinnitetäisiinkin huomiota, niin, usein tietojärjestelmien käyttöönoton jälkeen luodaan omia toimintatapoja. Asiaan ei Koiviston ja Aaltosen mukaan ole kiinnitetty tarpeeksi huomiota tietojärjestelmiin liittyvässä tutkimuksessa. Organisaation kannalta ongelmaksi saattaa muodostua epäyhtenäisyyden seurauksena ei saadakaan parantavaa tulosta ja laatua jota tavoiteltiin. Järjestelmähankinnan epäonnistuminen voi johtua usein siitä, että johto ei sitoudu tarpeeksi projektiin.

Ongelmia voi aiheuttaa henkilökunnan vastustus. He eivät välttämättä osaa käyttää tai eivät vain halua käyttää järjestelmää. Henkilökunnan asenteisiin on tärkeää yrittää johdon vaikuttaa, mutta se ei ole helppoa. Henkilökuntaa on paljon ja useimmat ovat toimineet alalla kymmeniä vuosia. Vanhoista tavoista ei ole helppo päästä irti. Minä näenkin yhtenäisten toimintamallien saavuttamisen suurena haasteena. Järjestelmän toimivuutta voi tarkastella monelta kannalta. Se ei välttämättä toimi teknisesti tai ei yksikköä palvelevalla tavalla.

6. Terveydenhuollon tietotekniikan erityispiirteitä

Kun ohjelmistoyritys kehittää ohjelmistoa terveydenhuollon toimintaympäristöön, on sen otettava huomioon asioita, jotka ovat juuri tälle ympäristölle ominaisia. Tässä luvussa tuon esille aluksi, miten laaja toimintaympäristö vaikuttaa tietotekniikan kehittämiseen. Tämän jälkeen kerron erilaisista laeista ja asetuksista, jotka määräväävät paljon ohjelmiston rakennetta. Tärkeää on myös tuoda esille erilaisten rekisterien merkitys alalla. Lopuksi kerron hiukan, mitä toimittaja-käyttäjä väliseltä suhteelta vaaditaan.

6.1. Laaja toimintaympäristö

Terveydenhuollon toimintaympäristö on laaja. Röppänen (2003) muistuttaa, että laajassa ympäristössä on myös paljon erilaisia järjestelmän käyttäjiä, joiden käyttötaidot ovat usein hyvin eritasoisia. Työyksiköt koostuvat moniammatillisista ryhmistä, joissa vallitsee erilaisia hierarkioita ammattiryhmien välillä ja niiden sisällä. Hän tuo esille myös erityyppiset päätelaitteet, joilla terveydenhuollon tietojärjestelmiä käytetään. Tämä on siis syytä ottaa myös huomioon suunnittelussa. Myös laitteistokanta hoitoyksiköissä vaihtelee suuresti.

Kun otetaan huomioon terveydenhuollon työntekijöiden ikärakenne ja ennakkotiedot, on Miettisen *et al.* (2003) mielestä mahdotonta vaatia heitä esittämään uudelle teknologialle etukäteen vaatimuksia. Ratkaisuna he näkevät koekäytön liittämisen osaksi käyttöönottoa. Teknologia sovitetaan olemassa oleviin välineisiin, työnjakoon ja organisaatioiden välisiin yhteistyösuhteisiin. Käyttäjän kannalta laitteiden ja välineiden on tuettava ja täydennettävä toisiansa, eivätkä ne saa aiheuttaa pelkästään lisätyötä. Käyttöönnotossa on myös otettava huomioon, miten laite sopii olemassa olevaan työnjakoon ja vaatiiko sen käyttö työn uudelleenorganisointia.

6.2. Lait ja asetukset

Lappalaisen *et al.* raportissa Terveydenhuollon erillisrekisterit (1997) kerrotaan terveydenhuollon toimintaan vaikuttavista laeista ja asetuksista. Mm. henkilörekisterilaki (471/1987) ja asetus (476/1987) säätelevät henkilötietojen keräämistä, tallettamista, käyttämistä ja luovuttamista (471/1987). Myös erilaisia säännöksiä on paljon jotka velvoittaa tietojen tai asiakirjojen salassapitoon. Terveydenhuollon kannalta tärkein laki on 13§, joka koskee potilaana asemaa ja oikeuksia. Lain mukaan potilasasiakirjoihin sisältyvät tiedot ovat salaosapidettäviä ja niiden käyttäminen ja siirtäminen on sallittua vain potilaan suostumuksella.

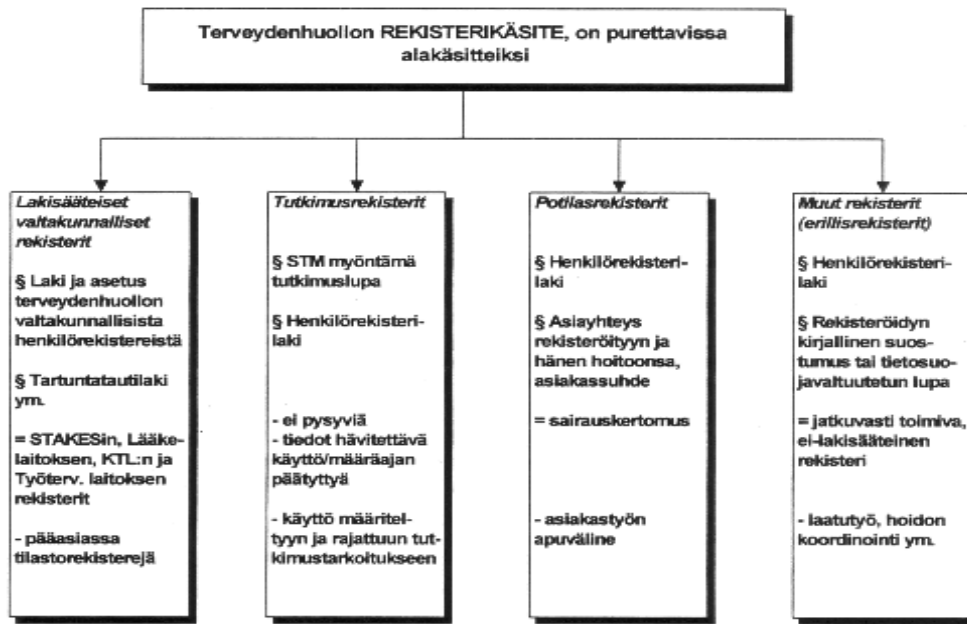
Klemolan (1998) huomauttaa, että tietojärjestelmien suunnittelussa ja kehittämisessä tulee myös ottaa huomioon nämä kymmenet eri laet ja

asetukset, jotka ohjaavat kaikkea henkilön tietojen käsittelyä. Terveystieteiden toimintayksikön vastuu on huolehtia siitä, että sen potilasrekisterit ja ylläpitävät järjestelmät täyttävät lainsäädännön vaatimukset. Potilasasiakirjat laaditaan rakenteellisesti siten, että ne ovat potilaan hoidon sekä hoidon jatkuvuuden kannalta tarkoituksenmukaiset. Hoitojakson/käynnin tietoja ei ole oikeus päivittää jälkikäteen, kuitenkin meneillä olevan hoitojakson tietojen päivitys on sallittua. Tietojen luovuttamiseen ja siirtämiseen esimerkiksi toiseen hoitoyksikköön tulee olla aina toiminnallisesti perusteltu syy. Tietojen luovuttamisessa tulee varmistua salassapitovelvoitteen täyttymisestä.

Terveystieteidenhuollossa on tarpeen käyttää runsaasti kaikkia eri multimedian tietotyyppisiä. Ension (1999) mukaan tämä ei yksin olisi poikkeuksellista, mutta kun huomioon otetaan vaatimukset tietoturvan ja arkistoinnin suhteen niin asiaan pitääkin jo kiinnittää enemmän huomiota. Kiistämättömyys (oikeusturva) ja salassa pidettävyys (tietoturva) ovat moniin muihin aloihin poikkeuksellisia ominaisuuksia. Myös kahdenkymmenen vuoden jälkeen pitää voida luottaa siihen, että tiedot ovat oikeita ja ettei niitä ole voinut tutkia kukaan henkilö, jolla ei ole siihen oikeutta. Arkistokelpoisuus on myös tärkeä kriteeri. Tietojen on oltava koko ihmisen elinikä tutkittavassa muodossa. Tiedot omistaa potilas ja niiden tutkimiseen tarvitaan siis aina hänen suostumus.

6.3. Rekisterit

Lappalaisen *et al.* (1997) mukaan terveystieteidenhuollossa käytössä olevat rekisterit voidaan jaotella usealla tavalla. Yhtenä jaottelutapana he pitävät seuraavaa (kuva 1). **Lakisääteiset valtakunnalliset rekisterit**, jotka perustuvat lakiin ja asetukseen terveystieteidenhuollon valtakunnallisista henkilörekistereistä tai erillislakeihin. Nämä erottuvat omaksi ryhmäksi juridisesta näkökulmasta katsottuna. Toisen ryhmän muodostavat **tutkimusrekisterit**, jotka toimivat sosiaali- ja terveysministeriön myöntämällä tutkimusluvalla tiettyä yksilöityä tutkimusta varten. Hoitoyksiköiden **potilasrekisterit** (sairauskertomusarkistot) sisältävät hoitosuhteessa olevien omien potilaiden tiedot. Näiden lisäksi on myös **muut rekisterit**, jotka voidaan jakaa sen mukaan millainen lupa rekisteritietojen tallentamiseen on. Rekisterillä voi olla esim. tietosuojalautakunnan lupa tai potilaan kirjallinen lupa.



Kuva 1 Terveydenhuollon rekisterikäsitteen jaottelu

Lappalainen *et al.* (1997) korostavat tekstissään rekisterinpidolle asetetuista yleisistä vaatimuksista. Rekisterinpidon tulee olla asianmukaista, suunnitelmallista ja huolellista. Lisäksi suojaamis- ja hävittämisvelvoitteet vaativat käyttöoikeuksien määrittämisen siten, että kukin henkilö pääsee vain niihin tietoihin, joita hän tehtäviensä vuoksi tarvitsee ja tietojen hävittäminen tapahtuu asetusten mukaisesti. Henkilörekisterilaki määrää monista asioista henkilörekisteriä koskien. Rekisterin käyttötarkoitus sekä säännönmukaiset tiedonsiirrot (tietojen hankinta- ja luovutustavat ja tietosisältö) on määriteltävä ennen henkilötietojen järjestämistä henkilörekisteriksi. Rekisterin käyttötarkoitus määrää, minkälaisia tietoja rekisteriin saadaan tallettaa ja mihin sen tietoja saadaan käyttää.

Lappalainen *et al.* (1997) uskoo tietotekniikan kehittymisen helpottavan rekisterien ylläpitoa ja tulosten analysointia tulevaisuudessa. Tällä hetkellä rekistereiltä kuitenkin heidän mukaan puuttuu sekä rahaa, aikaa että ammattitaitoa tietotekniikan tehokkaaseen hyödyntämiseen. Yleensä sairaaloiden atk-osastot ovat rutiinityöllä liian kuormitettuja voidakseen osallistua rekisterien tukemiseen. Nyt tietoa kerätään ja tallennetaan rekistereihin, mutta tiedon analysointi ja palaute on usein heikkoa tai ainakin niissä on pitkä aikaviive. Yleensä rekisteröitävät tiedot tulevat ylläpitäjälle kaavakkeella ja ne tallennetaan ylläpitopaikassa atk:lle. Monien toiveena on saada suora online-yhteys keskusrekisterin ja muiden rekisterien välille. Useita rekistereitä ylläpidetään yksittäisellä mikrolla ja joitakin keskustietokoneella.

Tietojen ilmoitustiheys vaihtelee päivittäisestä kerran vuodessa tapahtuvaan. Monet korostavat tiedon nopean kulun tärkeyttä.

Lappalaisen *et al.* (1997) mukaan Rekisterien taloudellinen asema tulee turvata, jotta luodaan edellytykset hyvälle toiminnalle. Pitääkin erityisesti pohtia sitä, kenelle kuuluu rekisterien toiminnan rahoitus. Rekisterit voivat hyödyntää olemassa olevia potilashallinnon järjestelmiä. Tämä vähentää turhaa työtä, jota tehdään kirjattaessa potilaan perustiedot moneen paikkaan. Rekisterit voivat olla myös kiinteitä osia potilashallinnon järjestelmissä. Sairaanhoidopiirit, yliopistosairaalat ja keskussairaalat voivat koota itse rekisteritietonsa ja keskittää rekisterien teknisen tuen atk-osastolle.

6.4. Tuottajien ja käyttäjien suhde

Lehenkari tuo esille artikkelissaan (2003) ohjelmiston tuottajien ja käyttäjien välisen suhteen. Tuotekehittäjät ajattelevat usein asioita liian teknologia keskeisesti eikä heillä ole yleensä koulutusta terveydenhuollon alalta. Tämä aiheuttaa ongelmia, sillä ohjelmiston suunnittelu vaatii hyvän tietämyksen alasta johon järjestelmää suunnitellaan. Tuotekehittäjien ja käyttäjien yhteistyöllä onkin siis suuri merkitys järjestelmän rakenteen ja käyttöönoton onnistumiselle.

Lehenkarin mukaan ongelmia tuottajille aiheuttaa mm. terveydenhuollon monimutkainen ja ulkopuolisen vaikeasti ymmärrettävä organisointitapa. Päätökset tulevat useimmin muilta tahoilta kuin itse loppukäyttäjiltä. Myös Miettinen *et al.* (2003) näkevät ongelmana sen, että käyttö ja harkintapäätösten tekeminen tapahtuvat terveydenhuollossa usein eri organisaatioissa. Lehenkarin mukaan lisäksi on otettava huomioon, että henkilökuntaa on hoitoyksiköissä paljon ja niiden välinen vuorovaikutus ei aina toimi. Myös opettaminen ja oppiminen eivät tapahdu käden käänteessä. Usein käyttäjät eivät osaa/uskalla vaatia asioita koskien ohjelmistoja. Lehenkari toteaaakin, että terveydenhuollon organisaatioiden olisi rohkeasti vaadittava tuotteiden ja teknologioiden kehittämistä omia tarpeita vastaaviksi. Miettinen *et al.* näkevät ongelmana myös käyttäjien passiivisen sopeutuvan asenteen. Teknologiaa pidetään monimutkaisena ja sitä koskevien vaatimusten ajatellaan edellyttävän erityistä asiantuntemusta. Kynnys vaatia muutoksia ja parannuksia voi nousta korkeaksi.

Uuden järjestelmän hankinta on pitkäaikainen sitoumus sekä toimittajalle että asiakkaalle. Miettisen *et al.* (2003) mukaan tämän pitäisi muistaa teknologian käyttäjäorganisaation, jonka tulisi vaatia pitkäkestoista sopimusta tuotteen muunneltavuudesta ja räätälöitävyydestä käyttötoiminnan tarpeita vastaaviksi. Pelkkä huolto- ja päivityssopimus ei riitä vaan sen lisäksi olisi sovittava tuotteen edelleen kehittelystä tarpeen tullen. Uudet

järjestelmähankinnat ovat taloudellisesti suuria sitoumuksia. Miettinen et al. huomauttavat, että rahoituksessa tulee ottaa huomioon teknologiahankinnan lisäksi myös organisaatiolle pitkälle aikavälillä aiheutuvat muut kuut kuten käyttäjäkoulutus ja työnjaon uudelleen järjestäminen.

7. Pegasos

Tässä luvussa esittelen lyhyesti Pegasos-nimisen sosiaali- ja terveystietojärjestelmän, jonka on kehittänyt Novo Group Oyj. Tuotteen tiedot ovat peräisin Novo Groupin tuote-esityksestä eli ne antavat hiukan yksipuolisen kuvan järjestelmästä. Valitettavasti minulla ei ollut mahdollisuutta saada muuta tietoa tästä kaupallisesta järjestelmästä. Otan kuitenkin lopuksi esille joitakin käyttötoissa tulleita ongelmia.

7.1. Tuote

Pegasos on sosiaali- ja terveystietojärjestelmä, jonka on tehnyt Novo Group Oyj. Se on viime vuosien aikana korvannut Suomessa monissa kaupungeissa aiemmin käytetyt moninaiset atk-järjestelmät. Sen tarkoitus on palvella kaikkia hoitotyötä tekeviä vähentäen rutiinitöitä ja parantaen tiedon saatavuutta. Asiakkaan näkökulmasta järjestelmän tarkoitus on nopeuttaa ja parantaa palvelua. Järjestelmää voidaan käyttää terveyskeskuksissa, sosiaalitoimen yksiköissä ja sairaaloissa. Se voidaan ottaa käyttöön joko yhtenä kokonaisuutena tai osa-alue kerrallaan. Järjestelmä on helposti laajennettavissa käyttäjämäärän tai toimintojen mukaan. [Novo, 2003]

Järjestelmässä on tietokanta, johon tiedot talletetaan ja jossa tietokantaohjelmisto huolehtii tietojen eheydestä ja niiden turvaamisesta säädösten mukaisesti. Pegasos on tuoteperhe, joka koostuu monista eri yhteensopivista järjestelmistä. Siihen voidaan integroida myös muita sektorilla toimivia järjestelmiä ja tietoliikenneominaisuudet takaavat yhteyden muihin terveydenhuollon organisaatioihin, rahalaitoksiin, eläkelaitokseen ja tietopankkeihin. Pegasoksessa on myös muita asiakaspalvelua parantavia etuja. Muun muassa lääkäri voi kirjoittaa sähköisiä lähetteitä ja konsultoida erikoislääkärin kanssa. Tutkimustulokset saadaan sähköisesti ilman hidastavaa postinkulkua. Kun asiakas soittaa puhelimella, hänen tietonsa ovat heti lääkärin luettavissa. [Novo, 2003]

Avohoito, osastohoito ja kotihoito ovat keskeiset osa-alueet Pegasos potilas- ja terveystietojärjestelmässä. Näiden osa-alueiden tueksi on kehitetty kattava joukko erilaisia lisäohjelmia helpottamaan henkilökunnan jokapäiväisiä työtehtäviä. Avohoidon sovellusosa sisältää mm. työaseman, asiakas- ja potilaskertomuksen ja laskutuksen. Osasto- ja kotihoito osa-alueet muodostavat

kokonaisuuden joka tukee moniammatillista yhteistyötä. Ohjelmistojen avulla osastot saadaan liitettyä osaksi asiakkaan saamaa palvelu- ja hoitokokonaisuutta. [Novo, 2003]

7.2. Käyttöönotto

Täysin ongelmitta uuden järjestelmän käyttöönotto terveydenhuolto sektorilla ei ole tapahtunut. Koivisto ja Aaltonen tuovat esille artikkelissa (2003), että Pegasos-järjestelmän käyttöönotto Turun terveystoimen päivystysvastaanotollakaan vuosina 2000–2001 ei sujunut aivan ongelmitta. Käyttöönottoa seurannet yllättyneitä heti alkuaikoinakin tiuhaan tulleista päivityksistä. Työkäytännöt eivät kerinneet vakiintumaan kun jo uusi muutos järjestelmän toimintaan tuli. Tästä seurasi helpolla epäyhtenäisiä toimintatapoja kun uutta koulutusta ei keritty antamaan ja silloin työyhteisön piti luoda oma toimintamalli. Epäyhtenäisistä toimintamalleista ei välttämättä synny haittaa, mutta pahimmassa tapauksessa terveydenhuollon sektorilla se saattaa vaikuttaa hoidon laatuun ja sitä kautta potilasturvallisuuteen.

Myös Helsingissä oli vuonna 2002 ongelmia käyttöönoton jälkeen. Siellä Pegasos ”jumittui” liian suuren yhtäaikaisen käytön takia. Varmasti jokaisella järjestelmän käyttäjällä on oma mielipiteensä järjestelmän käytettävyydestä ja hyödyllisyydestä. Valitettavasti tämän raportin osalta minun ei ollut mahdollista käydä haastattelemassa todellisia käyttäjiä. Luulen kuitenkin, että kaikkeen uuteen totutaan, kunhan asiat eivät muutu jatkuvasti ja niihin koulutetaan asianmukaisesti. Tietysti käyttäjien mielipide on kiinni siitäkin, kuinka palvelevan kokonaisuus yksikköön hankitaan. Täytyy siis muistaa, että järjestelmä ei ole kaikille samanlainen vaan sen toiminta riippuu siitä, kuinka paljon eri osioita siihen on liitetty.

8. Yhteenveto

Sosiaali- ja terveydenhuollon toimintaympäristön laajuus tekee alaa käsittelevästä tutkimuksesta vaikeaa. Kirjallisuuden perusteella seuraavien vuosien aikana pitäisi kehitystä terveydenhuollon palveluissa ja tietojärjestelmissä tapahtua. Mielenkiintoista olisi viiden vuoden kulutta tutkia onko tavoitteena olleet yhtenäiset käytännöt saatu hoitoyksiköihin ja toimiiko saumaton palveluketju -periaate.

Toimintaympäristöllä on yhteiskunnallisesti suuri merkitys. Ihmisillä on Suomessa oikeus palveluihin ja he myös vaativat sitä. Ongelmat alalla ovat saaneet ihmiset vaatimaan parempaa palvelua. Enää ihmiset eivät kuitenkaan suostu jonottamaan monia tunteja lääkärille pääsyä. Jotakin on tehtävä. Yhtenäisten järjestelmäratkaisut tehostavat varmasti alan toimintaa, jos ne vain

saadaan toimimaan suunnitellulla tavalla. Niiden käyttöönotto ei tule kuitenkaan olemaan helppo tehtävä.

Viiteluettelo

- [Ansamäki, 2002] Ansamäki Jarkko, Tietojärjestelmien suunnittelu (luentokalvo) <http://www.kyamk.fi/~zjaan/kurssit/tjs/Osa2.ppt> tarkistettu 10.12.2003
- [Ensio, 1999] Ensio, Antero, *Strateginen selvitys terveydenhuollon tietojärjestelmien standardoinnista ja ehdotus Suomen panostuksesta standardointiin tulevaisuudessa*. Sosiaali- ja terveysalan tutkimus- ja kehittämiskeskus, Helsinki, 1999.
- [Hautsalo, Häyrinen ja Korhonen, 2002] Hautsalo Annikki, Häyrinen Kristiina ja Korhonen Maritta, Terveydenhuollon tietojärjestelmien yhteensopivuus - kaukainen tavoite vai pian todellisuutta <http://www.uku.fi/atkk/plugin/julkaisut/docs/Hautsalo-Hayrinen-Korhonen-2002.pdf> tarkistettu 15.10.2003.
- [Häkkinen, 2003] Heidi Häkkinen, Integroitotarpeiden nopea arviointimenetelmä, *Sosiaali- ja terveydenhuollon tietotekniikan ja tiedonhallinnan tutkimuspäivät 2003-tutkimuspaperi* <http://www.oskenet.fi/uploads/gok8etje.pdf> tarkistettu 20.10.2003.
- [Häkkinen, Turunen ja Saranto, 2002] Heidi Häkkinen, Pekka Turunen ja Kaija Saranto, Hoitoketjujen tiedonkulku - arviointimittariston kokoaminen, *Sosiaali- ja terveydenhuollon tietotekniikan ja tiedonhallinnan tutkimuspäivät 2002-tutkimuspaperi* <http://www.oskenet.fi/uploads/86pwxvzchng.doc> tarkistettu 20.10.2003.
- [Häyrinen ja Saranto, 2003] Kristiina Häyrinen ja Kaija Saranto, Kohti elektronista potilaskertomusta, *Sosiaali- ja terveydenhuollon tietotekniikan ja tiedonhallinnan tutkimuspäivät 2003-tutkimuspaperi* <http://www.oskenet.fi/uploads/gok8etje.pdf> tarkistettu 20.10.2003.
- [Kiviaho, 2002] Kari Kiviaho, Missä paperit?, *Sosiaali- ja terveydenhuollon tietotekniikan ja tiedonhallinnan tutkimuspäivät 2002-tutkimuspaperi* <http://www.oskenet.fi/uploads/86pwxvzchng.doc> tarkistettu 20.10.2003.
- [Kleemola ja Terho-Pellikka, 1998] Kleemola Maija ja Tervo-Pellikka Raija, *Tietosuoja*. Gummerus, Jyväskylä, 1998, 1-145.
- [Koivisto ja Aaltonen, 2003] Juha Koivisto ja Satu Aaltonen, Yhtenäiset käytännöt tietojärjestelmätyössä usein kaikkien etu. Tapaustutkimus kotisairaanhoidosta. *Sosiaali- ja terveydenhuollon tietotekniikan ja*

tiedonhallinnan tutkimuspäivät 2003-tutkimuspaperi

<http://www.oskenet.fi/uploads/gok8etje.pdf> tarkistettu 20.10.2003.

- [Koskinen, 2003] Koskinen Minna, Johdatus tietojärjestelmätieteeseen
http://staff.cs.utu.fi/kurssit/johdatus_tietojarjestelmatieteeseen/syksy2003/14 tarkastettu 10.12.2003
- [Lappalainen, Mäkelä ja Orre, 1997] Marketta Lappalainen, Marjukka Mäkelä, ja Soili Orre, Terveystieteiden erillisrekisterit, Selvitys Suomessa ylläpidettävistä valtakunnallisista ja alueellisista potilasrekistereistä, FinOHTAn raportti 3
<http://www.stakes.fi/finohta/raportit/003/r003f.html> tarkistettu 25.10.2003.
- [Lehenkari, 2003] Janne Lehenkari, Teknologisten innovaatioiden haaste terveydenhuollossa. Teoksessa *Tuotteesta työvälineeksi? Uudet teknologiat terveydenhuollossa*, Reijo Miettinen, Sampsa Hyysalo, Janne Lehenkari ja Mervi Hasu. Gummerus, Saarijärvi, 2003, 143-167.
- [Miettinen *et al.*, 2003] Reijo Miettinen, Sampsa Hyysalo ja Janne Lehenkari, Johtopäätökset. Teoksessa *Tuotteesta työvälineeksi? Uudet teknologiat terveydenhuollossa*, Reijo Miettinen, Sampsa Hyysalo, Janne Lehenkari ja Mervi Hasu. Gummerus, Saarijärvi, 2003, 13-23.
- [Novo, 2003] Pegasos tuote-esitteitä Novo Group 21.3.2003
- [Ripatti, 1999] Sinikka Ripatti Tietojärjestelmien käyttöönotto. Teoksessa *Tietotekniikka ja tiedonhallinta sosiaali- ja terveydenhuollossa*, Kaija Saranto ja Mikko Korpela (toim.). WSOY, Porvoo, 1999, 117-119.
- [Röppänen, 2003] Päivi Röppänen, Terveystieteiden tietojärjestelmät ja niiden integrointi <http://hercules.pspt.fi/PlugIT/opinnaytetyopr.pdf> tarkistettu 20.10.2003.
- [STM, 2002:a] Kansallinen projekti terveydenhuollon turvaamiseksi Julkisen terveydenhuollon, yksityisten ja kolmannen sektorin työnjako ja yhteistyö -osaprojekti KT4
<http://www.vn.fi/stm/suomi/pao/thprojekti/terveys5.pdf> tarkistettu 27.10.2003.
- [STM, 2002:b] Sosiaali- ja terveystieteiden tutkimus 2002, julkaisu 2002:11
<http://www.stm.fi/suomi/pao/sostervkert02/sostervkert02.pdf> tarkistettu 27.10.2003.
- [STM, 1998:a] Sosiaali- ja terveydenhuollon tietoteknologian hyödyntäminen, osa 1, Saumaton hoito- ja palveluketju, Asiakaskortti, työryhmämuistio 1998:8 <http://www.oskenet.fi/uploads/0gyxd5d8rr.pdf> tarkistettu 25.10.2003.

- [STM, 1998:b] Sosiaali- ja terveydenhuollon tietoteknologian hyödyntäminen, Osa 2, Tietosuoja ja tietoturva, työryhmämuistio 1998:9 www.oskenet.fi/uploads/bj2ygh8mt.pdf tarkistettu 25.10.2003.
- [Tampere, 2002] Terveyttä tamperelaisille - yhdessä tehden Perusterveydenhuollon strategia vuoteen 2012 www.tampere.fi/sote/keke/strate_ptn.pdf tarkistettu 28.10.2003.