

Erkki Mäkinen (toim.)

**Pieniä tietojenkäsittely-
tieteellisiä tutkimuksia**

Syksy 2005



TIETOJENKÄSITTELYTIETEIDEN LAITOS
TAMPEREEN YLIOPISTO

D-2005-6

TAMPERE 2005

TAMPEREEN YLIOPISTO
TIETOJENKÄSITTELYTIETEIDEN LAITOS
JULKAISUSARJA D – VERKKOJULKAISUT
D-2005-6, JOULUKUU 2005

Erkki Mäkinen (toim.)

**Pieniä tietojenkäsittelytieteellisiä
tutkimuksia**

Syksy 2005

TIETOJENKÄSITTELYTIETEIDEN LAITOS
33014 TAMPEREEN YLIOPISTO

ISBN 951-44-6541-5
ISSN 1795-4274

Sisällysluettelo

Kaksijakoisen graafin värittäminen.....	1
<i>Timo Helander</i>	
Näkökulmia käytettävyyden alan toimenkuviin.....	15
<i>Jaana Huotari</i>	
Huomioita matkapuhelinten käyttöliittymien suunnittelussa sokeille käyttäjille.....	36
<i>Saija Hälikkä</i>	
ERP-järjestelmien implementoinnin ongelmat.....	53
<i>Antti Jämsen</i>	
Inhimillinen tekijä tietoturvallisuudessa.....	64
<i>Jari Kivelä</i>	
Käytettävyyden rooli verkko-opetuksessa	77
<i>Suvi Leander</i>	
Information Foraging.....	92
<i>Juha Leino</i>	
L-systeemit: Kasvien mallintaminen.....	118
<i>Juha Majanen</i>	
Käsinkosketeltavien käyttöliittymien luokittelusta.....	129
<i>Antti Nyman</i>	
Sukupuoliroolit tietotekniikan alalla Suomessa.....	145
<i>Anne Pyykönen</i>	
Tietohallintostrategian kehitys 2000-luvulle	157
<i>Antti Rikkinen</i>	
Tietojärjestelmästrategian merkitys pk-yrityksille.....	174
<i>Eerik Savolainen</i>	
Kotikäyttäjän tietoturva.....	187
<i>Atte Sepponen</i>	
Selainsovelman ja palvelimen välinen viestintä.....	216
<i>Bemmu Sepponen</i>	
Tietotekniikan hyödyntäminen pienten pirkanmaalaisten hoiva-alan yritysten liiketoiminnassa.....	230
<i>Janne Siltanen</i>	
Tasoverkon duaalin etsintäalgoritmi.....	245
<i>Harri Smått</i>	
Ohjelmistopatenteista.....	260
<i>Sakari Tamminen</i>	
Monikerroksisten neuroverkkojen opetusmenetelmiä.....	279
<i>Paavo Toivanen</i>	

Kaksijakoisen graafin värittäminen

Timo Helander

Tiivistelmä.

Tässä tutkielmassa keskitytään kaksijakoisen graafin solmujen ja särmien värittämiseen sekä muutama graafialgoritmiin, jota tarvitaan värittämiseen.

Avainsanat ja -sanonnat: Kaksijakoinen graafi, värittäminen.

CR-luokat: G.2.2

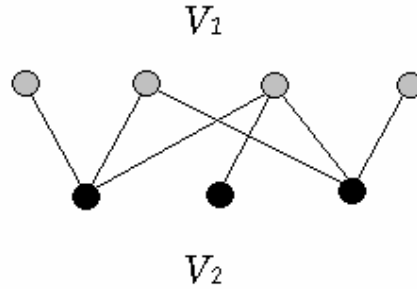
1. Johdanto

Tässä tutkielmassa käsitellään kaksijakoisen graafin tunnistamiseen tarvittavia keinoja sekä solmujen ja särmien värittämistä. Sisällöllisesti käydään läpi muutamia algoritmeja ja näytetään esimerkkikuvoin, miten kyseisessä tapauksessa algoritmi toimisi käytännössä. Kaksijakoisia graafeja käytetään monissa käytännön algoritmeissa.

Kaksijakoiset graafit ovat graafien erikoistapauksia. Graafi $G = (V, E)$ on kaksijakoinen, jos solmujoukko V voidaan jakaa kahteen alkiovieraaseen joukkoon V_1 ja V_2 siten, että $(u, v) \in E$, missä joko $u \in V_1$ ja $v \in V_2$ tai $u \in V_2$ ja $v \in V_1$, ja $V_1 \cup V_2 = V$ ja $V_1 \cap V_2 = \emptyset$ [Cormen et al., 2001]. Tällainen graafi on helppoa esittää piirroksena, jossa solmujoukot V_1 ja V_2 voidaan piirtää kahdelle erilliselle samansuuntaiselle suoralle ja kaikki graafin särmät kulkevat suorien välillä. Kuva 1 esittää tätä havainnollistamista. Kaksijakoisen graafin piirtämisestä tasoon mahdollisimman vähillä särmien leikkauksilla on tutkittu paljon, mutta tässä tutkielmassa keskitytään kaksijakoisen graafin värittämiseen.

Jos graafin solmut pystytään värittämään kahdella eri värillä siten, että minkään särmän eri päissä ei ole samaa väriä, niin se on kaksijakoinen. Kuvassa 1 joukon V_1 solmut on väritetty harmaalla ja joukon V_2 solmut mustalla.

Vielä voidaan todeta, että kaikki puut ovat kaksijakoisia. Tullaan huomamaan, että virittävällä puulla on merkitystä graafin kaksijakoisuuden tunnistamisessa.



Kuva 1. Kaksijakoinen graafi piirrettynä tasoon.

2. Kaksijakoisen graafin tunnistaminen

Kaksijakoisen graafin tunnistamiseen voidaan käyttää monia eri tapoja. Yksinkertaisimpia niistä ovat solmujen värittäminen kahdella eri värillä ja parittomien syklien pois sulkeminen. Molemmat näistä menetelmistä on toteutettavissa syvyys ensin -etsinnällä. Periaate on molemmissa menetelmissä sama. Ensiksi tehdään syvyysuuntainen etsintä, jonka jälkeen tarkastellaan paluusärmää.

2.1. Syvyysuuntainen etsintä

Syvyysuuntaisessa etsinnässä, nimensä mukaisesti, lähdetään etsimään graafin solmuja niin syvältä kuin mahdollista. Solmut merkitään löydetyksi heti, kun uuteen solmuun saavutaan. Kun kaikki aloitussolmusta saavutettavissa olevat solmut on käyty läpi, ja jos vielä on saavuttamattomia solmuja, aloitetaan etsintä uudelleen jostain vielä saavuttamattomasta solmusta ja merkitään se uudeksi aloitussolmuksi. Syvyysuuntainen etsintä graafissa tuottaa syvyys ensin -virittävän puun (DFS-tree), jäljempänä vain virittävä puu. Mikäli etsintää joudutaan suorittamaan useammin kuin kerran, tuloksena on virittävä metsä. Kaksijakoisen graafin määrittäminen tapahtuu syvyysuuntaisen etsinnän tuottaman virittävän puun analysoinnilla.

Syvyys ensin -algoritmi mahdollistaa myös löytämisajan merkitsemisen solmukohtaisesti, mikä on tässä jätetty pois, koska se ei ole tarpeen kyseisessä kaksijakoisen graafin tunnistamistavassa.

Algoritmi 1 esittää pseudokoodina, miten syvyysuuntainen etsintä tapahtuu. Ensiksi alustetaan graafi merkitsemällä solmut valkoisiksi ja vielä löytämättömiksi. Alustamisen jälkeen lähdetään käymään läpi solmuja, ja aina kun törmätään valkoiseen solmuun, kutsutaan funktiota $\text{DFS-Visit}(u)$.

```

DFS(G)
1  for each vertex  $u \in V[G]$ 
2      do  $color[u] \leftarrow WHITE$ 
3           $\pi[u] \leftarrow NIL$ 
4  for each vertex  $u \in V[G]$ 
5      do if  $color[u] = WHITE$ 
6          then DFS-Visit( $u$ )

```

```

DFS-Visit( $u$ )
1   $color[u] \leftarrow GRAY$ 
2  for each  $v \in Adj[u]$ 
3      do if  $color[v] = WHITE$ 
4          then  $\pi[v] \leftarrow u$ 
5              DFS-Visit( $v$ )
6   $color[u] \leftarrow BLACK$ 

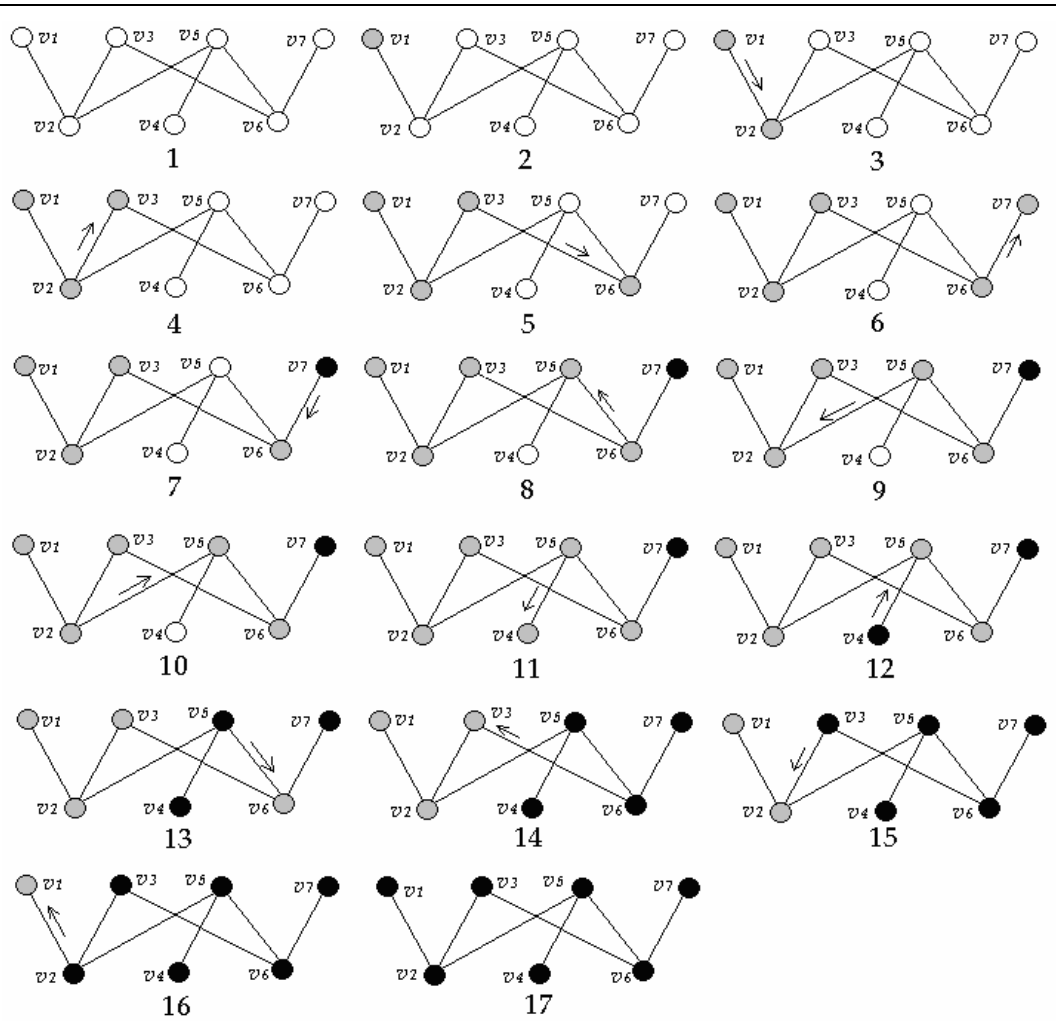
```

Algoritmi 1: Pseudokoodi esitys syvyys ensin etsinnästä [Cormen et al., 2001].

DFS-Visit(u) solmussa u alustaa solmun ensiksi harmaaksi, jonka jälkeen lähde-tään etsimään seuraavaa valkoista solmua solmun u särmien kautta. Kun löyde-tään taas uusi valkoinen solmu v , merkitään solmu u löydetyksi ja kutsutaan rekursiivisesti DFS-Visit(v). Rekursiosta palatessa maalataan solmut mustiksi merkitsemään solmussa tapahtuvan etsinnän lopettamista lopullisesti.

Kuva 2 esittää Kuvan 1 graafin syvyys-suuntaisen etsinnän. Kuvassa on käytetty nimenomaan Algoritmia 1. Kuvan kohdassa 9-10 tulee vastaan jo käy-tetty solmu, jolloin graafissa esiintyy sykli ja tämä särmä on takautuva.

Kaikki muut graafin särmät ovat puusärmiä, koska ne muodostavat syvyys-etsinnässä virittävän puun ilman takautuvia särmiä. Suuntaamattoman graa-fin syvyys-suuntaisen etsinnän aikavaatimus on $O(\max(|V|, |E|))$, [Aho et al., 2000].



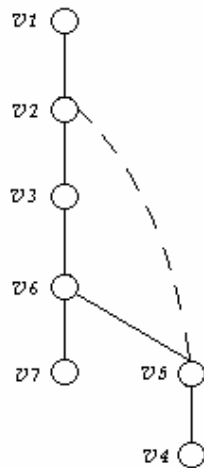
Kuva 2: Syvyysuuntainen etsintä suuntaamattomassa graafissa.

2.2. Virittävä puu

Graafista syvyysetsinnällä muodostettua puuta kutsutaan virittäväksi. Virittävä puu on helppo muodostaa Kuvan 2 syvyysuuntaisen etsinnän jälkeen. Puun juureksi otetaan etsinnän aloitussolmu. Tässä tapauksessa v_1 on puun juuri. Sitten aloitetaan muodostamaan puuta sen mukaan, miten solmut tulevat vastaan etsinnän aikana.

Graafin $G = (V, E)$ etsinnässä muodostuu kahdenlaisia särmiä: puusärmiä ja takautuvia särmiä. Näin ollen voidaan E jakaa kahteen osajoukkoon T ja B , missä T edustaa puusärmiä ja B takautuvia särmiä. Tämän jaon jälkeen on mahdollista muodostaa virittävä puu $S = (V, T)$. Tässä vaiheessa on jo hieman aavistusta, miten voidaan tulkita graafista parilliset tai parittomat syklit. Kuva

3 esittää Kuvan 2 etsinnän jälkeen muodostettua puuta. Takautuvat särmät on piirretty kuvaan katkoviivalla [Aho et al., 1975].



Kuva 3: Syvyysuuntaisella etsinnällä muodostettu virittävä puu.

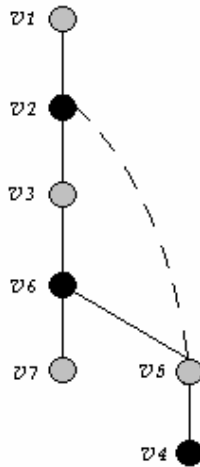
Puun muodostamisessa lähdetään liikkeelle syvyysetsinnän aloittavasta solmusta v_1 , josta tehdään juuri ja sitten asetellaan solmuja sen mukaan aina edellisen lapsiksi, kun niitä tulee vastaan etsinnässä. Etsintä joutuu palaamaan solmussa v_7 solmuun v_6 särmää (v_6, v_7) pitkin, jolloin etsinnän jatkuessa solmuun v_5 tehdään puuhun haara.

2.3. Solmujen värittäminen

Solmujen värittäminen mahdollisimman vähillä väreillä graafissa $G = (V, E)$ tarkoittaa, että solmujoukossa V ei esiinny samalla värillä väritettyjä solmuja vierekkäin. Solmut ovat vierekkäisiä, jos niitä yhdistää särmä. Jos graafi voidaan värittää kahdella värillä ja minkään särmän eri päissä ei esiinny saman värisiä solmuja, niin on se kaksijakoinen.

Virittävän puun väritys kahdella värillä tapahtuu siten, että aloitetaan juuresta jommalla kummalla värillä ja aina seuraavalla puun tasolla vaihdetaan väriä. Parittomilla ja parillisilla tasoilla puussa on eri värit. Kuvassa 4 on Kuvan 3 puu väritettynä.

Takautuvaa särmää kutsutaan huonoksi särmäksi, mikäli sen eri päissä on samalla värillä väritetyt solmut, ja hyväksi takautuvaksi särmäksi, mikäli eri päissä on eri väreillä väritetyt solmut [Damaschke, 2003]. Kuvan 4 puussa ei ole yhtään huonoa särmää, sillä ainoan takautuvan särmän (v_2, v_5) päätepisteet on väritetty eri väreillä.



Kuva 4: Virittävä puu väritettynä.

2.4. Parillinen vai pariton sykli

Graafi on kaksijakoinen, jos se ei sisällä paritonta sykliä. Sykli on pariton, jos ja vain jos se sisältää parittoman määrän huonoja takautuvia särmiä.

Syvyysuuntaisella etsinnällä muodostetussa virittävässä puussa olevaa väritettyä polkua kutsutaan huonoksi, mikäli sen alku- ja loppupäät yhdistää huono särmä. Hyväksi poluksi kutsutaan sellaista polkua, jonka alku- ja loppupäät yhdistää hyvä takautuva särmä. Kuvassa 4 on yksi hyvä polku: $(v_2, v_3), (v_3, v_6), (v_6, v_5)$.

Jos graafin virittävässä puussa ei ole huonoja polkuja, ei graafissa myöskään ole parittomia syklejä.

Seuraavassa esitellään Damaschken [2003] tapaa tunnistaa kaksijakoinen graafi ja yksi särmä. Kaksijakoisen graafin ja lisäksi yhden särmän tunnistaminen voidaan toteuttaa lineaarisessa ajassa.

Merkitään B :llä huonojen polkujen *leikkausta* ja J :llä hyvien polkujen *unionia*. Kaikille huonoille särmillä e määritellään joukko B_e , joka sisältää kaikki huonot polut paitsi polkua, jonka e määrittelee. Samalla tavalla hyvälle takautuvalle särmillä e määritellään J_e , joka on kaikkien hyvien polkujen joukko, lukuun ottamatta polkua, jonka e määrittelee.

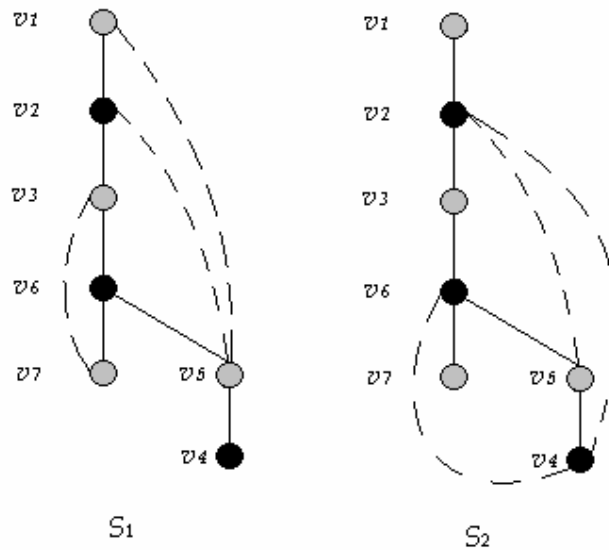
Oletetaan, että graafissa on vähintään kaksi huonoa särmää. Silloin kaikkien parittomien syklien *leikkaus* graafissa G on $B \setminus J$. Tästä seuraa, että ei-kaksijakoinen graafi G on kaksijakoinen graafi ja yksi särmä, jos ja vain jos $B \setminus J \neq \emptyset$.

Todistetaan väite tarkastelemalla särmää e , joka kuuluu jokaiseen parittomaan sykliin. Silloin e on puusärmä B :ssä. Jos e kuuluu johonkin hyvään polkuun, niin silloin voimme muodostaa syklin $G \setminus e$:ssä lisäämällä täsmälleen

yhden huonon särmän. Jos $e \notin J$, niin otetaan mikä tahansa $e \in B \setminus J$. Nyt särmän e poistaminen jakaa virittävän puun kahteen osaan, jotka on yhdistettynä kaikilla huonoilla särmillä, mutta ei yhdelläkään hyvällä särmällä, joten jokaisessa graafin $G \setminus e$ sykliässä on parillinen määrä huonoja särmiiä. Siis kaikki parittomat syklit G :ssä sisältävät särmän e . [Damaschke, 2003]

Tämä pätee siis ainoastaan graafeille, jotka sisältävät vähintään kaksi huonoa särmää.

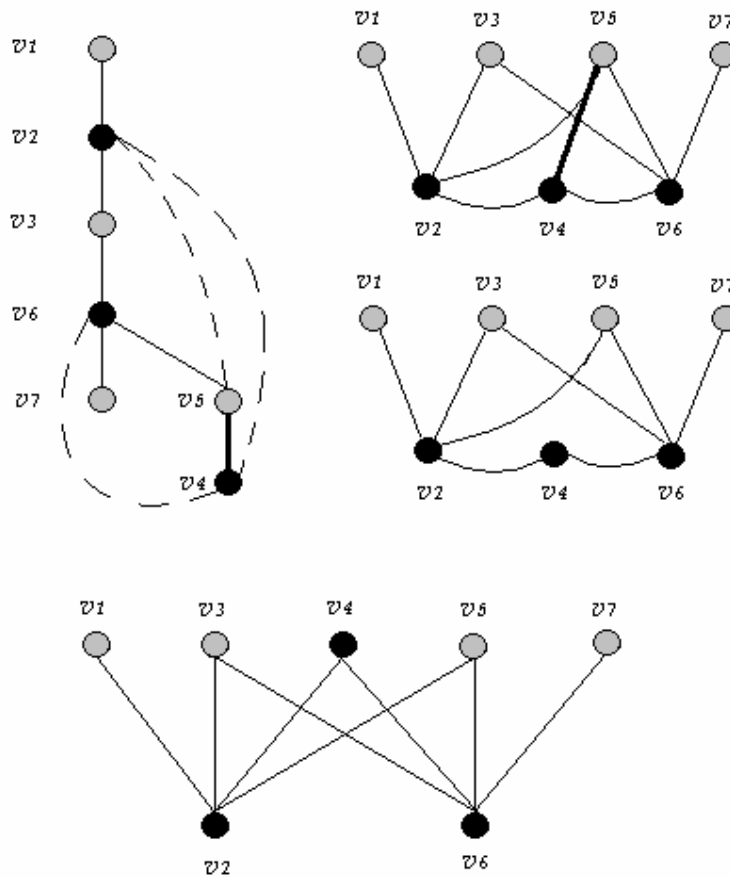
Havainnollistetaan viimeistä määrittelyä kuvalla. Kuvassa 5 on kaksi puuta S_1 ja S_2 . Nyt voimme muodostaa joukot B_1 ja B_2 sekä joukot J_1 ja J_2 seuraavasti: $B_1 = \{(v_3, v_6)\}$, $J_1 = \{(v_2, v_3), (v_3, v_6), (v_6, v_5)\}$, $B_2 = \{(v_6, v_5), (v_5, v_4)\}$ ja $J_2 = \{(v_2, v_3), (v_3, v_6), (v_6, v_7)\}$. Huomataan, että $J_1 = J_2$, joten voimme merkitä $J = J_1$.



Kuva 5: Kuva 4:n puusta tehty kaksi eri versiota.

Oletetaan, että puu S_1 on muodostettu graafista G_1 ja S_2 on muodostettu graafista G_2 . Nyt kaikkien parittomien syklien leikkaus graafissa G_1 on $B_1 \setminus J = \emptyset$ ja kaikkien parittomien syklien leikkaus graafissa G_2 on $B_2 \setminus J = \{(v_5, v_4)\}$. Graafi G_1 sisältää kaksi paritonta sykliä, jotka kuuluvat graafin rakenteeseen tiukasti, mutta graafissa G_2 kaikki parittomat syklit kulkevat särmän (v_5, v_4) kautta. Poistamalla graafista G_2 tämä särmä saadaan graafista kaksijakoinen. Kuva 6 esittää tätä prosessia.

Tämä on yksi tapa, jolla voidaan päätellä jotain graafin kaksijakoisuudesta. Yksinkertaisesti voisi todeta, että graafi on kaksijakoinen, jos ja vain jos se ei sisällä paritonta sykliä. Graafi ei myöskään sisällä paritonta sykliä, jos ja vain jos graafista muodostettava virittävä puu ei sisällä huonoja polkuja. Mikäli graafi sisältää vähintään kaksi paritonta sykliä, voidaan graafista ehkä saada kaksijakoinen poistamalla yksi särmä.

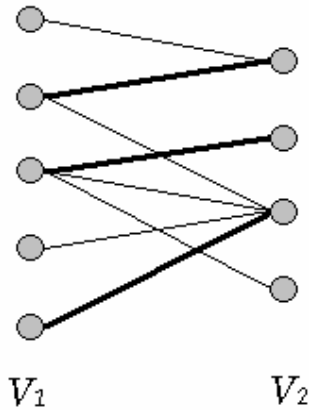


Kuva 6: Poistamalla parittomien syklien leikkaus, graafista tulee kaksijakoinen.

3. Maksimaalisesta sovituksesta

3.1. Sovituksesta ja maksimaalinen sovitus

Kaksijakoisille graafeille on helppo keksiä sovellusalueita. Yksi perussovellus liittyy tehtävien jakamiseen työntekijöille (ks. [Aho et al., 1983]). Tarkastellaan joukkoa opettajia ja joukkoa kursseja. Kukin opettajista on pätevä opettamaan osaa kursseista, mutta ei kaikkia. Ongelmana on asettaa opettajat opettamaan kursseja siten, että jokaisella kurssilla on pätevä opettaja. Jos on olemassa sellaisia kurssien ja pätevien opettajien yhdistelmiä, että on mahdotonta asettaa jokaiselle opettajalle kurssia, niin silloin halutaan käyttää niin montaa opettajaa kuin mahdollista. Kuvassa 7 solmut on jaettuna kahteen joukkoon V_1 ja V_2 siten, että solmut joukossa V_1 esittävät opettajia ja solmut joukossa V_2 kursseja. Opettajaa $v \in V_1$, joka valitaan opettamaan kurssia $w \in V_2$, kuvataan särmällä (v, w) .



Kuva 7: Opettajat ja kurssit, sekä maksimaalinen sovitus.

Graafin sovitus tarkoittaa sellaisten särmien joukkoa, että yhteen solmuun ei tule kahta tähän sovitukseen kuuluvaa särmää. Eli sovitus on sellainen osajoukko särmiä $M \subseteq E$ graafissa $G = (V, E)$, että kaikille $v \in V$ on ainoastaan yksi tuleva särmä M :ssä. Sanotaan, että solmu $v \in V$ on sovitettu sovituksella M , mikäli jokin särmä M :ssä on kiinni v :ssä, muuten v on ei sovitettu. Maksimaaliseksi sovitukseksi M sanotaan sellaista sovitusta, jossa mille tahansa sovitukselle M' pätee, että $|M| \geq |M'|$ [Cormen et al., 2001].

Kuvassa 7 on graafiin piirrettyä maksimaalinen sovitus paksummilla särmillä.

Maksimaalisen sovituksen löytämiseen kaksijakoisesta graafista voi käyttää esimerkiksi *täydennyspolun* löytävää algoritmia. Täydennyspolku graafissa on sellainen, jossa joka toinen särmä kuuluu sovitukseen ja joka toinen ei [Aho et al., 1983]. Täydennyspolku määritellään seuraavasti: $G = (V, E)$, missä $V = V_1 \cup V_2$. Sanotaan, että sovitus M graafissa G on $M \subseteq E$. Polku P graafissa G on täydennyspolku M :n suhteen, jos se alkaa sovittamattomasta solmusta V_1 :ssä, päättyy sovittamattomaan solmuun V_2 :ssä ja polulla esiintyvät särmät kuuluvat vuorotellen, joko M :ään tai $E \setminus M$:ään.

3.2. Särmien värittäminen ja maksimaalinen sovitus

Graafin särmät voidaan myöskin värittää kuten solmut. Periaate on sama eli vierekkäiset särmät eivät jaa samaa väriä. Vierekkäiset särmät ovat särmiä, jotka jakavat saman solmun. Kaksijakoisen verkon särmien värittäminen ei kuitenkaan onnistu aina kahdella värillä. Jos verkko on kaksijakoinen, niin verkon särmien värittämiseen tarvittavien värien lukumäärä on sama kuin graafin aste, jolla tarkoitetaan suurinta graafista löytyvän solmun astetta.

Maksimaalinen sovitus graafissa voidaan löytää myös särmiä värittämällä. Maksimaalisella sovituksella ja graafin särmien värittämisellä on erittäin paljon

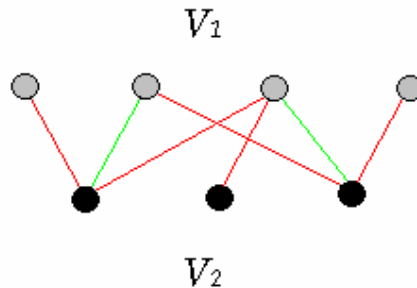
tekemistä toistensa kanssa. Voidaan sanoa, että maksimaaliseen sovitukseen kuuluvat särmät ovat samanvärisiä.

Seuraavassa esitellään ajassa $O(\Delta m)$ toimiva metodi värittämiseen. Todetakaan, että se on ekvivalentti täydellisen sovituksen löytämiseen graafista $O(km)$ ajassa [Schrijver, 1998]. Tässä Δ ja k tarkoittavat graafin astetta. Täydellinen sovitus graafissa on sellainen, jossa jokainen solmu on sovitettuna, mutta yksikään sovitukseen kuuluva särmä ei jaa samaa solmua. Täten ei ole kovinkaan vaikea kuvitella särmien värittämisen olevan ekvivalentti täydellisen sovituksen kanssa.

3.3. Särmien värittäminen Δ -värillä $O(\Delta m)$ ajassa

König on osoittanut, että kaksijakoinen graafi voidaan värittää Δ -värillä, missä Δ on graafin solmun suurin aste, ja se voidaan tehdä $O(nm)$ ajassa, missä n on solmujen ja m särmien lukumäärä.

Särmien väritys alkaa siitä, että etsitään sovitus, johon kuuluvat kaikki suurinta astetta olevat solmut, ja tähän sovitukseen kuuluvat särmät väritetään ensimmäisellä värillä. Kuvassa 8 on nyt yksi tällainen mahdollinen sovitus tehtynä Kuvan 1 graafiin.



Kuva 8: Suurinta astetta olevien solmujen sovitus.

Särmien värittäminen kaksijakoisessa graafissa voidaan tehdä samassa ajassa kuin k -säännöllisessä graafissa. Sen takia on mahdollista toteuttaa särmien väritys mihin tahansa kaksijakoiseen graafin Δ -värillä.

Lause 1 [Schrijver, 1998]. *Täydellisen sovituksen löytäminen k -säännölliseen kaksijakoiseen graafiin voidaan toteuttaa $O(km)$ ajassa.*

Todistus. Olkoon $G = (V, E)$ k -säännöllinen kaksijakoinen graafi. Mille tahansa funktiolle $w: E \rightarrow \mathbb{Z}_+$, olkoon E_w joukko särmä, joilla $w(e) > 0$, eli jotka eivät kuvaudu nolalle kuvauksessa w . Kaikille $F \subseteq E$ merkitään $w(F) := \sum_{e \in F} w(e)$. Olkoon χ^F kytkentävektori F :lle. Kytkentävektorilla kuvataan yhtä sellaista osajoukkoa, joka on sovitus joukossa E .

Aluksi asetetaan $w(e) := 1$ kaikille särmille e . Sitten suoritetaan seuraavaa iteratiivisesti:

- (2) Etsi sykli $C \subseteq E_w$. Asetetaan $C = M \cup N$, missä $M \geq N$. Sitten asetetaan uudelleen $w := w + \chi^M - \chi^N$.

Jokaisella iteraatiolla yhtäsuuruutta $w(\delta(v))$ pidetään yllä kaikille $v \in V$, missä $\delta(v)$ on solmulle v vierekkäiset särmät.

Lopetusehdon täyttämiseen tarvitaan ensin summa (3),

$$(3) \quad \sum_{e \in E} w(e)^2$$

Summaa (3) lisätään summalla (4).

$$(4) \quad \sum_{e \in M} ((w(e)+1)^2 - w(e)^2) + \sum_{e \in N} ((w(e)-1)^2 - w(e)^2) \\ = 2w(M) + |M| - 2w(N) + |N|$$

Tämä on vähintään $|C|$, koska $w(M) \geq w(N)$. Sitä paitsi (3) on rajoitettu, koska $w(e) \leq k$ jokaiselle särmälle e . Siis prosessi tulee loppumaan.

Lopussa meillä on joukko E_w , joka on metsä ja täydellinen sovitus. Tämä merkitsee, että summan (3) lopetus on yhtä suuri kuin $\frac{1}{2}nk^2 = km$. \square

Syvyysuuntaisella etsinnällä voidaan löytää sykli C kohdasta (2) keskimäärin $O(|C|)$ ajassa. Edellä kuvailtiin, miten tämä onnistuu.

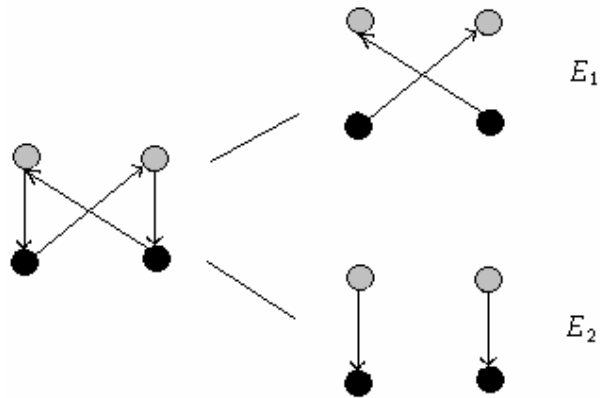
Nyt voidaan käyttää tätä metodologia graafin väritykseen. Seuraavassa todistuksessa käytettävä Eulerin orientaatio graafissa $G = (V, E)$ tarkoittaa sellaista särmille asetettavaa suuntaa, että jokaisella solmulla $v \in V$, saapuvien ja lähtevien kaarien lukumäärä on sama [Fehrenbach and Rüschemdorf, 2004].

Lause 2 [Schrijver, 1998]. *k*-särmän väritys *k*-säännöllisessä kaksijakoisessa graafissa voidaan toteuttaa $O(km)$ ajassa.

Todistus. Jos *k* on pariton, niin etsitään täydellinen sovitus *M* ja poistetaan *M* graafista *G*. Sitten tehdään sama rekursiivisesti jäljelle jääneelle graafille.

Jos *k* on parillinen, niin etsi Eulerin orientaatio graafista *G*. Asetetaan $k' = \frac{1}{2}k$. Sitten jaetaan *G* kahteen k' -säännölliseen graafiin G_1 ja G_2 , missä $G_1 = (V, E_1)$ ja $G_2 = (V, E_2)$. E_1 on särmien joukko solmun väri luokasta I luokkaan II ja $E_2 = E \setminus E_1$. Kuvassa 8 on esitetty, miten tällainen jako voisi tapahtua. Tästä edetään rekursiivisesti graafeihin G_1 ja G_2 . \square

Tässä algoritmossa aina täydellisen sovituksen löytymisen yhteydessä särmät saavat värinsä.

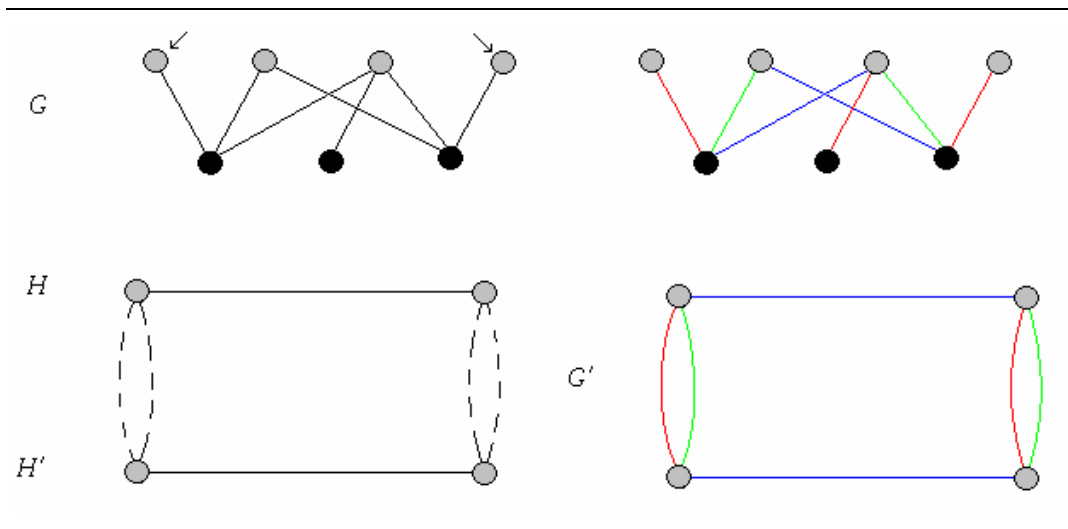


Kuva 9. Eulerin orientaatiosta jako kahteen eri graafiin.

Lause 3 [Schrijver, 1998]. Δ -värillä särmien väritys kaksijakoisessa graafissa $G = (V, E)$ voidaan toteuttaa $O(\Delta m)$ ajassa.

Todistus. Asetetaan $k = \Delta$. Ensiksi iteratiivisesti haetaan kaksi solmua samasta solmuluokasta G :ssä. Näillä haetuilla solmuilla täytyy olla astelukuna enintään $\frac{1}{2}k$. Sen jälkeen ne yhdistetään, jolloin siitä muodostuva graafi H tulee sisältämään enintään kaksi solmua, jotka ovat enintään astetta $\frac{1}{2}k$. Edelleen asetetaan $\Delta = k$ graafissa H ja jokainen k -väritys tuottaa k -värityksen G :ssä. Seuraavaksi tehdään kopio H' graafista H ja kaikista $v \in H$ yhdistetään $k - d_H(v)$ rinnakkaisilla särmillä v ja v' kopio H :ssa, tässä $d_H(v)$ tarkoittaa solmun v astetta graafissa H . Kuvassa 10 on vasemmalla ylhäällä tuttu graafi. Tästä on muodostettu nuolella valituista solmuista H ja siitä vielä H' , mistä on sitten muodostettu G' ja vielä huomataan, että koska G' on k -säännöllinen niin se on väritettävissä $O(k|E(G)|)$ ajassa Lauseen 2 mukaan. Tällöin on kaikki värit koossa mitä tarvitaan kaksijakoisen graafin G värittämiseen. G näkyy väritettynä kuvassa oikealla ylhäällä. \square

Tämän aliotsikon lauseet ja analyysit, sekä myöskin kaikki melkein muu paitsi kuvat löytyvät artikkelista [Schrijver, 1998].



Kuva 10. Graafin värityksen vaiheet.

4. Yhteenveto

Kaksijakoisille graafeille on helppo löytää sovellusalueita, koska monet asiat pystytään kuvaamaan relaatioilla kahden joukon välillä. Kaksijakoisen graafin tunnistaminen $O(m + n)$ ajassa on jo kansanperinnettä [Damaschke, 2003; Aho et al., 1975].

Kaksijakoisen graafin särmien väritykselle voidaan myös löytää erilaisia sovellusalueita. Yksi sellainen liittyy esimerkiksi tehtävien jakoon eri prosessoreille moni prosessorillisissa ympäristöissä.

Aikakompleksisuutta voidaan hahmottaa koko prosessille yhdistämällä nämä kaksi väritystekniikka, jolloin se on kaksijakoisella graafille todennäköisesti lähellä $O(n + m) + O(\Delta m)$, koska solmujen värittäminen onnistuu ajassa $O(n + m)$ [Damaschke, 2003; Aho et al., 1975] ja särmien värittäminen ajassa $O(\Delta m)$ [Schrijver, 1998].

Viiteluettelo

[Aho et al., 1975] Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1975.

[Aho et al., 1983] Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, *Data Structures and Algorithms*. Addison-Wesley, 1983.

[Aho et al., 2000] Isto Aho, Erkki Mäkinen and Timo Poranen, *Algoritmien suunnittelu ja analyysi*. Tampereen Yliopisto, Tietojenkäsittelytieteiden laitos, Raportti C-2000-1, Tammikuu 2000.

- [Cormen et al., 2001] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms*. 2nd Edition. MIT Press, 2001.
- [Damaschke, 2003] Peter Damaschke, Linear-time recognition of bipartite graphs plus two edges. *Discrete Mathematics* **262** (2003), 99-112.
- [Fehrenbach and Rüschemdorf, 2004] Johannes Fehrenbach and Ludger Rüschemdorf, A Markov chain algorithm for Eulerian orientations of planar triangular graphs. University of Freiburg, Department of Mathematics, Preprint 2004. Available as <http://www.stochastik.uni-freiburg.de/~rueschemdorf/papers/MarkovChainAlgorithm.ps>.
- [Schrijver, 1998] Alexander Schrijver, Bipartite edge coloring in $O(\Delta m)$ time. *SIAM Journal on Computing*. **3** (1998), 841-846.

Näkökulmia käytettävyyden alan toimenkuviin

Jaana Huotari

Tiivistelmä

Tämä kirjallisuuskartoitus tutkii, miten ja mistä lähtökohdista sekä millaisin tuloksin käytettävyyden alan toimenkuvatutkimusta on tehty Suomessa ja ulkomailla ja miten toimenkuvatutkimusta voidaan kehittää tästä eteenpäin. Toimenkuvatutkimus ja käytettävyys käsitteenä määrittävät tutkimuskohdetta ja alan toimenkuvatutkimuksen puutteellisuus Suomessa puolestaan vaikeuttaa Suomea koskevien johtopäätösten tekemistä.

Toimenkuvia tutkitaan ottamalla huomioon useita tekijöitä, jotka riippuvat myös tutkimuksen kohteena olevasta alasta. Käytettävyyden toimenkuviin vaikuttavat uraan yleisesti vaikuttavat tekijät, työntekijän asema organisaatiossa, työssä käytetyt menetit ja työntekijän oma osaamistausta. Myös kulttuurilla ja sukupuolella on omat vaikutuksensa. Toimenkuvatutkimuksessa kysymyksenasettelulla ja tutkimuksen toteutuksella on suuri merkitys myös siihen, millaisia tuloksia saadaan.

Avainsanat ja -sanonnat: käytettävyys, käyttäjälähtöinen suunnittelu, käytettävyyden alan työnkuva, toimenkuva, toimenkuvatutkimus

CR-luokat: K.7.1

1. Johdanto

Tämän kirjallisuuskartoituksen tarkoituksena on hahmottaa käytettävyyden alan ammattilaisten työnäkymiä ja toimenkuvia työnteon näkökulmasta katsottuna. Vastaavaa toimenkuvatutkimusta on tehty hyvin vähän Suomessa, mutta ulkomaisia tutkimuksia aiheesta on olemassa useita. Ulkomaiset esimerkit pyrkivät kuitenkin kuvaamaan kyseisiä toimenkuvia hyvin yleisesti. Aluksi on syytä tutkia käytettävyyden alaa myös osana IT-alaa ja selvittää, mitä käytettävyyden työ tarkalleen sisältää, minkä jälkeen voidaan tarkastella tarkemmin esimerkkejä käytettävyyden alan toimenkuvatutkimuksesta. Lisäksi alalle hakeutuvat opiskelijat kertovat osaltaan työnäkymistä ja alaan suhtautumisesta.

Toimenkuvatutkimus sinänsä on ongelmallinen tutkimuskohde monien tutkimukseen vaikuttavien tekijöidensä vuoksi ja käytettävyyden alan yleinen hajanaisuus vaikeuttaa tutkimuskohteen määrittelyä. Olen kerännyt tähän kartoitukseen mukaan niin toimenkuvananimikkeitä, työssä käytettyjä metodeja,

tarkkaa erittelyä työntekijöiden koulutustaustasta, työkokemuksesta, sukupuoli- ja ikärakenteesta kuin myös yleisestä suhtautumisesta IT-alaan ja käytettävyyteen. Käytettävyyttä tulee tarkastella myös osana organisaation toimintaa, koska se määrittää suuresti toimenkuvien muotoutumista. Käytettävyyden toimenkuviiin voidaan siis liittää useita vaikuttavia tekijöitä, niin alan ulkoisia kuin sisäisiäkin. Käytettävyyden alan ammattilaisen täytyy ottaa huomioon työssään useita vaatimuksia, mikä vaikuttaa yhdessä ammattilaisen oman osaamistaustan kanssa toimenkuvan muodostumiseen.

Luku 2 käsittelee IT-alan yleisiä työnäkymiä ja alalle suuntautuvien suhtautumista tulevaisuuteen. Lisäksi luvussa tutkitaan, onko sukupuolella vaikutusta toimenkuvien rakentumiseen. Luvussa 3 tutkitaan tarkemmin käytettävyyttä ammattina ja tarkastellaan toimenkuvien sisältöä. Luku tarkastelee ammattia organisaation osana sisältäen niin työtehtävien määrittelyä kuin myös työnimikkeiden kuvausta. Luvussa 4 käsitellään lopuksi paikallisesta näkökulmasta alaan liittyvää toimenkuvatutkimusta.

2. IT-alan yleisiä työnäkymiä

Yleisesti katsottuna kaikki teknologiaan ja tietojenkäsittelyyn liittyvät alat työllistävät paljon ihmisiä suhteutettuna maan kokoon. IT-ala (information technology) on myös kasvava ala ja siksi onkin syytä tarkemmin tutkia alan rakenteita, jotta näemme, mihin suuntaan ala on kehittymässä. Pahin teknologiakupla lienee onneksi puhjennut ja sitä kautta tilanne hieman normalisoitunut. Täytyy myös muistaa, että kunkin maan yleinen työllistymistilanne vaikuttaa IT-alaan samalla tavalla kuin muihinkin aloihin. Alan sisäisten ja ulkoisten paineiden lisäksi alan kasvuun ja kehittymiseen vaikuttavat myös koulutettujen ihmisten määrä ja osaamisen taso.

IT-alaa pidetään yhtenäisenä alana, vaikka se koostuu useista hyvin erilaisista toimenkuvista. Myös muut alat kuin suoraan teknologiaan ja tietojenkäsittelyyn liittyvät alat sisältävät IT-osaajia ja siksi IT-ala kokonaisuutena on hyvin hajanainen. IT-alan koulutuksen saaneet asettuvat usein työelämään mielenkiinnon, taitojen ja työkokemuksen mukaan [Kaarst-Brown and Guzman, 2005]. Tarvittavat taidot onkin usein hankittu varsinaisen alan koulutuksen ulkopuolelta. Koulutusohjelmien suunnittelussa ja toteutuksessa tulisi-kin suosia tyypiltään sellaisia koulutusohjelmia, jotka tukevat paremmin jokaisen oppilaan omia mielenkiinnon kohteita. [Kaarst-Brown and Guzman, 2005]

IT-alan toimenkuvia voidaan tarkastella vastuualueiden, tehtäväkentän ja työskentelytapojen avulla [Haapakorpi, 2000]. Toinen tapa opetushallituksen mallin mukaan on määrittellä toimenkuvaa siten, että se jaetaan vielä useampiin tekijöihin. Toimenkuvalle liitetään mallin mukaan ensin työministeriön luoki-

tus ja määritetään siihen liittyvät tehtävänimikkeet. Toimenkuva myös kuvataan yleisesti päätehtävineen ja sille hahmotellaan rekrytointikriteerit sekä määritellään, mitä kuuluu ammatin hyvään hallintaan ja vaadittaviin tietoihin ja taitoihin. Tämän jälkeen kuvataan vielä toimenkuvan yleiset lähitulevaisuuden näkymät sekä uranäkymien ja -polkujen suunnat. [Pelkonen, 1999]

Toimenkuvien tarkastelussa täytyy siis ottaa huomioon useita eri tekijöitä. Yhdysvaltojen väestönlaskentatoimisto on käyttänyt vuodesta 2003 lähtien kahdeksaa työkategoriata kerätessään tietoa teknologian parissa työskentelevistä henkilöistä [Chabrow, 2003]. Nämä kategoriat ovat ohjelmistokehittäjät (software engineers), tietokonetieteilijät (computer scientists) ja systeemisuunnittelijat (systems analysts), ohjelmoijat (programmers), verkko- ja dataliikennesuunnittelijat (network systems and data communications analysts), tietokone- ja tietojärjestelmäjohtajat (computer and information systems managers), tukiasiantuntijat (support specialists), verkko- ja tietojärjestelmäylläpitäjät (network and computer systems administrators) ja tietokantahallinnoijat (database administrators) [Chabrow, 2003].

Suurin ongelma tässä jaottelussa on se, ettei kaikkia IT-alan toimenkuvia voida jakaa yksiselitteisesti näihin tiettyihin kategorioihin. Tarkasteltaessa varsinkin käytettävyyteen liittyviä toimenkuvia, on toimenkuvia hyvin vaikea sisällyttää kyseisiin kategorioihin. Työtehtävistä riippuen käytettävyys voidaan nähdä osana järjestelmän suunnittelu- ja kehitystyötä tai tukityönä, jossa toteutuksessa autetaan ottamaan huomioon käyttäjän näkökulma.

2.1. Alalle suuntautuvien suhtautuminen tulevaisuuteen

Tutkittaessa IT-alan opiskelijoiden suhtautumista työmahdollisuuksiin, voidaan nähdä useita vaikuttavia tekijöitä. Etelä-Yhdysvalloissa [Gardiner et al., 2004] keskikokoisissa yliopistoissa tehty tutkimus toi esiin näitä tekijöitä selvittämällä opiskelijoiden suhtautumista IT-alan työmarkkinoihin. Yliopistojen viimeisen vuosikurssin IT-alan tutkintolinjan 86 opiskelijalta kysyttiin heidän ammatillista sitoutumistaan ja havaittiin, että useat tekijät vaikuttavat tähän sitoutumiseen. Tutkimuksen mukaan IT-alan työmarkkinat, työnsaannin epävarmuuden havaitseminen, asenteet IT-alaa kohtaan ja opiskelijan koulutusohjelma vaikuttavat kukin osaltaan urasitoutumiseen. [Gardiner et al., 2004]

Kyseisen tutkimuksen tulokset tukivat hypoteesia, että mitä enemmän opiskelija luottaa taitoihinsa ja mitä enemmän hänellä on työkokemusta IT-alasta, sitä sitoutuneempi hän on uraansa. Lisäksi opiskelijan positiiviset kokemukset opiskelun alussa vaikuttavat tulevaan urasitoutumiseen, sekä myöhemmät haasteelliset ja palkitsevat opiskelukokemukset vahvistavat myös urasitoutumista. Tutkimustulokset eivät kuitenkaan tukeneet asetettua hypo-

teesia siitä, että opiskelijan epävarmuus alasta vähentäisi urasitoutumista. Sukupuolella, valittujen kurssien tyypeillä ja jäljellä olevalla ajalla valmistumiseen ei ollut vaikutusta urasitoutumiseen. Myöskään eroja sitoutumisessa erityylisten koulutusohjelmien välillä ei löytynyt. Tutkimuksen mukaan oppilailta oli yleisesti syvä sitoutuminen uraan ja erityisesti luottamuksella omiin taitoihin oli suuri vaikutus sitoutumiseen. Kuitenkin tutkimuksen mukaan naiset eivät ole niin luottavaisia omiin taitoihinsa kuin tutkimukseen vastanneet miehet. [Gardiner et al., 2004]

Tutkijat myöntävät, että kyseessä on alustava tutkimus ja he ovat sitä mieltä, että tutkimus täytyy tehdä uudelleen laajemmassa muodossa. Tutkimukseen tulisi siis ottaa mukaan myös muita kuin viimeisen vuoden opiskelijoita ja laajentaa tutkimus muihin yliopistoihin. Tulevaisuudessa tulisi tutkia myös opiskelun lopettaneiden syitä. [Gardiner et al., 2004] Tutkimus tukee opiskeluajan merkityksen tärkeyttä työelämään sijoittumisessa. Opiskeluaika vaikuttaa niin työelämään hakeutumiseen kuin työtehtävissä oman tietämyksen luottamiseen. Tietenkin luottamus kasvaa kokemuksen myötä ja työkokemus opiskeluaikana voi myös vaikuttaa positiivisesti, vaikka yleistä kritiikkiä Suomessakin on esitetty työssäkäyviä opiskelijoita kohtaan.

Opiskelijoiden urasuhtautumiseen vaikuttavat myös alan yleiset kasvunäkymät ja mahdollinen työntekijöiden ulkoistaminen. Paitsi että opiskelijat pystyvät saamaan tietoa työmarkkinoista opiskelujensa aikana, on myös medialla suuri vaikutus tämän kuvan luomisessa. Vaikka tutkimus ei tue väitettä, että epävarma suhtautuminen vähentäisi sitoutumista, niin voidaan kysyä, kuinka hyvin se on tämän tutkimuksen avulla saatu selville. Opiskelijat eivät varmasti mielellään myönnä, että suhtautuvat epävarmasti valitsemansa alan työnäkyelmiin. IT-alan uutisointi vaikuttaa opiskelijoiden suhtautumiseen ainakin jossain määrin ja vaikutus on vielä suurempi alalle hakevien keskuudessa.

Tarkasteltaessa vielä tulosta naisten pienemmästä luottamuksesta omiin taitoihin suhteessa miehiin, voi eron osittain selittää IT-alan miesenemmistöllä. Vaikka todelliset taidot eivät olisi yhtään huonommat suhteessa toiseen sukupuoleen, voi opiskelija vähemmistön edustajana tuntea epävarmuutta omia taitojaan kohtaan. Varsinkin jos vähemmistösukupuolen osallistumista alalle ei tueta tarpeeksi opiskeluympäristössä ja yleisessä keskustelussa.

Miten sitten kulttuuriset tekijät voisivat vaikuttaa näkemykseen tulevasta työurasta? Vuodelta 2003 oleva tutkimus [Seymour et al., 2004], joka käsittelee 497 eteläafrikkalaisen opiskelijan mielipiteitä mahdollisesta tulevasta työurasta IT-alalla, valottaa edellistä kysymystä. Opiskelijoilta kysyttiin heidän aikomuksiaan opiskella tietojenkäsittelytieteitä ja heidän silloista tietämystään alasta,

sekä lisäksi mielipiteitä mahdollisesta palkasta ja työsaannista. [Seymour et al., 2004]

Tutkimuksen mukaan ne opiskelijat, joilla ei ollut tietokonetta tai internetiä käytettävissä, aikoivat opiskella enemmän juuri tietojenkäsittelyä, ja mikä oli huomattavaa, erityisesti näin oli naisten kohdalla. Kyseisillä opiskelijoilla oli myös optimistisemmat näkemykset työmahdollisuuksista ja aloituspalkoista, ja samaan tapaan ajattelivat myös naisopiskelijat. Tutkittaessa eri koulutusohjelmien vaikutusta esitettyihin kysymyksiin, huomattiin, että tietojenkäsittelytiede oli huomattavasti suositumpaa kuin tietojärjestelmät ja tietojenkäsittelytieteestä myös tiedettiin enemmän. Lisäksi tutkimuksen mukaan tummaihoiset olivat muita vastanneita luottavaisempia alan työnäkymiin. [Seymour et al., 2004]

Tarkoittavatko tutkimustulokset sitä, että maassa, jossa opiskelijat pääsevät hyödyntämään tietokoneita enemmän, tietokoneisiin liittyviin aloihin suhtautuminen on erilaisempaa? Jos sama tutkimus toistettaisiin vaikkapa Suomessa, olisiko kiinnostus alaa kohtaan laimeampaa ja samalla alan tulevaisuuteen suhtauduttaisiin skeptisemmin? Totta on, että uudet ja harvojen saatavilla olevat asiat kiinnostavat yleisesti ihmisiä enemmän ja osittain vääristyneetkin käsitykset saattavat hallita yleistä mielipidettä. Uutta kohtaan ollaan kuitenkin usein myös tavallista epäileväisempiä. Saisimmeko kenties samoja tuloksia kuin Afrikassa, jos tutkisimme Suomessa opiskelijoiden suhtautumista käytettävyyteen liittyviin opiskelusuuntauksiin? Nimittäin samalla tapaa erilaiset käytettävyyteen liittyvät periaatteet ja opetussuunnaukset eivät ole vaikuttaneet pitkään Suomessa, monista muista maista puhumattakaan.

2.2. Sukupuolierot IT-alalla

IT-alaa, kuten monia muita teknisiä ja matemaattisia aloja, on pidetty yleisesti miehisinä aloina. Miehet ovat enemmistönä näillä aloilla, mutta paikoittain osat ovat myös vaihtuneet. Työympäristön erilaiset valtasuhteet voivat kuitenkin vaikuttaa työpaikan sisäiseen kommunikointiin ja sitä kautta myös työtehtäviin ja -ympäristöön. Onko sukupuolella siis vaikutusta työelämässä? Ovatko naiset miehiä tyytymättömämpiä työelämän laatuun ja mitkä työ- ja organisaatio-tekijät voivat vaikuttaa tähän?

Yhdysvalloissa vuonna 2003 tehty tutkimus, joka kohdistui 1278 samassa organisaatiossa työskentelevää henkilöön, osoitti IT-työntekijöiden yleensä olevan tyytyväisempiä työhönsä ja pitävät työmäärän räsitystä kohtuullisempina kuin muilla aloilla työskentelevät. Tutkimuksen mukaan sukupuolella ei ollut vaikutusta työelämän laatuun. Palautteen saaminen ja työntekijän autonomia yhdistettiin yleisesti tyytyväisyyteen ja työn räsittävyys puolestaan työmäärään, riippumatta sukupuolesta ja työn tyypistä. Toisaalta naisten

työtyytyväisyyteen vaikutti työmäärä ja tehtävien merkittävyys puolestaan vaikutti työn raskauteen, erityisesti niiden naisten keskuudessa, jotka työskentelevät IT-alalla. IT-alalla olevien naisten työn raskauteen ei taas vaikuttanut työn autonomisuus, joka puolestaan oli vaikuttavana tekijänä muilla aloilla työskentelevien naisten keskuudessa. Ainoastaan miehillä tehtävän merkittävyys ja vaadittavien taitojen monipuolisuus IT-alalla vaikutti työtyytyväisyyteen. [Carayon et al., 2003]

Yleisesti tutkimuksen mukaan IT-alalla työskentelevät ovat tyytyväisiä ja työmäärä on oikeanlainen, mutta tutkimus oli tehty yhdessä organisaatiossa, ei tätä tulosta voida siksi yleistää muihin alalla toimiviin yrityksiin. Miehet eivät ole tutkimuksen mukaan sen tyytyväisempiä työelämäänsä kuin naisetkaan, mutta tekijät, jotka vaikuttavat tähän, vaihtelevat sukupuolesta riippuen. Naiset pitävät vaativia ja aikaa vieviä työtehtäviä rasittavina tekijöinä, kun taas miesten kohdalla samat tekijät vaikuttivat työtyytyväisyyteen positiivisesti.

Sytä näihin suhtautumisten eroihin voi olla monia. Huomionarvoista on kuitenkin se, että naiset ja miehet suhtautuvat työtehtäviin vaikuttaviin tekijöihin eri tavoin ja sitä kautta kokevat työtehtävät erilaisina. Voitaisiinkin tarkastella sitä, millaisissa työasemissa naiset ovat miehiin nähden ja tätä kautta myös sitä, miten työntekijät tukevat toisiaan. Onko yrityksen sisäisessä kommunikaatiossa eroja miesten ja naisten keskuudessa? Kokevatko miehet saavansa enemmän tukea vaativissa työtehtävissä, vai onko taustalla jotain muita tekijöitä? Jälleen kerran tutkimuksen tekemistä laajemmalla otannalla tulisi myös miettiä.

3. Käytettävyys ammattina

Käytettävyyteen liittyvään opiskeluun ja ammatteihin voidaan hyvinkin katsoa vaikuttavan kaikki edelliset tekijät samalla tapaa kuin yleisesti IT-alalla. Tämä siksi, että käytettävyys alkaa saada yhä parempaa jalan sijaa mitä erilaisimmissa ohjelmistoprojekteissa. Suurimpia ongelmia puhuttaessa käytettävyyden ammateista on kuitenkin yleisesti hyväksytyjen täsmällisten määrittelyjen puuttuminen siitä, mitä käytettävyyden ammattilainen tarkalleen tekee, mikä on hänen todellinen asemansa organisaatiossa ja mitä säännöksiä hänen tulisi työssään noudattaa.

Käytettävyyden alan hajanaisuus ja työtehtävien vaihtelevat menetöt ja käytännöt, kuten myös niiden sulautuminen osaksi aivan toisentyypisiä työtehtäviä ja lähtökohtia, sekoittavat tilannetta entisestään. Tavoitteenani on kuitenkin yrittää määrittää jollakin tapaa työntekijöiden käytännön kautta, minkälainen tämä kokonaiskuva mahdollisesti on. Käytettävyyden alan

ammattien tutkimusta on tehty enemmän juuri käytettyjen metodien tarkastelun näkökulmasta.

3.1. Määrittelyn ongelmia

Käytettävyys voidaan määritellä usealla tavalla, painottamalla erilaisia näkökulmia käytettävyyteen ja tutkimalla varsinaiseen työhön ja tutkimukseen liitettyjä metodeja. ISO 13407-standardin [ISO 13470:1999] mukaan käytettävyys mittaa, miten hyvin määrätyt käyttäjät voivat käyttää tuotetta tietyssä käyttötilanteessa saavuttaakseen määritetyt tavoitteet tuloksellisesti, tehokkaasti ja tyytyväisinä. Standardin mukaan käyttäjäkeskeisessä suunnittelussa luodaan vuorovaikutteisia järjestelmiä, ottamalla samanaikaisesti huomioon useita aloja. Se huomioi inhimilliset tekijät ja ergonomian, joka parantaa tehokkuutta, työolosuhteita ja ehkäisee haitallisten tekijöiden vaikutuksia. Tärkeää on siis tutkia tarkkaan käyttäjien taitoja, tarpeita ja rajoituksia. [ISO 13470:1999]

Käytettävyyden alan edustajia kuuluu useisiin yhdistyksiin, joita ovat esimerkiksi ACM (The Association for Computing Machinery), UPA (Usability Professionals' Association) ja IEEECS (The IEEE Computer Society). Myös nämä osaltaan valvovat käytettävyyden periaatteiden toteutumista käytännön tasolla. Jokainen käytettävyyden ammattilainen todellisuudessa kuitenkin valvoo itse sitä, miten käytettävyys kunkin henkilön työssä toteutuu. Omalla kohdallani käytettävyys on saanut merkityksen juuri tuon tavallisen käyttäjän huomioimisen näkökulmasta. Tavoitteenani on kehittää kriittistä ajattelua siten, että pystyn asettumaan jokaisen käyttäjän asemaan ilman, että suunnittelussa olettais in mitään todellisen tiedon sijasta.

Käytettävyyden ammattien työskentelytavat ovat yleistyneet eri tavoin eri puolilla maailmaa. Madsen [1999] on tutkinut eroja käytettävyyden huomioon ottamisessa yhdysvaltalaisissa ja skandinaavisissa yrityksissä. Yhdysvalloissa on ollut aiemmin enemmän käytettävyyslaboratorioita, mutta Skandinaviassa käyttäjät on otettu huomioon epämuodollisesti suunnitteluprosessissa. Tämä tarkoittaa myös, että muodollisia käytettävyysohjelmia ei ole laadittu aiemmin Skandinaviassa. Nyt perinteitä on kuitenkin sekoitettu enemmän keskenään ja tilanne samankaltaistuu laajemmassa skaalassa katsottuna. [Madsen, 1999] Työtehtävien muotoutumiseen maiden sisäisesti vaikuttaa siis se, mitkä käytettävyyden metodit ovat yleistyneet ja missä aikataulussa. Kansainvälistyminen vaikuttaa myös käytettävyyden alan parissa ja tulevaisuudessa se tulee muokkaamaan työtehtäviä yhä enemmän.

Verrattaessa kolmea amerikkalaista yritystä (IBM, AIR – the American Institutes for Research ja Microsoft) ja kolmea tanskalaista yritystä (Danfoss,

B&O – Bang & Olufsen, KMD - Kommunedata) voidaan havaita eroja maiden välillä [Madsen, 1999]. Tanskalaisissa yrityksissä käytettävyyssryhmät perustettiin amerikkalaisia myöhemmin, 1980-luvun sijasta 1990-luvulla. Lisäksi käytettävyyssryhmissä työskentelee tanskalaisissa yrityksissä vähemmän ihmisiä – alle kymmenen henkilöä verrattuna yli sataan henkilöön. Vertailussa mukana olleet tanskalaiset yritykset olivat kuitenkin kokonaishenkilömäärältään pienempiä kuin amerikkalaiset yritykset, joten se voi selittää osaltaan ryhmien kokoa. Tämä vertailu on kuitenkin vain yksi esimerkki havaituista eroista maiden välillä, joten tulevaisuudessa tulisikin tutkia tarkemmin kaikkien alan yritysten käytettävyyden henkilökunnan kokoa suhteessa yritysten kokoon.

3.2. Käytettävyyden alan työtehtävien määrittelyä

Tuoreinta ja lähinnä suomalaista näkökulmaa on vuonna 2003 toteutettu tutkimus, joka käsittelee käytettävyyden alan ammatteja Ruotsissa. Tutkimuksen tarkoituksena oli selvittää, missä ja miten käytettävyyden alan työntekijät työskentelevät. Käytettävyyden alan työntekijäksi kyseisessä tutkimuksessa laskettiin henkilöt, jotka ajattelivat olevansa vastuussa käytettävyydestä organisaatiossa tai olivat mukana sen tyyppisessä projektissa työnimikkeestä riippumatta. [Gulliksen et al., 2004]

Tutkimus määrittelee tarkemmin työntekijöiden taustoja ja kokemuksia, sekä työllistymisen, organisaation, tuotteiden ja järjestelmien tyyppiä. Lisäksi tutkimus käsittelee myös projekteissa käytettyjä ohjelmistotuotannon tyyppisiä ja käytettävyyden sisältämiä olennaisia menestymisen tekijöitä. Tarkastelussa on mukana niin metodeja ja tekniikoita, kuin myös käytettävyyden vaikutuksia organisaatioon ja projekteihin. Tutkimustulokset viittaavat siihen, että organisaation ja projektin johdon tuki ovat olennaisia käytettävyyden alan työntekijälle. [Gulliksen et al., 2004] Organisaation tuella ja työntekijöiden välisellä kommunikoinnilla helpotetaan käytettävyyden työntekijään kohdistuvia paineita.

Kyseinen tutkimus toteutettiin internetlomakkeella, joka oli Nita:n (Nationellt IT-användarcentrum - National IT User Centre) sivustolla. Kysely oli avoin kaikille kiinnostuneille ja heitä myös pyydettiin lähettämään tietoa lomakkeesta muille kiinnostuneille. Vastaajia oli kaiken kaikkiaan 194. Täyttämispyyntöjä lähetettiin myös erilaisille sähköpostilistoille, jotta tutkijat olisivat saaneet mahdollisimman laajan tutkimusmateriaalin. Vastaajilta kysyttiin muun muassa taustoista ja kokemuksesta ja siitä, kuinka paljon he käyttävät aikaa käytettävyyteen liittyvään työhön. Lisäksi heiltä kysyttiin työllistymisen, organisaation, tuotettujen tuotteiden ja järjestelmien tyyppiä ja

sitä, kuinka käytettävyytyö on rahoitettu. Vastaajista 59 % oli miehiä, 115 miestä ja 79 naista, ja keski-ikäsi saatiin 38,9 vuotta. Vastaajat olivat työskennelleet keskimäärin 15,1 vuotta yhteensä ja 6,71 vuotta käytettävyyden parissa. Nykyisessä työssään he olivat olleet keskimäärin 5,26 vuotta. Suurimmalla osalla oli tutkinto tietojenkäsittelytieteestä tai tekniikan alalla verrattuna suunnittelu- ja ei-tekniikan alan yliopistotutkintoon. [Gulliksen et al., 2004]

Vastanneilta kysyttiin myös, missä ja miten he olivat kouluttautuneet käytettävyyden puolelle. Puolet vastaajista oli opiskellut käytettävyyttä opintokokonaisuutena tai käynyt yksittäisillä kursseilla osana päätutkintoaan. Itseoppineita oli 30 % vastaajista ja 20 % vastanneista oli oppinut käytettävyyttä työhön liittyvässä koulutuksessa. Vastaajista 26 % käytti työaikansa kokonaan tai melkein kokonaan käytettävyytyöhön, 19 % käytti enemmän kuin puolet. Loput 55 % vastaajista käytti vähemmän kuin puolet työajastaan käytettävyyteen tai työskenteli harvoin käytettävyyden parissa. [Gulliksen et al., 2004] Huomattavaa on, että alle puolet vastaajista oli opiskellut käytettävyyttä enemmän kuin yksittäisiä kursseja ja yli puolet käytti suurimman osan työajastaan muuhun kuin käytettävyyteen, vaikka tutkimuksen jakelu suunnattiin juuri alan ammattilaisille. Ongelmana on usein se, että IT-alalla yleisesti järjestäytyminen on pientä verrattuna muihin aloihin ja siksi tutkimuksiin on vaikeaa saada mukaan niitä henkilöitä, jotka eivät ole järjestäytymisessä aktiivisesti mukana.

Suurin osa tutkimuksen vastaajien työnimikkeistä sisälsi käsitteitä HCI-alalta (Human Computer Interaction), kuten käytettävyys, käyttöliittymä, sekä vuorovaikutus liitetynä koulutustaustaan, erikoistumisen tasoon ja työtehtäviin viittaaviin sanoihin. Vastaajista 18 % oli itsenäisiä työntekijöitä sekä palkattuja asiantuntijoita organisaatiossa, 73 % työskenteli yrityksen tai muun tyyppisen organisaation alaisena ja 9 % oli yliopistossa työskenteleviä. [Gulliksen et al., 2004] Käytettävyyden alan ammattilaiset toimivat siis useimmiten tietyn yksikön tarpeiden ja toiveiden alaisena.

Tutkimukseen osallistuneiden toimijoiden alat vaihtelivat suuresti, ja työtä tehtiin useimmiten projekteissa. Rahoitustyypeistä yleisimmät olivat erillinen laskuttaminen tai vuosittaisessa budjetissa huomioiminen, mutta vastaajat pystyivät valitsemaan myös useamman kuin yhden vaihtoehdon, joten erilaiset rahoituskombinaatiot ovat mahdollisia. Puolet vastaajista oli tyytyväisiä käytettyyn ohjelmistokehitysmalliin ja näkivät, että käytetystä mallista oli etua työssä. [Gulliksen et al., 2004]

Käytettävyyden alan työntekijät huomioivat usein työssään useampia tekijöitä samanaikaisesti ja varsinaisen työn helpottamiseksi usein tarvitaankin täydentäviä käytettävyyden huomioivia työtapoja. Käytetyt käytettävyyss-

metodit olivat muun muassa yleiset standardit, HCI-metodit (käyttäjätutkimukset, käytettävyydestit, kenttätestit) ja erilaiset heuristiikat. Käytettyjä metodeja ja tekniikoita oli monia, joista suosituimpia olivat ääneenajattelu, prototyypit, haastattelut, kenttätutkimukset ja skenaariot. [Gulliksen et al., 2004] Kyseisiä metodeja suositaan myös opetuksessa Tampereen yliopistossa.

Tutkimukseen vastanneista 62 prosenttia oli sitä mieltä, ettei käyttäjiä huomioida tarpeeksi ohjelmistokehitysprosessissa. Vastaajat sanoivat huolehtivansa käytettävyydestä yleisellä sekä tuote- ja käyttöliittymätasolla ja lisäksi kertoivat, että vastuu vaihtelee projektista riippuen. Vastaajien mielestä yksi tärkeimpiä tekijöitä onnistuneessa työssä on se, että käytettävyyden pitää olla alusta lähtien mukana projektissa ja lisäksi johdon, käyttäjien ja ohjelmistotuottajien tuki koettiin tärkeäksi. [Gulliksen et al., 2004] Valitettavaa on, että yli puolet vastanneista kokee, että käyttäjiä huomioidaan liian vähän ohjelmistokehitysprosessissa. Tämä vaikeuttaa käytettävyytyötä ja vaikuttaa myös työn lopputulokseen.

Tutkimukseen vastanneista itseoppineiden määrä on huomattavan suuri ja tutkinnon suorittaneita tulisi olla enemmän. [Gulliksen et al., 2004] Tämä kohdistaa paineita siis myös koulutuksen kehittämiseen. Käytettävyys ei saisi myöskään olla pelkästään pitempään alalla olleiden työtehtävänä tai ainoastaan ei-käytettävyysalan työtehtävien lisänä [Gulliksen et al., 2004]. Vastoin kyseisten tutkijoiden odotuksia, resurssit eivät olleet käytettävyyden huomioon ottamisen tiellä työssä, vaikka näin voisi ensin nopeasti kuvitella. Kaikkien projektin sidosryhmien tulisikin olla nykyistä paremmin tietoisia käytettävyyden eduista. [Gulliksen et al., 2004] Käytettävyyden alan kasvusta ja kehittymisestä huolimatta alan ammattilaiset painivat siis edelleen samojen ongelmien parissa, joita on ollut olemassa jo alan alkuvaiheista lähtien. Yksittäisen työntekijän kannalta tämä on harmillista, koska tällöin alan työtehtävien ja yleisten säännösten kehittyminen on hitaampaa.

Toinen esimerkkitutkimus selvittää käyttäjälähtöisen suunnittelun metodien käyttöä ammateissa koko teollisuudessa. Tutkimukseen osallistuneet 103 henkilöä olivat CHI'2000:n osallistujia ja UPA:n jäseniä, eli käyttäjälähtöisen suunnittelun harjoittajia, jotka olivat työskennelleet UCD-alalla (User Centered Design) vähintään kolme vuotta ja pitivät sitä päätyönään. Vastaajista 60 % työskenteli Yhdysvalloissa ja loput pääosin Euroopassa. Suurimmalla osalla vastaajista oli ammatillinen tutkinto ja lisäksi heillä oli siis usean vuoden kokemus käyttäjälähtöisestä suunnittelusta. Ammatillisen tietämyksen lähteinä olivat useimmiten kirjat ja lehdet, konferenssit, työpajat ja kollegat. [Vredenburg et al., 2002]

Tutkimuksessa asetettuja kysymyksiä olivat, mitä UCD-metodeja käytetään useimmin, mitkä ovat kunkin käytetyn metodin edut ja haitat harjoittajien mielestä ja mitä vaikutuksia UCD:lla on organisaatioon. UCD:lla tarkoitetaan tässä tutkimuksessa käyttäjien osallistumista ja heidän ymmärtämistä ja tehtävävaatimuksia, iteratiivista suunnittelua ja arviointia, sekä monipuolista lähestymistapaa. Kysely sisälsi muun muassa useita kysymyksiä UCD-metodien vaikutuksesta työntekoon ja vastaajia pyydettiin myös määrittämään käytettyjä metodeja ja niiden tehokkuutta. [Vredenburg et al., 2002]

Vastaajista melkein 63 % oli ollut viimeisen 12 kuukauden aikana mukana projektissa, joka käytti UCD:a ja käsitteli Internetiä tai intranetiä. Projektin yleisin koko oli 10 henkilöä ja 65 % tapauksista UCD:sta vastuussa olevia henkilöitä oli yhdestä kahteen henkilöä. Yleisimmin budjetista käytettiin 10 prosenttia UCD-työhön. Suurin osa vastaajista myös kertoi käytetyillä metodeilla olleen huomattavaa merkitystä tuotekehityksessä. Tämä havainto tehtiin ehkäpä myös siksi, että 41 % vastanneista toimi organisaatioissa omana yksikkönä. [Vredenburg et al., 2002]

Tutkimuksen mukaan käytetyimpiä metodeita olivat iteratiivinen suunnittelu, käytettävyyсарviointi, tehtäväanalyysi, epävirallinen asiantuntija-arviointi ja kenttätutkimukset. Tutkittaessa kunkin metodin hyötyjä ja haittoja, oli kuitenkin vaikeaa löytää yksittäisiä hyviä metodeja, sillä joku ominaisuus on toisessa metodissa hyöty ja toisessa taas haitta. Yleisimmin käytetyt metodit ja tekniikat olivat siis nopeita, informaaleja ja halpoja verrattuna muihin tekniikoihin. Organisaatioissa on lisäksi tutkimuksen mukaan hyvin vähän keinoja tutkia puolueettomasti UCD:n tehokkuutta. Tehokkaimmaksi havaittiin moninäkökulmainen lähestymistapa ja osittain tämän vuoksi myös työntekijöiden määrä vaikuttaa onnistumiseen. [Vredenburg et al., 2002]

Käytettävyyden työ yritetään saada usein siis hyvin edulliseksi, osasyynä voi olla juuri se, ettei sen etuja pystytä tarkasti määrittelemään. Myös työskentelytapojen arviointi toisiinsa nähden on vaikeaa, sillä mielipiteet vaihtelevat henkilöistä riippuen, kun tapojen tarkka teho ei ole mitattavissa. Toimenkuvista voidaan havaita myös se, että työntekijät pyrkivät tehokkuuteen usein puutteellisilla resursseilla, mutta ovat kehittäneet työtänsä siten, että se on jossain määrin mahdollista.

3.3. Käytettävyyden alan työnimikkeiden määrittelyä

Bernin ja Davisin [2002] tekemä tutkimus tukee myös havaintoa käytettävyyden alan työskentelijöiden koulutustaustan moninaisuudesta. Käytettävyyden pariin hakeutuvien työntekijöiden koulutus pohjaksi on olemassa useita eri koulutusaloja, joissa käytettävyyttä opetetaan aina HCI:stä

kognitiiviseen psykologiaan. Tämä kuitenkin heijastaa osittain sitä tosiasiaa, ettei ole olemassa yhtä oikeaa tapaa tehdä työtä käytettävyyden parissa. Useat työntekijät myös käyttävät käytettävyydmetodeja ilman, että tietävät niiden olevan todella käytettävyyden alan työtehtäviä. [Berni and Davis, 2002]

Bernin ja Davisin [2002] mukaan käytettävyyden alalla voidaan tunnistaa useita eri työnimikkeitä. *Käytettävyydestaajan* (Usability Tester) tehtävänä on käytettävyydestien suunnittelu ja toteutus sekä konsultointi. Hän toimii osana ulkopuolista yksikköä, joka palkataan tekemään käytettävyydestaukset. Tehtäviin kuuluu testauslaboratorion pystyttäminen, testausmateriaalin luominen, mukaan lukien ensimmäiset prototyypit ja dokumentaatio, testien käsikirjoitusten, kyselylomakkeiden ja haastattelujen tekeminen, osallistujien hankkiminen, testien läpivienti, tiedon jakaminen osallistujille ja johtopäätösten tekeminen kerätystä informaatiosta. Vaikeinta näissä työtehtävissä ovat juuri ne vaiheet, jossa osallistuja on mukana eli osallistujien seuraaminen testeissä ja heidän haastattelemineen jälkeenpäin.

Käyttöliittymäsuunnittelija (User Interface Designer) on osa tuotannon suunnittelutiimiä ja käyttää käytettävyydperiaatteita ja standardeja apuna työssään. Työtehtäviin kuuluu muun muassa käyttäjäprofiilien, tehtäväanalyysin, käytettävyydestausten tekeminen ja käyttöliittymän kielen ja tekstien suunnittelu [Molloy, 2000]. Tarkoituksena on siis kartoittaa käyttäjän tarpeet jo alussa. *Tekninen kommunikoija* (Technical Communicator) puolestaan toimii useissa työtehtävissä, joissa käyttäjälähtöinen suunnittelu otetaan huomioon jollakin tapaa. Hän ei välttämättä ole käytettävyydenalan varsinainen ammattilainen, vaan hän huomioi samat periaatteet oman alansa kautta. [Berni and Davis, 2002]

Käytettävyyden alan työntekijät voidaan jakaa lisäksi kahteen ryhmään koulutukseltaan ja työkokemukseltaan. Ensimmäinen ryhmä käsittää ne henkilöt, jotka ovat saaneet tietyn käytettävyydskoulutuksen ja joilla on paljon työkokemusta ja toisena ne, jotka ovat saaneet vähän käytettävyydskoulutusta ja jotka työskentelevät yrityksissä tutustuen työssään käytettävyyteen. Syynä tähän on usein myös se, ettei käytettävyyden periaatteita ole hyväksytty vielä kansainvälisesti. Tämä seikka vaikuttaa tietenkin myös koulutusohjelmien muotoutumiseen. Käytettävyyden ammattilaiseksi tullaan kouluttautumalla joko suoraan alalle tai sitten vaikkapa tietojenkäsittelytieteen kautta erikoistumalla käytettävyyteen vasta yrityksen sisällä. Suurin osa käytettävyydstyöntekijöistä työskentelee ohjelmistotehtävissä ja merkittävä vähemmistö akateemisissa tehtävissä. [Berni and Davis, 2002]

Opiskeluvaiheessa kuin varmasti myös myöhemmin työelämässä käytettävyyden alan toimijat joutuvat siis miettimään omia valintojaan, jotka puoles-

taan määrittävät edelleen toimenkuvien muotoutumista. Jokainen alalle valmistuva joutuu myös perustelemaan työtekniikoiden valintoja. Tarvittavan osaa-mispohjan saavuttamiseksi alalle pyrkivillä on useita vaihtoehtoja kouluttautua ja sitä kautta myös työllistyä, eikä oman aktiivisuuden merkitystä voi väheksyä. Alan kasvaessa vaihtoehdot myös entisestään paranevat. Monipuolinen koulutus kuitenkin takaa parhaiten työelämään sijoittumisen ja koulutuksen kehittämisen puolestaan alan kehittämistä oikeaan suuntaan.

3.4. Lisää lukuja alan rakenteen hahmottamiseksi

Vuodelta 2000 oleva UPA:n jäsen- ja palkkatutkimus kertoo lisää hyvinkin tarkkoja lukuja käytettävyyden ammattilaisista. Kyselyyn vastanneista 573 henkilöstä 57 % oli naisia, miehiä 43 %, sekä 75 % vastanneista oli iältään 26-45 -vuotiaita. Vastanneista 42 % oli vähintään maisterintutkinto. Vastaajista 75 % oli Yhdysvaltojen ulkopuolelta ja tästä osasta 1 % oli suomalaisia. Puolet vastaajista raportoi heidän työnsä liittyvän kokonaan käytettävyyteen, HCI:iin, käyttäjävuorovaikutukseen ja human factors -puoleen. Vastaajista 33 % oli työskennellyt alalla kahdesta neljään vuotta. [UPA, 2001] Nämä luvut toistavat sitä huomiota, että alan koulutus ja työtehtävien painottuminen vaihtelee suuresti.

UPA:n tutkimuksen kaikista vastaajista 540 vastannutta ilmoitti palkkatietonsa. Kaikkien vastanneiden keskimääräinen vuosipalkka oli 74 838 \$ (luvussa ei mukana konsultteja) ja käytettävyystyöntekijöiden keskimääräinen vuosipalkka oli puolestaan 70 094 \$ (luvussa ei mukana konsultteja). Tutkittaessa sukupuolten välisiä eroja, oli miesten keskimääräinen vuosipalkka 80 052 \$, kun naisten vuosipalkka oli puolestaan 70 820 \$ (alin raportoitu vuosipalkka 16 800 \$ sekä ylin 250 000 \$). [UPA, 2001] Vastanneista 57 % oli tyytyväisiä palkkaan työhönsä nähden, joten korjattavaa tässä suhteessa olisi. Voidaankin kysyä, olivatko naiset tyytymättömämpiä palkkaukseensa kuin miehet.

3.5. Käytettävyys osana isompaa organisaatiota

Venturin ja Troostin [2004] tutkimus keskittyy tarkastelemaan, kuinka käyttäjäkeskeinen suunnittelu on integroituna teollisuuteen. Tutkimus oli suunnattu nimenomaan käytettävyyden parissa työskenteleville ja kysely jaettiin sähköpostin kautta uutisryhmille ja foorumeille, jotka liittyvät käytettävyyteen ja käyttäjäkeskeiseen suunnitteluun. Vastaajia oli niin isoista kuin pienistä yrityksistä aina erikoistuneisiin konsultteihin asti, yhteensä 83 henkilöä. Vastaajista 34 % oli human factor -asiantuntijoita ja 33 % käyttöliittymä suunnittelijoita ja vastaajien työkokemus vaihteli viiden ja kolmentoista vuoden välillä. Tutkimukseen osallistujia oli erityyppisiltä yritysaloilta - esimerkkinä telekommuni-

kaatio, koulutus, ja markkinointi - mutta suurin osa oli HCI- ja käytettävyydensuunnittelija, yhteensä 16 vastaajaa. Lopuille 67 vastaajalle käytettävyys oli vain yksi näkökulma työnteossa, kun taas käytettävyydentekijät pitivät sitä päätyönään. [Venturi and Troost, 2004] Tutkimuksen otanta on kuitenkin edellisiä tutkimuksia hajanaisempi ja siksi tulokset eivät kenties anna tarpeeksi täsmällistä tulosta.

Vastaajia pyydettiin myös ajattelemaan tutkimukseen vastattaessa jotain tiettyä yksittäistä UCD-projektia. Tutkimuksen mukaan käytettävyydentekijät ovat usein työllistyneinä isoissa yrityksissä, mutta heitä on vähän lukumäärällisesti – he ovat usein myös vähemmän kuin yksi prosentti yrityksen työntekijämäärästä. Työtehtävissään käytettävyydentekijät suosivat käyttäjähaastatteluita ja matalan ja korkean tason prototyyppijä useimmin. Jotta käytettävyyden työntekijöiden tilanne paranisi nykyisestä tuloksesta, täytyisi organisaatiossa parantaa johdon tukea, ottaa tarkastelun alle organisaatiossa oleva infrastruktuuri ja käytetty kommunikaatiotapa. [Venturi and Troost, 2004] Jälleen kerran voidaan havaita, että käytettävyyden ammattilaisten lukumäärä ja käytetyt metodit määrittyvät yrityksissä siten, että ne vievät vähiten aikaa ja rahaa.

Tutkimus antoi siis hyvin samansuuntaisia tuloksia edellisiin verrattuna. Käytettävyys pitäisi edelleen ottaa paremmin huomioon yrityksissä, sillä se koetaan usein liian monimutkaisena lähestymistapana, sekä aikaa kuluttavana ja kalliina. Tämä vaikuttaa puolestaan työtehtävien sisältöön. Käyttäjälähtöisen suunnittelun integraatiolla tarkoitetaan sitä, että jokaisessa ohjelmistokehityksen vaiheessa otetaan huomioon suunnittelumallin periaatteet. Tämä onnistuu kun kyseistä mallia noudattavat asiantuntijat ovat kokeneita ja heillä on tarvittavat tiedot ja taidot asiasta ja tätä kautta johtokin saadaan sitoutumaan paremmin suunnittelumalliin. [Venturi and Troost, 2004] Käytettävyyden työntekijöiden tulee myös miettiä sitä, miten he näyttäytyvät ulospäin yrityksen sisällä.

HCI-alan ulkopuolella toimivat vastaajat työskentelivät isoissa yli 1000 hengen yrityksissä, joissa käytettävyys oli otettu huomioon muutaman vuoden ajan. [Venturi and Troost, 2004] Tämä voi tarkoittaa myös sitä, että käytettävyys ei välttämättä ole täysin sulautunut vielä mukaan yritysten toimintaan. Käytettävyydentekijöiden määrä oli myös hyvin pieni, puolessa yrityksistä alle yksi prosentti työntekijöistä. Tämä tarkoittaa sitä, että useimmiten isot yritykset palkkaavat tutkimuksen mukaan 2-6 käytettävyydentekijää [Venturi and Troost, 2004].

UCD-suuntautunut työryhmä oli tutkimuksen mukaan organisoituna keskusosastoon tai erilliseksi tiimiksi tai vaihtoehtoisesti molempia, ja työryh-

mät rahoitettiin tutkimus- ja kehitysbudjetin kautta, erillisellä laskutuksella tai huomioiden vuosittaisbudjetissa. Tutkimuksen mukaan rahoitustapoja voi olla myös useita samaan aikaan. HCI-konsultit työskentelevät alle 50 henkilön yrityksissä ja sitä kautta myös heidän määrä on suurempi suhteessa työllistettyihin nähden. Suurin osa näistä konsulteista on myös organisoitu tiimeiksi, jotka on rahoitettu projektitasolla. [Venturi and Troost, 2004] Tehokkuuden vertailussa tulisi siis ottaa huomioon käytettyjen metodien lisäksi käytettävyyden ammattilaisten sijoittuminen koko yritykseen nähden, koska se vaikuttaa osaltaan onnistumiseen.

Vastaajilta kysyttiin myös, mitä metodeja he käyttävät missäkin ohjelmistokehityksen vaiheessa. Useimmin käytetty metodi oli käyttäjähaastattelut - vastaajista 80 prosenttia käytti sitä ainakin kerran. Korkean ja matalan tason prototyypit, asiantuntija- ja heuristinen arviointi, käytettävyytestaus ja todellisen käytön tarkkailu olivat toiseksi suosituimpia. UCD-metodit otettiin useimmiten käyttöön projektin vaatimus-, analysointi- ja suunnitteluvaiheessa. Johto yleensä huomioi käytettävyyden osana yritysstrategiaa ja teki aloitteita ylläpitääkseen ja parantaakseen käytettävyyden osuutta. Tutkimuksen mukaan johto ei tosin kuitenkaan tarjoa käytettävyysepäjämiä tai keinoja saavuttaa niitä. Pääsääntöisesti vastanneiden mielestä kommunikaatio käytettävyyden puolella pelaa yrityksen sisällä, mutta kyseinen kommunikointi ei kuitenkaan näy yrityksestä ulospäin. [Venturi and Troost, 2004]

Alan kokonaiskehityksen kannalta tässä suhteessa olisi siis parannettavaa. Käytettävyyden työntekijät joutuvat tekemään isoja päätöksiä toimenkuvan kehittämisestä ilman organisaation tukea. Käytettävyyttä voidaan lisäksi mainostaa ulospäin ottamalla se huomioon yhtenä markkinatekijänä, jolloin sen tulisi näkyä myös asiakkaalle yrityksestä ulospäin. Jos käytettävyyden työn teho voitaisiin mitata tarkemmin rahassa, saataisiin varmasti myös johto kiinnostumaan ja luomaan kannustavia toimintatapoja. Käytettävyyden periaatteiden markkinointi riippuu siis myös tämän näkökulman hyväksymisestä alan ammattilaisten parissa.

Venturin ja Troostin tutkimusasettelussa on kuitenkin useita ongelmia. Koska vastaajia pyydettiin ajattelemaan yhtä yksittäistä projektia, voi valittu projekti perustua täysin vastaajan kuvitelmiin tai vastaavasti olla todellista ideaalimpi. Tutkimuksessa ei myöskään vaadittu antamaan omia henkilötietoja siinä pelossa, että vastaaminen voisi vaikuttaa työntekijän asemaan yrityksessä. Nämä tekijät vaikuttavat myös tulosten todenmukaisuuteen.

4. Paikallinen näkökulma käytettävyyteen

Ulkomailla on siis suoritettu hyvinkin tarkkaa toimenkuvatutkimusta käytettävyyden saralla. Suomessa tilanne on kuitenkin toinen, mikä voi osaltaan johtua siitä, ettei ala ole vaikuttanut ajallisesti kauan, vaikka olemme IT-puolella yksi edistyneimmistä maista. Alan toimenkuvatutkimus Suomessa on ainakin toistaiseksi hyvin karkeajakoista. Muilla aloilla toimenkuvatutkimusta on tehty paljon enemmän ja niistä tulisikin ottaa oppia kehitettäessä käytettävyyden alan toimenkuvatutkimusta.

4.1. Alan toimenkuvatutkimus Suomessa

Suomalaisena esimerkkinä on Haapakorven [2000] toimenkuvatutkimus IT-alalla. Tämä tutkimus tehtiin strukturoidulla puhelinhaastatteluilla ja osallistujat valittiin satunnaisotannalla vuosina 1984, 1989 ja 1994 Helsingin kauppakorkeakoulusta ja Helsingin yliopistosta, yhteiskunta-, kauppa-, luonnon- ja maatalous-metsätieteistä valmistuneista. Haastatteluja tehtiin yhteensä noin sata.

Tutkimuksessa havaittiin, että vastaajien oli vaikea kuvailla työtään, joka jokapäiväisenä on heille itsestään selvää. Työn vaativuutta on myös vaikea arvioida objektiivisesti ja siksi vastaajia pyydettiin kuvailemaan juuri kyseistä haastattelun aikaista työpäivää ja heille esitettiin paljon jatkokysymyksiä. Vastaajilta kysyttiin myös kuvausta epätavallisesta ja ongelmallisesta työpäivästä. Tutkimuksessa haastateltiin työntekijöiden lisäksi myös heidän esimiehiään. Esimiesten on kuitenkin vaikea kuvailla toimenkuvia ilman, että siihen liittyisi työntekijän omia henkilökohtaisia ominaisuuksia. Esimiesten ja haastateltujen kertomukset kuitenkin erosivat harvoin olennaisesti toisistaan, mutta työtehtäviä, työorganisaation rakennetta ja toimintaa painotettiin eri tavoin. Nämä painotukset johtuivat juuri erilaista asemista yrityksen sisällä. [Haapakorpi, 2000] Myös käytettävyyden alalla tulisi suosia työntekijöiden haastattelun lisäksi esimiesnäkökulmaa, sillä suurin osa työntekijöistä toimii vahvasti esimiesten alaisena.

Tutkimuksen tuloksia atk-alan asiantuntijoista tarkasteltaessa täytyy muistaa, että vastaajien lukumäärä oli hyvin pieni. Haastateltujen työn kohde oli tietokonepohjainen standardoitu tietojenkäsittely [Haapakorpi, 2000]. Asiantuntijoiden ammattiryhmän tutkimus koostui kuudesta toimenkuvahaastattelusta ja kahdesta esimieshaastattelusta, ja vastanneiden koulutustausta oli kirjava. Yksi kuudesta henkilöstä oli opiskellut pääaineena tietojenkäsittelyoppia ja yksi sivuaineena tietojenkäsittelyoppia. Muut haastatellut olivat käyneet tietokonekursseja ja työn tekemisen kautta opetelleet, eivätkä he olleet

varsinaisessa esimiesasemassa. Tutkimuksen mukaan asiantuntijuus muotoutuu atk-tiedon ja soveltamisalueen mukaisesti. [Haapakorpi, 2000]

Haapakorven mukaan ryhmä jakaantuu kolmeen osaan erikoistumisen ja toimenkuvan mukaan. *Konsulttien* tehtävänä on asiakkaiden konsultointi, koulutus, suunnittelu ja ylläpito. He ovat erikoistuneet tietyn alan sovelluksiin, ja he ovat toimineet sovelluksiin liittyvällä alalla aiemmin. Heidän täytyy siis tietää sekä tietokoneista että asiakkaan liiketoiminnasta. Tarkempia työtehtäviä ovat suunnittelu, ohjelmointi, testaus, kouluttaminen, tuki ja ylläpito. Työtehtävät olivat myös painottuneet hieman eri tavoin vastaajasta riippuen ja työtehtäviin vaikuttavat lisäksi muutokset asiakkaiden toimintatavoissa. [Haapakorpi, 2000]

Toisena ryhmänä olevien *atk-suunnittelijoiden* tehtävänä on puolestaan atk-järjestelmien ja ohjelmien toiminnasta vastaaminen organisaation sisällä. Tarkempia työtehtäviä ovat käyttöönoton suunnittelu, ylläpito ja kehitys. Lisäksi he toimivat asiantuntijoina ulkoisia palveluita ja tuotteita ostettaessa. Näillä henkilöillä koulutus ja työkokemus ovat tärkeämmässä asemassa kuin muilla ryhmillä. He toimivat myös yhteistyössä muiden työntekijöiden kanssa projektien suunnittelussa, mutta heidän työnsä ei kuitenkaan ollut kovin laaja-alaista työtehtäviltään ja vastuualueiltaan. Työtehtäviin voi vaikuttaa kuitenkin myös mahdollinen ulkoistaminen. Kolmannen ryhmän eli *atk-alan suunnittelija-kouluttajien* tehtävänä on atk-suunnittelu, tukihenkilötoiminta, koulutus ja sen suunnittelu. Heidän työkentänsä on puolestaan hyvin laaja ja myös heillä työtehtäviin voi vaikuttaa mahdollinen ulkoistaminen. [Haapakorpi, 2000]

Tutkittaessa Suomessa käytettävyyden alan toimenkuvia tulisi siis huomioida edelliseen tapaan samoja tekijöitä. Lisäksi, tulisi miettiä, mitä muita työtehtäviä käytettävyyden lisäksi heidän toimenkuvaansa kuuluu ja miten nämä tehtävät painottuvat ajallisesti. Myös yrityksen sisäinen kommunikaatio ja yhteistyö vaikuttavat työtehtäviin ja ne tulisikin ottaa huomioon. Alan kansainvälistymis- ja ulkoistamispaaineet vaikuttavat yhä enemmän työtehtäviin ja työkentän laajenemiseen. Ja nämä tekijät ovat vain pieni osa monien muiden joukosta.

4.2. Käytettävyyden ammattilaisia Tampereen yliopistosta

Tampereen yliopistossa ylemmän korkeakoulututkinnon suoritti 1049 opiskelijaa vuonna 2003. [TAY, 2004] Samana vuonna ura- ja rekrytointipalveluiden teettämässä selvityksessä kartoitettiin 743 vastanneen kesken heidän työllistymistään valmistumisen jälkeen. Työtilanne vuoden kuluttua valmistumisesta oli luonnontieteellisellä koulutusosalalla yleisesti ottaen hyvä. Tarkastelen tässä

juuri luonnontieteelliseltä koulutusalaalta valmistuneita, koska ne sisältävät myös tietojenkäsittelytieteiden edustajat.

Vastanneista 58 % oli vakituisesa työssä ja 33 % määräaikaisessa työsuhhteessa, 6 % puolestaan työttömänä työnhakijana. Suurin osa oli ollut työssään melkein koko vuoden: 85 % työssä olevista oli ollut työssä yli yhdeksän kuukautta vuosi valmistumisen jälkeen ja 11 % puolestaan yhdestä kuuteen kuukautta. Voidaan sanoa, että he ovat työllistyneet suurelta osin myös koulutustaan vastaavaan työhön, sillä yli yhdeksän kuukautta työssä olleita, sekä myös koulutusta vastaavassa työtehtävässä olevia oli yhteensä 72 % vastanneista. [TAY, 2004]

Hieman yli puolessa tapauksista (59 %) työnantajana oli yritys tai järjestö ja 39 % vastanneista toimi julkisen organisaation alaisena. Kyselyn mukaan nykyinen työtehtävä vastaa tutkinnon vaatimustasoa kokonaan 46 % vastanneista ja 43 % osittain. [TAY, 2004] Kyseinen luku tietenkin voisi olla korkeampi. Tutkimustehtävissä vastaajista on kolmasosa (34 %), opetus- ja koulutustehtävissä (23 %), hallinto-, suunnittelu- tai kehitystehtävissä (20 %) viidennes ja johto- ja esimiestehtävissä 11 % vastanneista. [TAY, 2004] Nämä luvut eivät kuitenkaan kerro tilannetta käytettävyyden alan näkökulmasta, vaan antavat enemmän näkökulmaa koko IT-alan tilanteeseen.

Tutkittaessa tarkemmin tietojenkäsittelytieteiden laitokselta valmistuneita, vastasi heidän tilanteensa koko tiedekunnan työllistymistilannetta. Vakituisesa työssä tietojenkäsittelyopista valmistuneista oli 17 henkilöä 23 vastanneesta ja vuorovaikutteisesta teknologiasta molemmat kaksi henkilöä kahdesta vastanneesta. Vuorovaikutteisen teknologian molemmat vastanneet olivat olleet koulutusta vastaavassa työssä yli yhdeksän kuukautta vuosi valmistumisensa jälkeen. Molempien työnantajana oli yksityinen yritys ja toinen heistä vastasi, että työ vastaa vaatimustasoltaan tutkintoa kokonaan ja toisen kohdalla työ vastasi osittain. Työn luonne oli toisella tutkimustehtävät ja toisella johto- ja esimiestehtävät. Ammattinimikkeikseen he raportoivat design engineer- ja section manager- nimikkeet.

Kysely ei kuitenkaan anna todellista tietoa, sillä otannan tulisi olla huomattavasti suurempi. Tutkittaessa vuorovaikutteisesta teknologiasta valmistuneita yhden ainoan vuoden osalta, on kuitenkin mahdotonta saada useampia vastaajia vähäisen valmistumislukumäärän vuoksi. Siksi vuorovaikutteisesta teknologiasta valmistuneiden työelämään sijoittumista tulisikin tutkia useamman vuoden ajalta. Toimenkuvatutkimusta tulisi myös tarkentaa edellä mainitun tavoin. Pelkkien nimikkeiden keräämisellä ei saada tietoa, sillä ne eivät kerro tarpeeksi työtehtävien rakenteesta. Tutkimus onkin syytä viedä vielä konkreettisemmalle tasolle.

5. Yhteenveto

Käytettävyys on käyttäjästä lähtöisin olevien tekijöiden huomioimista, joilla parannetaan tuotetta. Toimenkuvatutkimuksen tehtävänä on puolestaan tutkia toimenkuvaa kokonaisuutena huomioiden tiettyyn toimenkuvaan vaikuttavat tekijät. Käytettävyiden alan toimenkuviin voidaan väittää liittyvän tavallista useampia tekijöitä. Alan hajanaisuus ja toimenkuvien vakiintumattomuus vaikuttavat toimenkuvien muotoutumiseen, kuten myös alan toimenkuvatutkimukseen. Käytettävyiden alan toiminimikkeet ovat myös vakiintumattomia. Koulutuksen valinnalla ja omalla aktiivisuudella työllistymisen ja työtehtävien suhteen on vaikutusta työtehtävien sisältöön. Myös kulttuuritaustasta ja sukupuolesta riippuen työtehtävät koetaan eri tavoin.

Käytettävyiden alan toimenkuviin vaikuttaa siis niin alan sisäiset kuin ulkoiset tekijät ja yksittäisellä työntekijällä on suurin vastuu työssään onnistumisesta ja toimenkuvan kehittamisestä. Yksittäinen työntekijä yleensä vaikuttaa myös työssä käytettyihin metodeihin ja sitä kautta niiden yleistymiseen. Käytettävyiden ammattilainen toimii usein osana isompaa organisaatiota, joka vaikuttaa työntekoon oleellisesti. Organisaatiotuki ja kommunikointi ovat siis työn onnistumiseen suuresti vaikuttavia tekijöitä. Tulevaisuudessa alan kasvaminen ja kansainvälistyminen tulee muuttamaan toimenkuvia huomattavasti.

Nämä tekijät asettavat haasteita myös tulevaisuuden käytettävyiden alan toimenkuvatutkimuksen kehitykselle. Tämän kirjallisuuskartoituksen ongelmana onkin juuri nykyisen toimenkuvatutkimuksen puutteellisuus. Toimenkuvatutkimukset ulkomailla ovat paikallisia ja keskittyvät yksittäisiin kohteisiin ja Suomessa toimenkuvatutkimus on puolestaan hyvin karkeajakoista. Siksi tämä kartoitus haastaakin käytettävyiden alan toimenkuvatutkimusta kehittämään, jotta saisimme selville, mitä käytettävyystyö tarkoittaa Suomessa tänä päivänä on.

Viiteluettelo

[Berni and Davis, 2002] Kevin Berni and Marjorie T. Davis, *Careers in Technical Communication: Usability*. Formal Report for TCO 351 Reports and Proposals, Mercer University, 2002.

[Carayon et al., 2003] Pascale Carayon, Peter Hoonakker, Segolene Marchand and Jen Schwarz, *Job characteristics and quality of working life in the IT workforce: the role of gender*. In: *Proceedings of the 2003 SIGMIS Conference on Computer Personnel Research: Freedom in Philadelphia--Leveraging*

- Differences and Diversity in the IT Workforce*, Philadelphia, Pennsylvania, 2003. ACM Press, New York, NY, 58-63.
- [Chabrow, 2003] Eric Chabrow, Jobless recovery: IT isn't producing more jobs, but new data shows some categories fare worse than others. *InformationWeek*, (October 20, 2003).
- [Gardiner et al., 2004] Adrian Gardiner, Vladan Jovanovic and Han Reichgelt, Second thoughts about a career in IT? In: *Proceedings of the 5th Conference on Information Technology Education*, Salt Lake City, UT, USA, 2004. ACM Press, New York, NY, 194-201.
- [Gulliksen et al., 2004] Jan Gulliksen, Inger Boivie, Jenny Persson, Anders Hektor and Lena Herulf, Making a difference: a survey of the usability profession in Sweden. In: *Proceedings of the Third Nordic Conference on Human-Computer Interaction* Tampere, Finland, 2004. ACM Press, New York, NY, 207-215.
- [Haapakorpi, 2000] Arja Haapakorpi, *Nörtti, pomo ja yleismiesjantunen – akateemisten urat ja toimenkuvat*. Helsingin yliopiston Lahden tutkimus- ja koulutuskeskuksen raportteja ja selvityksiä, 33/2000.
- [ISO 13407:1999] ISO 13407:1999, Human-centered design processes for interactive systems.
- [Kaarst-Brown and Guzman, 2005] Michelle L. Kaarst-Brown, and Indira R. Guzman, Who is "the IT workforce"?: challenges facing policy makers, educators, management, and research. In: *Proceedings of the 2005 ACM SIGMIS CPR Conference on Computer Personnel Research* Atlanta, Georgia, USA, 2005. ACM Press, New York, NY, 1-8.
- [Madsen, 1999] Kim Halskov Madsen, The diversity of usability practices. *Commun. ACM* **42**, 5 (1999), 60-62.
- [Molloy, 2002] Karen Molloy, Getting involved in product usability as a technical communicator. *Usability Interface* **6**, 3 (Jan. 2000).
- [Pelkonen, 1999] Tommi Pelkonen, *Uusmediatoimialan toimenkuvat ja osaamistarpeet 1999*. ESR-julkaisut -sarja, 40/99, 1999.
- [Seymour et al., 2004] Lisa Seymour, Mike Hart, Peter Haralamous, Tasneem Natha, and Chia-Wen Weng - Inclination of scholars to major in information systems or computer science. In: *Proceedings of the 2004 Annual Research Conference of the South African institute of Computer Scientists and information Technologists on IT Research in Developing Countries* (Stellenbosch, Western Cape, South Africa, 2004. South African Institute for Computer Scientists and Information Technologists, 97-106.
- [TAY, 2004] Tampereen yliopistosta 2003 valmistuneiden työelämään sijoittuminen. Tampereen yliopiston ura- ja rekrytointipalvelut 2004.

- Saatavissa myös: <http://www.uta.fi/rekrytointi/sijoittuminen/sijoittumisseuranta03.pdf> [viitattu 05.07.2005].
- [UPA, 2001] Usability Professionals' Association, 2000 UPA Member Profile and Salary Survey, 2001. Also available: http://www.usabilityprofessionals.org/upa_publications/upa_voice/survey/2000_survey.html
- [Venturi and Troost, 2004] Giorgio Venturi and Jimmy Troost, Survey on the UCD integration in the industry. In: *Proceedings of the Third Nordic Conference on Human-Computer Interaction* Tampere, Finland, 2004. ACM Press, New York, NY, 449-452.
- [Vredenburg et al., 2002] Karel Vredenburg, Ji-Ye Mao, Paul W. Smith, and Tom Carey, A survey of user-centered design practice. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves* Minneapolis, Minnesota, USA, 2002. ACM Press, New York, NY, 471-478.

Huomioita matkapuhelinten käyttöliittymien suunnittelusta sokeille käyttäjille

Saija Hälikkä

Tiivistelmä.

Tässä tutkielmassa tarkastellaan uusien käyttöliittymien ja vuorovaikutus-tekniikoiden soveltuvuutta sokeiden käyttäjien matkapuhelimiin. Aihetta on tutkittu etenkin erilaisten puettavien ja kannettavien mobiililaitteiden parissa. Motiivina on kuitenkin kehittää vaihtoehtoisia käyttöliittymiä näkeville käyttäjille tilanteisiin, joissa katsominen on rajoittunut tai jopa estynyt kokonaan. Sokeiden käyttäjien kannalta tutkimus on painottunut lähinnä pöytätietokoneiseen tai muihin mobiililaitteisiin kuin matkapuhelimiin. Tutkielmassa käydään läpi sokeille käyttäjille sopivia vuorovaikutusmuotoja ja lopuksi ehdotetaan kaksi vaihtoehtoa sokeille käyttäjille mahdollisesta käyttöliittymästä matkapuhelimeen.

Avainsanat ja -sanonnat: Matkapuhelin, käyttöliittymät, sokeat käyttäjät, vuorovaikutustapa, käytettävyys

CR-luokat: H.5.2, H.1.2, I.3.6

1. Johdanto

Matkapuhelimet ovat yleistyneet viime vuosina Suomessa, jossa vuonna 2004 matkapuhelimia oli 91 prosentilla 15 - 79 -vuotiaista, eli noin 3.7 miljoonalla henkilöllä [Snellman, 2005]. Samaan aikaan kiinteiden puhelinverkkojen osuus on kääntynyt laskuun. Kiinteiden verkkojen puheluhinnat ovat nousseet noin 40 % 90-luvun puolivälistä vuoteen 2003 asti, kun taas matkapuhelujen hinnat ovat puolestaan laskeneet lähes 40 % [Tikkanen et al., 2004].

Matkapuhelinten kehittyessä yhä monipuolisemmaksi tulee niiden käytöstä myös haastavampaa varsinkin erityisryhmille. Enää matkapuhelimet eivät ole pelkkiä lankapuhelimia korvaavia puhelimia vaan yhä monimutkaisempia multimedia- ja viihdekeskuksia. Jotta saatavilla olevien laitteiden käyttö olisi mahdollista myös erityisryhmille, kuten näkörajoitteisille ja sokeille, tarvitaan käyttöliittymiä, jotka tukevat myös muunlaista kuin pelkkään näkökykyyn perustuvaa vuorovaikutustapaa.

Sokeille ja näkörajoitteisille käyttäjille on kehitetty jo kauan useita erilaisia käyttöliittymiä. Tutkimuksia aiheesta onkin ilmestynyt mm. pöytätietokoneiden käytöstä sokeiden käyttäjien keskuudessa [Dobrišek et al., 2003; Donker et al., 2002; Edwards et al., 1995; Pitt and Edwards, 1996]. Myös sokeille käyttäjille suunnatuista käyttöliittymistä muissa kannettavissa tai puettavissa

mobiililaitteissa kuin matkapuhelimissa on ilmestynyt tutkimuksia [Hub et al., 2004;]. Lisäksi erilaisia ns. katseen vapauttavia (eyes-free) vuorovaikutusmenetelmiä kannettaviin laitteisiin on kehitetty näkevien käyttäjien tarpeisiin, jolloin katsominen on vaikeaa tai jopa mahdotonta (mm. [Holland et al., 2002; Brewster et al., 2003; Silfverberg, 2003; Pirhonen et al., 2002]). Tämän tutkielman tarkoituksena on mm. esitellä erilaisia mahdollisia tekniikoita sokeille tarkoitetuissa matkapuhelinten käyttöliittymissä.

Matkapuhelinta on mahdollista käyttää useissa erilaisissa tilanteissa. Sen vuoksi uusiin vuorovaikutustekniikoihin on alettu kiinnittää huomiota myös näkevien käyttäjien kannalta. Ihanteena olisikin, että markkinoilla olevista matkapuhelimista ei tarvitsisi tehdä erillisiä erikoisvarusteltuja malleja erityisryhmille, vaan parantaa olemassa olevien matkapuhelinten käyttöliittymiä ja parantaa siten puhelinten saatavuutta erityisryhmille. Uudet vuorovaikutusmuodot voivat näin ollen hyödyttää myös valtaosaa muista käyttäjistä, joiden käyttötavat ovat yhä monipuolisempia ja vaativimpia. Erityisesti markkinoilla olevat älypuhelimet tarjoavat tilaisuuden parantaa käyttöliittymiä, sillä älypuhelimet vastaavat suorituskyvyltään ja ominaisuuksiltaan jo enemmän pientä tietokonetta kuin perinteistä lankapuhelinta.

2. Perinteiset käyttöliittymät matkapuhelimissa

Matkapuhelinten käyttöliittymässä useimmiten on näppäimiä, yleisesti lankapuhelimista tutut numerot sekä muita näppäimiä, näyttö, mikrofoni ja kuuloke. Nykyään matkapuhelimista pyritään tekemään yhä pienempiä fyysiseltä kooltaan. Puhelimet ovat kevyitä ja tehokkaita sekä niihin pyritään lisäämään puhelimen lisäksi muita ominaisuuksia, jotka osaltaan tuovat haasteita matkapuhelinten käyttöliittymäsuunnitteluun. Matkapuhelinten lähestyessä ominaisuuksiltaan kämmentietokoneita puhelinominaisuuksiin soveltuva käyttöliittymä ei riitä. Yksi ratkaisu on kasvattaa näytön kokoa, mutta samalla pyritäessä pienentämään puhelimen kokoa on pakko tehdä kompromisseja tai kehittää aivan uusia vuorovaikutustekniikoita. Toisaalta ominaisuuksien lisääntyessä myös tekniikka kehittyy, joka taas osaltaan luo uusia resursseja käyttöliittymille niin paremman laitteiston kuin uusien vuorovaikutustapojen myötä.

Ensimmäiset matkapuhelimet olivat langattomia puhelimia, mutta nykyään on vaikea löytää markkinoilta matkapuhelinta, jossa ei olisi myös monia muita ominaisuuksia puhelimen lisäksi. Puhelin laitteena on helposti saatavilla myös näkörajoitteisille ja sokeille, sillä kymmenen numeron sijainnin ulkoa opettelu ei liene mahdotonta. Lisäksi numeroiden sijainnit noudattavat useimmiten samaa standardia eri laitteissa. Mutta viimeistään tekstiviestien yleistyttyä on

matkapuhelimista alkanut tulla monimutkaisia hallita ilman näköä. Puhelinta on vaikea käyttää, sillä valikoiden koko on kasvanut, näppäimiä saattaa olla useampia eri tarkoituksiin ja samalla kuitenkin laitteen kokoa pienennetään.

Nykyisin matkapuhelimia voidaan käyttää moniin eri tarkoituksiin. Sillä voidaan soittaa puheluita, kirjoittaa ja vastaanottaa tekstiviestejä, käyttää sähköpostia ja muita internet-palveluita sekä lukuisia muita palveluita aina bussilippujen ostosta musiikin lataamiseen ja soittamiseen. Lisäksi kamerakäynykät ovat yleistyneet. Kaikki nämä toiminnallisuudet vaativat matkapuhelinten valmistajia kehittämään ja parantamaan puhelinten käyttöliittymiä ja käytettävyyttä.

2.1. Perinteisten käyttöliittymien ongelmat sokeilla käyttäjillä

Suurin ongelma perinteisissä matkapuhelinten käyttöliittymissä sokeille ja näkörajoitteisille käyttäjille on visuaaliseen näyttöön nojaava vuorovaikutus. Koska puhelimissa on rajallinen määrä näppäimiä, ei jokaiselle toiminnalle voi olla omaa näppäintään. Näppäinten toiminta muuttuu riippuen siitä, missä valikossa tai toiminnossa käyttäjä on, ja dynaaminen sisällön muuttuminen vaatii myös näyttöä tukemaan muuttuvia näppäinkomentoja.

Näppäinten yhtenäisyys numeroissa auttaa niiden käytössä myös ilman näköä, mutta muiden näppäinten toimintaa on vaikea hallita ilman näköä. Näppäinäänten ollessa päällä, voi numeroiden äänet oppia ja täten varmistua, että valittu numero on oikea. Muista näppäimistä ei voi olla varma, sillä jos käyttäjä onkin eri toiminnallisuudessa kuin luuli, ei oikean näppäimen painalluksesta ole hyötyä, sillä kontekstista ei ole varmuutta.

Matkapuhelimen näppäimillä on paljon eroja riippuen malleista ja merkistä. Jotta eri näppäimet olisi helppo erottaa tunnustelemalla, tulisi niiden olla selkeästi sijoiteltu, muotoiltu ja näppäimistön sisältää ns. sokea piste. Silfverberg [2003] onkin osoittanut, että mitä paremmat tuntovihjeet puhelimen näppäimistössä on, sitä vähemmän puhelimen käyttöön kuluu aikaa ja tapahtuu virheitä. Tutkimus tehtiin näkeville käyttäjillä, jotka käyttivät puhelinta siten, että näköyhteyttä laitteeseen ei ollut. Sokeilla käyttäjillä tehty tutkimus luultavimmin antaisi samansuuntaisia tuloksia. Tosin sokeiden käyttäjien sormien tuntoherkkyys voi erota näkevästä käyttäjästä, jotka eivät yleensä ole yhtä tottuneita erottelemaan muotoja ja vihjeitä sormilla ns. sokeassa tilanteessa.

Sokeille tarkoitettu matkapuhelimen käyttöliittymä ei voi hyödyntää näyttöä ja näkökykyä. Näin ollen muilla aisteilla on tärkeä osa vuorovaikutuksessa. Ääni ja tuntoaisti ovatkin tärkeässä asemassa matkapuhelinta käytettäessä ilman näköä. Tuntoaistiin perustuvia vuorovaikutustapoja on erilaisia ja yleisin lienee käsien ja sormien käyttö. Äänen ollessa osa vuorovaikutusta voidaan

tarkoittaa puhekäyttöliittymiä sekä muita, ei puheeseen perustuvia vuorovai-
kutustapoja. Vaihtoehtona on myös käyttää eri vuorovaikutustapojen yhdistel-
miä, joilla saavutetaan monia erilaisia käyttäjäryhmiä, kuten näkevät ja sokeat
käyttäjät.

3. Käyttöliittymiä sokeille ja näkörajoitteisille

Sokeiden ja näkörajoitteisten käyttäjien tarpeisiin on kehitetty jo kauan erilaisia
käyttöliittymiä varsinkin pöytätietokoneisiin. Useasti käyttöliittymä on toteu-
tettu puhekäyttöliittymän ja muiden tekstinsyöttötapojen yhdistelmänä, kuten
Dobrišekin tutkimusryhmän [2003] kehittämä informaation jakelusysteemi,
joka lukee tekstin näytöltä ja tukee niin näppäimistö- kuin puhesyötteitäkin.

Nykyään pc-tietokoneiden käyttöliittymien ollessa graafisia ruudunlukijat
eivät pysty esittämään kaikkea informaatiota näytöltä. Toisaalta ruudunlukijat
saattavat lukea epäoleellisia asioita, kuten sisältöön epäoleennaisesti liittyviä
kuvia tai muita graafisia komponentteja, jotka eivät ole käyttäjän kannalta
oleellisia varsinkaan auditiivisessa kontekstissa. Kaiken informaation esittävästä
ruudunlukijoista onkin menty pidemmälle ja kehitetty käyttöliittymiä mallinta-
maan graafisen käyttöliittymän rakennetta semanttisella tasolla. Edwards ja
muut [1995] ehdottavat useita eri tapoja mallintaa graafista käyttöliittymää
ilman visuaalista ulkoasua. Auditiivisia ikoneita voidaan käyttää eri kompo-
nenttien esittämisessä, jossa erilaiset äänet kertovat, onko kyse esimerkiksi
editoitavissa olevasta tekstikentästä. Valikkojen eri vaihtoehdot voidaan
mallintaa eri korkeudella olevilla äänillä, kun taas painikkeiden tekstit voidaan
lukea sellaisenaan. [Edwards et al., 1995] Myös Donker tutkimusryhmineen
[2002] loi kuuloaisteihin perustuvan kolmiulotteisen käyttöliittymän mallinta-
maan graafista käyttöliittymää. Sokeat käyttäjät eivät tarvitse tietoa siitä, minkä
näköinen käyttöliittymä on, vaan haluavat ymmärtää sen rakennetta ja täten
päästä käsiksi ominaisuuksiin, joiden avulla käyttäjän on helpompi ja
nopeampi päästä tavoitteisiinsa [Donker et al., 2002].

Matkapuhelimet eroavat kuitenkin pc-tietokoneista, joten niissä hyviksi
havaitut käyttöliittymät eivät välttämättä sovi matkapuhelimiin sellaisenaan.
Matkapuhelimen pieni näyttö ei useinkaan sisällä yhtä monimutkaisia ja
monipuolisia graafisia elementtejä ja ylipäättään matkapuhelimeissa oleva sisältö
on rajoitetumpi ja suppeampi. Lähempänä matkapuhelimia ovatkin erilaiset
mobiililaitteet, joista yleisimpiä sokeille suunnattuja käyttöliittymiä löytyy ns.
navigaatio- ja paikannuslaitteista.

Erityisesti sokeille ja näkörajoitteisille käyttäjille on kehitetty useita paikan-
nuslaitteita, joiden avulla heidän on helpompi liikkua vieraisissa paikoissa.
Koska kyse on laitteesta, joka auttaa liikkumisessa, on itsestään selvää, että

kyseiset laitteet ovat mobiileja eli langattomia ja kulkevat käyttäjän mukana. Holland ja muut [2002] ovat kehittäneet prototyypiversion GPS-paikannuslaitteesta, joka kulkee käyttäjän mukana ja on vähän huomiota vaativa. Sen käyttöliittymä perustuu ei-puheellisiin ääniin, jolloin käyttäjä voi samanaikaisesti kommunikoida puheella muiden ihmisten kanssa. Laite on pakattu reppuun ja sen lisäksi käyttäjällä on korvakuulokkeet.

Paljon erityisiä ja erillisiä laitteita vaativa käyttöliittymä ei kuitenkaan välttämättä sovi matkapuhelimiin, jolta yleensä odotetaan helppokäyttöisyyttä ja pientä kokoa. Sen vuoksi tämä tutkielma rajautuu tarkastelemaan käyttöliittymiä, jotka eivät vaadi suuria apuvälineitä. Nykyään jo yleistyneet korvakuulokkeet sekä erilliset näppäinlaitteet tekevätkin poikkeuksen, mutta kummassakaan tapauksessa erillinen laite on pieni kokoinen ja helppo kuljettaa mukana.

Seuraavat luvut on rajattu pääasiallisen vuorovaikutustyyppin mukaan, jonka avulla sokeat käyttäjät voivat matkapuhelinta käyttää. Useat menetelmät eivät kuitenkaan perustu vain yhteen vuorovaikutustyyppiin, joten päällekkäisyyksiä voi esiintyä. Ensiksi tarkastellaan erityisesti tuntoaistiin liittyviä vuorovaikutustapoja ja sen jälkeen kuuloaistiin perustuvia vuorovaikutustapoja.

4. Tuntoaistiin perustuva vuorovaikutus

4.1. Näppäimet ja näppäimistöt

Ensimmäinen tuntoaistiin liittyvä kokemus matkapuhelimen käyttäjällä on se, miltä laite tuntuu kädessä. Nykyisin matkapuhelimet on usein suunniteltu siten, että niitä on helppo käyttää myös yhdellä kädellä, joka parantaa laitteen hallintaa. Olennaista myös sokeilla käyttäjillä on näppäinten sijoittelu ja muoto eli niiden tunnistettavuus.

Silfverberg [2003] on tutkinut erilaisten matkapuhelinten näppäinten vaikutusta niiden käyttöön vuorovaikutustilanteissa, joissa näköä on käytetty rajoitetusti tai ei lainkaan. Hän huomasi, kuinka tuntoaistin huomioonottavissa näppäimistöissä on huomattava ero vertailunäppäimistöön. Tuntovihjeitä tarjoava näppäimistö vähensi virhepainallusten esiintymistä näön ollessa rajoitettua. Näppäimistön käyttö on myös nopeampaa ja koettiin helpommaksi kuin vertailussa ollut huonosti tuntoaistin huomioonottava näppäimistö. Tuntoaistin huomioonottamisesta on näin ollen hyötyä myös näkeville tilanteissa, joissa näkökyky on rajoitettua vuorovaikutustilanteessa.

Yleistynyt, laajalle levinnyt näppäimistö, 12-näppäimisen numeronäppäimistön [Silfverberg, 2003] käyttö on usein suhteellisen helppoa ja hallittavaa

ilman näkökykyä, sillä näppäinten standardisijoittelu, näppäinäät ja numeroiden rajallisuus edesauttaa niiden käytön oppimista. Muiden näppäinten hallinta on kuitenkin vaikeampaa, sillä tilaa muille näppäimille on rajoitetusti ja siksi ne toimivat useissa eri tilanteissa erilailla kontekstista riippuen. Yksi vaihtoehto ratkaista muiden näppäinten ongelma on käyttää matkapuhelimissa QWERTY-tyyppisiä näppäimistöjä, jotka ovat nykyään yleisiä myös matkapuhelimissa [Silfverberg, 2003].

QWERTY-näppäimistö matkapuhelimessa tarjoaa useampia näppäimiä, jotka voivat olla tiettyyn toiminnallisuuteen sidottuja. Esimerkiksi kirjaimet saavat omat näppäimensä, joka voi olla suuri parannus sokeille käyttäjille. Näin ollen myös yksittäisillä, toisistaan erotelluilla näppäimillä voi olla toiminnallisuutta, joka ei vaihdu kontekstista riippuen, vaan pysyy aina samana.

Toisaalta QWERTY-näppäimistö aiheuttaa myös ongelmia, sillä suuri määrä näppäimiä pienessä tilassa voi aiheuttaa puutetta niiden tuntovihjeissä [Silfverberg, 2003]. Pieniä näppäimiä on myös vaikea käyttää sormilla kuten tietokoneen näppäimistöä 10-sormijärjestelmällä. Matkapuhelimen näppäimistön käytössä tulee myös ottaa huomioon se, että laitetta joutuu useimmiten pitämään kädessään. Näin ollen vapaiksi sormiksi jäävät usein vain peukalot [Silfverberg, 2003].

4.2. Tuntolevyt

Tuntolevyjä (touch pads) on yleisesti käytetty kämmentietokoneissa ja mobiililaitteissa, joissa ohjaus tapahtuu kynän tapaisella tikulla. Tällainen hiirtä korvaava osoitin ei kuitenkaan sovellu sokeille tarkoitettuihin laitteisiin, sillä ne edellyttävät näköä vaativaa tarkkaa kohdistusta. Erillisen syötelaitteen käyttö ei sovellu välttämättä myöskään matkapuhelimen konseptiin, sillä sitä on oltava mahdollista käyttää helposti myös liikkeessä.

Tuntolevyjä on kuitenkin kehitetty myös tilanteisiin, jossa näkö on rajoitettu tai sitä ei ole tarkoitus käyttää laitteen ohjaamisessa. Useasti tällaisia vuorovaikutustapoja on tutkittu erityisesti kannettavissa laitteissa ja liikkuvissa tilanteissa, eli tilanteissa, joissa näkevä käyttäjä on liikkeessä eikä pysty katsomaan laitetta [Pirhonen et al., 2002]. Koska näköä ei ole tarkoitus käyttää vuorovaikutuksessa, perustuu ohjaus kosketusnäytön avulla sormin tehtäviin metaforisiin eleisiin, jotka on helpompi tehdä ilman näköä toisin kuin esimerkiksi kirjoitus [Brewster et al., 2003].

Brewster ja muut [2003] ovat tutkineet mobiililaitteiden käyttöä mm. eleiden avulla ilman näköä. He osoittivat, että valintaa ohjaavat ja tukevat äänet paransivat eleillä tehtyjen valintojen tarkkuutta. Myös Pirhonen ja muut [2002] käyttivät tutkimuksessaan yksinkertaisia ääniä, jotka osoittivat valinnan oikeellisuudesta ja eleiden tunnistuksesta. Brewsterin tutkimuksessa käytetyt dynaa-

miset äänet ohjasivat käyttäjän eleiden tekoa eri vaiheiden mukaan. Näin ollen äänimaailmasta oli helpompi tunnistaa eri vaiheet toisin kuin jos äänet olisivat perustuneet samaan piippaus-ääneen, joka ilmaisee syötteen onnistumista, syötteen tunnistusta tai valikossa liikkumista askel kerrallaan.

Tällaiset tuntolevyllä toimivat vuorovaikutustekniikat toimivat lähinnä sovelluslaitteissa, joissa on pienehkö määrä informaatiota. Pirhosen ja muiden [2002] tutkimuksessa kohteena oli PDA-musiikkisoitin, jonka valikossa oli vain viisi toimintoa. Myös Brewsterin ja muiden [2003] tutkimuksessa laitteen toimintojen määrä oli rajattu, tosin niitä oli jopa kaksitoista. Heidän tutkimuksensa tuloksista oleellista on, että tuntoon ja eleisiin perustuva vuorovaikutus vaatii ehdottomasti jonkinlaisen ilmaisun syötteen oikeellisuudesta, esimerkiksi helposti tunnistettavan äänen. Mahdolliset virhetapaukset lisääntyvät, mikäli laite ei ilmaise käyttäjälle valintaa tai eleen tunnistusta. Liian heikosti piirrettyä elettä ei välttämättä tunnisteta käyttäjän antamaksi syötteeksi.

Vaikka tuntolevyyn perustuva eleiden käyttö voi olla hankala matkapuhelimen käyttöliittymäksi, Pirhonen ja muut [2002] todistivat, että kannettava laite voi olla toimiva myös vaihtoehtoisilla käyttöliittymillä, jotka eivät perustu näköön. Myös Brewster ja muut [2003] saivat tutkimuksessaan positiivisia tuloksia eleiden käytöstä mobiililaitteen käyttöliittymässä. Eleiden käytössä matkapuhelimen käyttöliittymänä on monia hankaluuksia. Mitä laajempi on matkapuhelimen toiminnallisuus, sen enemmän käyttäjän tulee muistaa erilaisia eleitä. Ihmisen työmuistin kapasiteetti on kuitenkin rajallinen [Baddeley, 2004]. Suuri kognitiivinen kuormitus häiritsee myös muita kognitiivisia toiminnallisuuksia [Matthews et al., 2000], joita käyttäjällä saattaa olla samanaikaisesti meneillään matkapuhelimen käytön kanssa.

5. Kuuloaistiin perustuva vuorovaikutus

Erilaiset puhe- ja muihin ääniin perustuvat käyttöliittymät ovat luonteva ratkaisu sokeille käyttäjille. Puheikäyttöliittymiä voi olla erilaisia, mutta yleisesti puheikäyttöliittymissä on puhesyntetisaattori, joka lukee ääneen informaation, sekä puheentunnistusjärjestelmä, jonka avulla voidaan antaa puhesyötteitä. Muihin ääniin kuin puheeseen perustuva vuorovaikutus voi olla erilaisista äänimerkeistä tai ns. kuuloikoneista koostuvia vihjeitä. Myös spatiaalisia vihjeitä voidaan käyttää hyväksi, jolloin äänimaailman kautta voidaan viestittää objektien sijaintia kolmiulotteisessa ympäristössä.

Tässä luvussa käydään läpi tarkemmin kuuloaistiin perustuvia vuorovaikutustapoja käyttöliittymissä. Ensin tarkastellaan puheikäyttöliittymiä ja sen jälkeen muihin ääniin kuin puheeseen perustuvaa vuorovaikutusta.

5.1. Puhekäyttöliittymät

Puhe vuorovaikutustapana on tehokas, onhan puhe ihmisille luonnollinen tapa kommunikoida. Puhekäyttöliittymissä on kuitenkin omat ongelmansa, sillä ne eivät toimi yhtä joustavasti ja luotettavasti kuin ihmisten keskinäinen kommunikointi. Puhekäyttöliittymissä on vielä puutteita ja matkapuhelimen kaltaisissa pienissä laitteissa koko voi rajoittaa laskentatehoja. Toisaalta puhekäyttöliittymät sopivat juuri matkapuhelimiin, joissa on jo valmiiksi mikrofoni ja jonkinlainen kaiutin.

Puhekäyttöliittymäominaisuuksia on jo markkinoilla useilla suurilla matkapuhelinvalmistajilla kuten Nokialla [Nokia], Motorolalla [Motorola] ja Sony Ericssonilla [Sony Ericsson]. Puhekäyttöliittymää ei kuitenkaan tietääkseni ole käytössä yhdessäkään matkapuhelimessa ensisijaisena vuorovaikutusmuotona, sillä sen käytössä on useita ongelmia. Ensinnäkin, matkapuhelinten kannalta puhekäyttöliittymän käyttö julkisilla paikoilla ei ole kovin yksityistä, sillä vieressä olijat kuulevat kaiken vuorovaikutuksen. Usein voi tulla tilanteita, joissa ei voi käyttää puhekäyttöliittymää julkisella paikalla joko yksityisyyden suojan tai häiritsevyyden takia. Toisaalta korvakuulokkeet voivat auttaa tiedon yksityisyyden suojaamisessa ja lähellä suuta sijaitseva mikrofoni mahdollistaa hiljaisen puheen syötön. Mutta kovin hiljaisella äänellä puhuminen tai kuiskaminen voi aiheuttaa ongelmia puheen tunnistuksessa.

Matkapuhelimen pieni koko vaikuttaa myös sen sisällön esittämiseen. Suhteellisen pieni näyttö pakottaa graafisen käyttöliittymän esitystavan olevan suhteellisen yksinkertainen ilman muita kuin sisällön kannalta oleellisia komponentteja. Näin ollen puhekäyttöliittymä ruudunlukijan kaltaisena lienee helpompi soveltaa matkapuhelimiin kuin esimerkiksi pc-tietokoneisiin, joiden graafisen käyttöliittymän sisältö on monipuolisempaa, vaihtelevampaa ja kompleksisempaa ruudunlukijan kannalta (ks. luku 3). Ruudunlukijan kaltaisen puhekäyttöliittymän voisi näin ollen liittää yleisesti käytetyn visuaalisen käyttöliittymän rinnalle.

Puheentunnistus on haastava teknologia, jota on tutkittu ja kehitetty paljon. Tämän tutkielman tarkoituksen ei kuitenkaan ole ottaa kantaa puheentunnistuksen haasteisiin tai teknologioihin, vaan keskittyä olemassa olevien teknologioiden käytettävyyteen ja sopivuuteen matkapuhelinkontekstissa. Puhekäskyt voivat olla samoja, joita graafisessa käyttöliittymässä on jo tarjolla. Lyhyt ja ytimekäs käsky voi olla helpompi käyttää ja tunnistaa kuin pitkä, vapaamuotoinen dialogi.

5.1.1. Puhekäyttöliittymien käytettävyys

Pitt ja Edwards [1996] löysivät useita suunnitteluun vaikuttavia seikkoja tutkiessaan puhekäyttöliittymien käytettävyyttä sokeilla käyttäjillä. Tutkimus tehtiin pc-tietokoneella ruudunlukijajärjestelmällä käyttäen näppäimistöä syötelaitteena, mutta käytettävyyseikat antavat suuntaa myös puhekäyttöliittymän suunnittelussa matkapuhelimissa, varsinkin jos järjestelmän tuotos vastaa ruudunlukijan kaltaista tuotosta.

Puhe tuotoksena käyttöliittymissä kuormittaa enemmän muistia kuin näkö- tai tuntoaisti [Pitt and Edwards, 1996], minkä vuoksi esimerkiksi valikot eivät saisi vaatia käyttäjältä liikaa resursseja informaatiota mielessä pitäessään. Myös liian pitkä dialogi tai puhe vaikeuttaa viestin ymmärtämistä. Tutkimuksen mukaan myös vähemmän tärkeää informaatiota tulisi välttää, sillä se vaikeuttaa tärkeän informaation muistamista. Puheen prosodialla, ääntämisellä ja puheen laadulla on myös vaikutuksensa ymmärrettävyyteen. Tärkeää on myös muistaa säilyttää kontrolli käyttäjällä, eli dialogi tulisi voida pysäyttää ja toistaa tarvittaessa, mikäli asia ei heti tullut ymmärretyksi. [Pitt and Edwards, 1996]

5.1.2. Puhe informaationsyötteenä

Puhekäyttöliittymissä puhe syötteenä asettaa myös haasteita käyttöliittymän suunnitteluun ja käyttöön. Sawhneyn ja Schmandtin [2000] mukaan varsinkin julkisilla paikoilla puhekäyttöliittymällä on omat ongelmansa. Puheen käyttö voi olla hidasta, varsinkin jos tunnistus ei heti toimi ja käyttäjän täytyy yrittää uudelleen. Puhesyötteen komennot täytyy muistaa ulkoa tai ainakin tuntea käyttöliittymän konteksti hyvin, jotta käyttäjä pystyy antamaan niitä. Myös meluisa ympäristö voi vaikeuttaa puheen tunnistuksen tarkkuutta, joka usein on hyvin altis virheille. [Sawhneyn and Schmandt, 2000]

5.1.3. Puhekäyttöliittymän kontrollointi

Puhekäyttöliittymä, joka ottaa vastaan puhesyötteitä, tarvitsee usein jonkinlaisen kontrollin, koska tunnistus tulisi aloittaa. Käyttäjän kannalta helppokäyttöinen järjestelmä ei tarvitsisi minkäänlaista vihjettä, koska se "kuuntelee" käyttäjää, mutta tämä voi aiheuttaa useita ongelmia. Oletetaan, että tällainen puhekäyttöliittymä olisi matkapuhelimessa. Jos systeemi jatkuvasti kuuntelisi käyttäjää, olisi käyttäjän mahdotonta esimerkiksi puhua puhelua ilman, että järjestelmä yrittäisi tehdä tunnistuksia ja mahdollisia operaatioita komennon tunnistettuaan. Tunnistusjärjestelmän voisi tietysti sammuttaa automaattisesti puheluiden ajaksi, mutta seuraava ongelma syntyy tilanteista, joissa matkapuhelin yrittää tunnistaa käyttäjän tai muiden ihmisten puhetta tilanteissa, joissa puhetta ei ole osoitettu käyttöliittymän ohjaamiseen. Tällaisesta tilan-

teesta puhelin voisi tehdä tunnistuksia ja suorittaa komentoja täysin käyttäjän tietämättä niistä.

Sawhney ja Schmandt [2000] ehdottavat, että tällaisen jatkuvasti puhetta monitoroivan järjestelmän tulisi voida asettaa "kuuntelu-" tai "nukkumistilaan", jolloin tunnistuksen voisi asettaa väliaikaisesti pois päältä. Tunnistusjärjestelmän ollessa päällä järjestelmä pyrkii tunnistamaan mahdollisia komentoja ja reagoi käyttäjälle puhesyötteellä vain tunnistuksen tehtyään. Tämä vähentää oleellisesti häiritsevää ja epäoleellista dialogia, mutta ei auta mahdollisten virhetunnistusten eliminoinnissa. Siksi Sawhney ja Schmandt ehdottavat askeleen pidemmälle vietyä tunnistusjärjestelmää, *kontekstuaalista* tunnistusta. Tämä potentiaalinen järjestelmä menisi päälle vain silloin, kun järjestelmä todennäköisimmin odottaa käyttäjältä komentoja. Tällaisia tilanteita voisi syntyä esimerkiksi tekstiviestin saapuessa. Kontekstuaalisesta tunnistusjärjestelmästä ei kuitenkaan ole tehty ainakaan Sawhneyn ja Schmandtin osalta järjestelmällistä tutkimusta, joten sen toimivuudesta käytännössä ei ole osoituksia. Jatkuvasti päällä olevissa tunnistusjärjestelmissä on varsinkin matkapuhelinten kannalta otettava huomioon, että kyseinen järjestelmä vaatii resursseja koko ajan päällä ollakseen. Pienessä laitteessa ei välttämättä niitä ole tarpeeksi.

Toinen, enemmän käyttäjän kontrolloinnissa oleva tunnistusjärjestelmä toimii vasta, kun käyttäjä osoittaa aloittavansa puhesyötteen erillisellä näppäimellä (ns. push-to-talk -toiminto). Järjestelmä tunnistaa automaattisesti syötteen loppumisen. Tämä järjestelmä ei kuluta resursseja niin paljon kuin jatkuvasti päällä oleva tunnistusjärjestelmä. Lisäksi sen aiheuttamien epäoleellisten ja tahattomien komentojen tunnistus on epätodennäköisempää. Sawhneyn ja Schmandtin [2000] mobiilijärjestelmässä push-to-talk -järjestelmä antaa puhepalautteen myös ei-tunnistetusta komennosta.

Jonkinlaisen palautteen anto myös virheellisestä ja ei-tunnistetusta syötteestä onkin oleellista käytettävyyden kannalta, sillä muutoin käyttäjä ei voi tietää tunnistuksen tilaa. Pelkkä puhe käyttöliittymän vuorovaikutusmuotona ei siis ole riittävä, vaan käytettävyyden kannalta on oleellista, jos järjestelmässä on puheen lisäksi myös jonkinlaisen tuntoon perustuvan syötteen mahdollisuus kuten push-to-talk -järjestelmässä [Sawhney and Schmandt, 2000].

5.1.4. Virheiden käsittely

Sawhney ja Schmandt [2000] totesivat, että puhe käyttöliittymä voi olla hidaskuorovaikutusmuoto varsinkin virheiden sattuessa. Puhe onkin hyvin tehokas vuorovaikutusmuoto niin kauan, kun toinen osapuoli osaa tulkita sitä. Puheen

tunnistus ei kuitenkaan ole usein kovin tarkkaa, ja siksi virhetilanteista selviämiseen tulisi kiinnittää huomiota.

Virhetilanteiden hallinta voi perustua jatkuvasti uuteen yritykseen, mutta jonkinlainen vaihtoehtoinen virheestä toipuminen voisi tuoda järjestelmään tehokkuutta ja parantaa käytettävyyttä. Vaihtoehtoisen hyvän tavan löytämiseen tarvitaan systemaattista tutkimusta, jotta voidaan varmistua sen sopivuudesta puhekäyttöliittymään ja erityisesti sokeille käyttäjille.

5.2. Ei-puheelliseen ääneen perustuva vuorovaikutus

Puhe ei ole ainoa vuorovaikutustapa käyttöliittymässä sisällön esittämiseen. Kuten kohdassa 4.2 todettiin, muita ääniä kuin puhetta voidaan käyttää käyttöliittymissä vahvistamaan valintoja, ilmaisemaan syötteen oikeellisuus tai laitteen tila. Tällaisella puheettomalla äänimaailmalla voidaan huomattavasti parantaa matkapuhelinten käytettävyyttä sokeilla käyttäjillä. Monissa markkinoilla olevissa matkapuhelimissa on jo käytössä näppäinäniä sekä muita varoitus- ja ilmoitusääniä.

Monet markkinoilla olevat älypuhelimet käyttävät graafista käyttöliittymää, jossa käytettävyyks on otettu huomioon useilla visuaalisilla ratkaisuilla. Matkapuhelinten käyttöliittymien graafisuus ei kuitenkaan vastaa pc-tietokoneita, joissa on paljon erilaisia elementtejä ja objekteja manipuloitavissa sekä jäsentämässä ja hahmottamassa toiminnallisuutta. Matkapuhelinten käyttöliittymien sisältö on kuitenkin usein esitetty hierarkkisesti: valikot saattavat sisältää alavalikoita ja eri toimenpiteet, ominaisuudet ja ohjelmat jaoteltu jonkin kategorian mukaan. Tämä jaottelu on tarpeen varsinkin, kun sisältö on suurehko ja ominaisuuksia on paljon. Samalla kuitenkin sokeiden käyttöliittymän käyttö hankaloituu ilman helppokäyttöistä käyttöliittymää. Hierarkioihin saattaa eksyä helposti ja toiminnallisuudet eivät välttämättä olekaan samankaltaisia eri ominaisuuksissa tai sovelluksissa.

Kuten luvussa 3 todettiin, sokeille käyttäjille on kehitetty vaihtoehtoisia tapoja esittää graafisen käyttöliittymän ominaisuudet tietokoneessa. Graafisten käyttöliittymien teho perustuu niiden tapaan esittää komponentit ja tapahtumat käyttäjän sisäisten mallien mukaan. Pyrkimyksenä onkin pystyä esittämään käyttöliittymän semanttiset mallit sokeille järkevästi ilman visuaalista maailmaa. [Edwards et al., 1995]

Edwardsin ja muiden [1996] mukaan mm. ääni-ikoneilla, auditorisilla, kuvaa korvaavilla vihjeillä, voidaan esittää objektien tyyppi. Ääni-ikoneilla voitaisiin erottaa esimerkiksi eri toiminnallisuuksia valikossa (puhelinmuistio, viestit, sähköpostit jne.), henkilön eri puhelinnumeroita (koti, työ, matkapuhelin) ja muita luokitteluita. Niiden avulla voitaisiin myös viestittää, johtaako valinta esimerkiksi valikosta uuteen alavalikkoon vai aukeaako uusi sovellus.

Myös erilaiset pysähtyneet tilat, kuten sovelluksen lataus tai verkon haku voitaisiin osoittaa äänivihjein.

5.2.1. Kolmiulotteinen äänimaailma

Visuaalinen käyttöliittymä voidaan korvata myös kolmiulotteisella äänimaailmalla, joka käyttää hyväkseen avaruudellista hahmottamista. Brewster ja muut [2003] toteuttivat mobiililaitteen potentiaalisen käyttöliittymän, joka pohjautui kolmiulotteiseen piirakkavalikkoon. Valinnat suoritettiin pään nyökkäyksillä egosentrisessä tai eksosentrisessä ääniympäristössä. Egosentrinen ääniympäristö, joka osoittautui tehokkaammaksi, levittäytyi kuulijan ympärille neljään suuntaan (eteen, sivuille ja taakse). Eksosentrinen ääniympäristö sijoittui vain kuulijan eteen neljään eri suuntaan 40 asteen välein. Nyökkäykset tuli kohdistaa aina haluttuun suuntaan, josta valinta suoritettiin. Mobiililaitte edellytti kuulokkeita, jotka esittivät kolmiulotteisen ääniympäristön ja sisälsivät pään liikkeiden tunnistimen.

Brewster ja muut saivat suhteellisen positiivisia tutkimustuloksia käyttöliittymän käyttökokeesta. Kolmiulotteisen ääniympäristön ja nyökkäysten käyttö valinnoissa eivät hidastaneet paljoakaan käyttäjien kävelynopeutta. Lisäksi osa käyttäjistä piti käyttöliittymää helppokäyttöisenä. Vaikka kuulokkeiden käyttö julkisesti on nykyään yleistä, oli tutkimuksen järjestelmässä suhteellisen isot ja näkyvät kuulokkeet. Niiden suuri koko voi tuntua epämu-kavalta käytössä ja jopa estää kuulemasta muuta ympäristöä. Lisäksi pään liikkeiden tunnistusjärjestelmän tarkkuutta tulisi parantaa, jotta käyttäjien ei tarvitsisi tehdä kovin isoja liikkeitä [Brewster et al., 2003]. Järjestelmän laitteisto ja pään nyökyttäminen voivat tuntua joidenkin käyttäjien mielestä epämu-kavilta ja vaikuttaa negatiivisesti järjestelmän hyväksyntään. Lisäksi järjestelmä rajoittui neljään valittavissa olevaan komponenttiin, joka voi aiheuttaa ongelmia matkapuhelimen kontekstissa. Valinta- tai elementtimahdollisuuksia tulisi olla useampia.

Myös Sawhney ja Schmandt [2000] käyttivät kolmiulotteista äänimaailmaa mobiililaitteessaan. Hyvin suunniteltuna se voi parantaa kuulomuistin kapasiteettia ja mahdollistaa monien äänien samanaikaisen esittämisen käyttäen hyväksi cocktail-party-ilmiötä [Sawhney and Schmandt, 2000]. Monista äänistä ihminen havainnoi itselleen tärkeitä asioita sekä huomioi esimerkiksi muutoksia ja äkillisiä hälytysääniä. Toisaalta varsinkin hälyisessä ympäristössä kolmiulotteinen äänimaailma voi aiheuttaa suurempaa kognitiivista kuormitusta äänien havainnoinnissa [Sawhney and Schmandt, 2000].

Sawhneyn ja Schmandtin [2000] sovellus käytti erityisesti mobiililaitteeseen suunniteltua niskan ympärille asetettavaa äänilaitetta (Soundbeam Neckset). Laite ei peittänyt korvia, joten se mahdollistaa myös muun ympäristön

kuulemisen. Erillisen uuden laitteen käyttö voi kuitenkin olla kynnyksellä käyttäjille. Tutummilla kuulokkeilla laite on myös käytettävä, mutta siinä tapauksessa korvat on peitetty eikä ympäristön muiden äänien kuuleminen ole yhtä helppoa.

6. Potentiaalisia käyttöliittymiä näkörajoitteisille

Tässä luvussa esitetään kaksi potentiaalista, edellisten lukujen ehdotuksien mukaan rakennettua käyttöliittymää matkapuhelimiin, jotka voisivat sopia sokeille käyttäjille. Ensimmäinen ehdotus on puheeseen perustuva käyttöliittymä. Tämä malli sopii myös näkeville käyttäjille, jotka voivat hyödyntää puheominaisuuksia tilanteissa, joissa näkö on rajoitettua. Puheikäyttöliittymän käyttö julkisissa tilanteissa ei kuitenkaan suojaa yksityisyyttä. Sitä voidaan kuitenkin parantaa esimerkiksi kuulokkeita käyttämällä, mutta edelleen puhe-
käskyt täytyy antaa julkisesti muiden kuulla.

Toinen ehdotus on matkapuhelimen käyttöliittymästä, jonka vuorovaikutus pohjautuu tunto- ja kuuloaistiin. Se sopii hyvin myös julkiseen käyttöön, sillä ei aiheuta ongelmia yksityisyydessä puheikäyttöliittymien tapaan. Toisaalta tuntoaistin ja etenkin ei-puheellisten äänten vuorovaikutustapoja ei ole vielä tutkittu kovin paljon eikä kokemuksia arkielämässä ole pääsyt syntymään. Tutkimukset ovat kuitenkin lupaavia, joten potentiaalia näissä uusissa vuorovaikutustavoissa on.

6.1. Puheeseen perustuva käyttöliittymä

Potentiaalinen ja käytettävä matkapuhelin voisi pitää sisällään hyvän näppäimistön lisäksi puheikäyttöliittymäominaisuudet. Puhelimen näppäimistön tuntovihjeet tulisi olla tarpeeksi selkeät, joita ovat selkeä muotoilu ja erotettavuus muista näppäimistä. Näppäinäännet voisi olla erotettuna toisistaan, kuten myös sisältö ääni-ikoneineen. Myös eri tilat voisi erotella äänivihjein.

Puhelinta voisi ohjata näppäimillä äänivihjeiden avulla. Sen lisäksi puhelimessa olisi puhesyntetisaattori, joka lukisi tekstisisällön. Näin ollen käytön ei tarvitse perustua äänivihjeiden ja näin ollen muistin varaan. Käyttöliittymässä olisi myös puheentunnistusjärjestelmä, jota voi kontrolloida puhenäppäimellä (push-to-talk -järjestelmä). Eri toiminnot ja sovellukset tottelisivat puheentunnistusta ja sen lisäksi syötteet, kuten nimen lisääminen ja muu kirjoitus olisi myös mahdollista puheella.

Kyseinen ehdotus ei vaadi puhelimelta lisävarusteita mahdollisten kuulokkeiden lisäksi. Ominaisuudet sopisivat myös muille käyttäjille kuin sokeille, sillä myös visuaalisen käyttöliittymän olemassaolo on mahdollista. Haittoina voisi olla puheikäyttöliittymän laatu ja vaatimukset: pystytäänkö nykytekno-

logialla rakentamaan tarpeeksi luotettava ja tarkka puheentunnistusjärjestelmä sekä syntetisaattori, joka on ymmärrettävä. Matkapuhelin laitteena ei kuitenkaan mielellään saa olla kovin iso tai painava.

6.2. Tuntoaistiin ja ääniin perustuva käyttöliittymä

Tämä matkapuhelin toimisi sormin piirretyillä eleillä ja antaisi palautteen joko puheena tai ei-puheellisina ääniä. Matkapuhelimessa tulisi olla kosketusnäyttö, joka voisi pitää sisällään myös jonkinlaisen visuaalisen käyttöliittymän. Lisäksi siinä voisi olla ainakin joitakin tärkeimpiä näppäimiä puhelintoimintoihin, mahdollisesti jopa kaikki numeronäppäimet.

Tätä käyttöliittymää olisi tarkoitus ohjata eleillä, mutta sitä olisi myös mahdollista ohjata muutamilla perinteisillä näppäimillä. Käyttöliittymässä olisi voimakas ja tehokas äänimaailma, joka ohjaisi valintoja ja auttaisi navigoinnissa. Äänimaailman avulla käyttäjä tietäisi, missä toiminnossa on, ja onko hänen valintansa tunnistettu ja oikea.

Tämän käyttöliittymän vahvuuksia on sen huomaamaton käyttö julkisilla paikoilla, sillä matkapuhelimen tuntolevyä voisi käyttää vaikka vyötärölle kiinnitettynä. Heikkouksina voi olla vaikea käytettävyyys varsinkin alussa ja käytön vaatima eleiden ja kuulovihjeiden muistaminen.

7. Yhteenveto

Matkapuhelinten käyttöliittymien käyttö sokeiden ja näkörajoitteisten keskuudessa vaatii ehdottomasti käytettävyytutkimusta. Lisäksi mahdollisten uusien vuorovaikutus- ja käyttöliittymätapojen suunnittelu kaipaava järjestelmällistä tutkimusta. Matkapuhelimet ja muut kannettavat laitteet ovat nykyään tutkijoiden mielenkiinnon kohteina, ja uusia vuorovaikutustapoja on jo tutkittu. Silti tutkimusta ei ole tehty juurikaan sokeiden käyttäjien näkökulmasta. Käytettävyytutkimusten avulla voidaan ennen kaikkea selvittää sokeiden käyttäjien tarpeet, jotka ovat oleellinen lähtökohta suunnittelussa.

Kun suunnitellaan uusia käyttöliittymiä erilaisille käyttäjäryhmille, eivät jonkun ryhmän tutkimustulokset välttämättä päde toisen käyttäjäryhmän osalta. Sokeille käyttäjille on jo kauan kehitetty tietokoneisiin vaihtoehtoisia käyttöliittymiä korvaamaan graafista käyttöliittymää. Aina ei kuitenkaan tiedetä, mikä olisi paras vaihtoehto. Esimerkiksi Donker ja muut [2002] epäilevät äänikoneiden vaikeakäyttöisyyden syyksi sokeiden parissa sitä, että sokeiden käyttäjien sisäiset mallit eivät vastaa näkevien käyttäjien malleja, joita suunnittelijat edustivat.

Sokeilla käyttäjillä on varmasti erilaisia tarpeita kuin näkevilla käyttäjillä koskien matkapuhelinten käyttöliittymää, mutta silti monet eri ryhmät voivat

hyötyä aivan uudentlaisista käyttöliittymistä. Tämä voi olla tärkeä osatekijä laitteiden saatavuuteen markkinoilla. Toisaalta, eri käyttäjäryhmien käyttäessä samoja malleja, voi se aiheuttaa vaikeuksia käytettävyyden kannalta. Juuri esimerkiksi sokeiden ja näkevien käyttäjien erilainen tapa hahmottaa maailmaa voi olla olennaista käyttöliittymän suunnittelussa ja voi näin ollen aiheuttaa huonoa suunnittelua eri käyttäjien kannalta.

Erilaiset ja uudet käyttöliittymät voivat osoittautua tutkimuksissa lupaaviksi, mutta todellisuus niiden käytössä voi olla toista. Vaikka tutkimus osoittaisi tehokkuutta ja helppokäyttöisyyttä, voi uuden vuorovaikutusmuodon vakiintuminen käyttäjien keskuudessa estyä muista syistä. Esimerkiksi matkapuhelinten julkinen käyttö herättää kysymyksen hyväksyttävyydestä. Lumsden ja Brewster [2003] kuvaavat, kuinka erikoiset puettavat laitteet voivat tuntua käyttäjistä epämukavilta varsinkin julkisilla paikoilla. Myös julkisilla paikoilla puhuminen puheikäyttöliittymään ei välttämättä ole sopivaa ja esimerkiksi eleiden kuten pään nyökkäilyt voivat tuntua käyttäjästä yhtä oudoilta kuin erikoisten laitteiden pukeminen ylleen.

Kaiken kaikkiaan monet lupaavat tutkimukset osoittavat, että uusia vuorovaikutustapoja käyttöliittymiin kehitetään ja halutaan kehittää. Matkapuhelimet laitteena ovat yleisiä ja leviävät yhä enemmän uusien käyttäjäryhmien pariin. Perinteinen visuaaliseen käyttöliittymään pohjautuva käyttöliittymä ei enää riitä, sillä niin matkapuhelinten käyttötavat kuin käyttäjäryhmätkin ovat yhä monimuotoisempia.

Viiteluettelo

- [Baddeley, 2004] Alan D Baddeley, The psychology of memory. In: A.D. Baddeley, M.D.Kopelman and B.A. Wilson (eds), *The Essential Handbook of Memory Disorders for Clinicians*. John Wiley & Sons, 2004, 1-13.
- [Brewster et al., 2003] Stephen Brewster, Joanna Lumsden, Marek Bell, Malcolm Hall and Stuart Tasker, Multimodal 'eyes-free' interaction techniques for wearable devices. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing System* 5 (2003), ACM 473-480.
- [Dobrišek et al., 2003] Simon Dobrišek, Jerneja Gros, Boštjan Vesnicer, Nikola Pavešić and France Mihelič, Evolution of the information-retrieval system for blind and visually-impaired people. *International Journal of Speech Technology* 6 (2003), 301-309.
- [Donkel et al., 2002] Hilko Donker, Palle Klante and Peter Gorny, The design of auditory user interfaces for blind users. In: *Proc. NordiCHI*, (Oct. 2002), 19-23.

- [Edwards et al., 1995] W. Keith Edwards, Elizabeth D. Mynatt and Kathryn Stockton, Access to graphical interfaces for blind users. *interactions* 2 (Jan. 1995), 56-67.
- [Holland et al., 2002] Simon Holland, David R. Morse and Henrik Gedenryd, AudioGPS: spatial audio navigation with a minimal attention interface. *Personal and Ubiquitous Computing* 6 (Jan. 2002), 253-259.
- [Howell et al., 2005] Mark Howell, Steve Love and Mark Turner, Spatial metaphors for a speech-based mobile city guide service. *Personal Ubiquitous Computing* 9 (2005), 32-45.
- [Hub et al., 2003] Andreas Hub, Joachim Diepstraten and Thomas Ertl, Design and development of an indoor navigation and object identification system for the blind. In: *Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility Assets '04* (Sep. 2003), 147-152.
- [Lumsden and Brewster, 2003] Joanna Lumsden and Setphen Brewster, A Paradigm shift: alternative interaction techniques for use with mobile and wearable devices, In: *The 13th Annual IBM Centers for Advanced Studies Conference (CASCON'2003)* (Oct. 2003).
- [Matthews et al., 2000] Gerald Matthews, D R Davies, Stephen Westerholm and Rob B Stammers, *Human Performance: Cognition, Stress and Individual Differences*. Psychology Press (UK), 2000.
- [Motorola] Motorola, <http://www.motorola.com>
- [Nokia] Nokia, <http://www.nokia.com>
- [Pitt and Edwards, 1996] Ian J. Pitt and Alistair D.N. Edwards, Improving the usability of speech-based interfaces for blind users. In: *Proceedings of the Second Annual ACM Conference on Assistive Technologies* (1996), 124-130.
- [Pirhonen et al., 2002] Antti Pirhonen, Stephen Brewster and Christopher Holguin, Speech, audio, gesture: gestural and audio metaphors as a means of control for mobile devices. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves* 4 (2002), ACM 291-298.
- [Sawhney and Schmandt, 2000] Nitin Sawhney and Chris Schmandt, Nomadic radio: speech and audio interaction for contextual messaging in nomadic environments, *ACM Transactions on Computer-Human Interaction* 7 (Sep. 2000), 353-383.
- [Silfverberg, 2003] Miika Silfverberg, Using mobile keypads with limited visual feedback: implications to handheld and wearable devices. *Lecture Notes in Computer Science* 2795 (2003), 76 – 90.
- [Snellman, 2005] Kalle Snellman, Mobiilipalvelumarkkinat Suomessa 2004. *Liikenne- ja viestintäministeriön julkaisuja* 34/2005, (huhtikuu 2005).

Saatavilla http://www.mintc.fi/oliver/upl497-Julkaisu%2034_2005.pdf
(8.11.2005).

[Sony Ericsson] Sony Ericsson, <http://www.sonyericsson.com>

[Tikkanen et al., 2004] Terho Tikkanen, Pertti Haikonen ja Markku Kääriäinen,
Kiinteiden verkkojen tulevaisuus. *Liikenne- ja viestintäministeriön julkaisu*
39/2004, (kesäkuu 2004). Saatavilla http://www.mintc.fi/oliver/upl757-39_2004.pdf (8.11.2005).

ERP-järjestelmien implementoinnin ongelmat

Antti Jämsén

Tiivistelmä.

ERP-järjestelmät (enterprise resource planning) eli toiminnanohjausjärjestelmät ovat integroituja tietojärjestelmäkokonaisuuksia. Valtaosa suurista- ja keskisuurista yrityksistä on implementoinut ERP-järjestelmän mm. paremman tiedon laadun ja tuotannon tehokkuuden vuoksi. Toiminnanohjausjärjestelmän laaja integroituneisuus ja etenkin sen vaikutus yrityksen kulttuurin sosiaalisiin rakenteisiin tekee implementointiprojektista hyvin haastavan organisaation muutosprosessin. Olen tähän tutkimukseen koostanut aiemmissa tutkimuksissa ilmenneitä ongelmia ja riskejä, joita ERP-järjestelmien implementoinnissa on ilmennyt. Täten pyrin tuomaan esille seikkoja, jotka huomioon ottamalla sekä ennen implementointiprojektin alkua että sen aikana on mahdollista välttää ainakin osa ongelmista.

Avainsanat ja -sanonnat: ERP, implementointiongelmat.

CR-luokat: H.4.2

1. Johdanto

ERP-järjestelmä koostuu tietojärjestelmämoduuleista, joita yhdistämällä saadaan aikaan standardoitu ja integroitu kokonaisuus, jossa tietojärjestelmien tiedot liikkuvat sujuvasti eri tietokantojen ja järjestelmäosien välillä. Näin ollen ERP-järjestelmästä pystytään mm. koostamaan raportteja, joiden tieto voi olla peräisin sen eri tietojärjestelmämoduuleista, esim. varastonhallinnan, tuotannonohjauksen ja myynnin tietojärjestelmistä. Tällainen kombinoitu ja reaaliaikainen tieto antaa yritysjohtolle huomattavasti erillisjärjestelmiä paremman mahdollisuuden yrityksen toiminnanohjaukseen. Parhaimmillaan ERP-järjestelmien avulla saavutetaan nopeampi varastonkierto ja -tuotanto sekä niiden kontrolli. Tämän lisäksi sillä voidaan saavuttaa parempi tiedon laatu, liikkuvuus sekä näkyvyys koko organisaation tasolla [Holland *et al.*, 1999]. Joissakin tapauksissa toiminnanohjausjärjestelmällä haetaan jopa suoranaista kilpailuetua.

Toiminnanohjausjärjestelmän implementointiin liittyy kuitenkin monia riskejä ja ongelmia. ERP-järjestelmän implementointi maksaa paljon, monille yrityksille se voi olla jopa niiden historian kallein yksittäinen investointi [Sumner, 2000]. ERP-järjestelmät ovat laajasti konfiguroitavissa, mikä tekee niistä hyvin monimutkaisia. Tämä on kuitenkin pakollista, koska ERP-toimittajat haluavat tuotteidensa soveltuvan mahdollisimman monelle toimialalle ja eri yritykselle.

Implementoinnin vaikutukset eivät rajoitu pelkästään tekniselle tasolle, vaan sillä on myös suuri vaikutus organisaatiokulttuuriin ja koko siihen tapaan, jolla yritys toimii ja tämän muutoksen hallitseminen on sen todellinen haaste.

2. Ongelmat ja riskit

Usein implementoinnin epäonnistumisen syyksi mainitaan tekniset, johdolliset tai taloudelliset asiat. Syynä on kuitenkin useimmiten huono muutoksen hallinta nimenomaan inhimillisellä puolella [Kemppainen, 2004]. ERP-järjestelmän implementointi vaatii siihen ohjelmoitujen standardoitujen prosessien takia yritykseltä sen omien prosessien mukauttamista ERP-järjestelmän prosesseihin sopiviksi. Kyseessä on siis yrityksen muutosprosessi myös muussa kuin teknisessä mielessä.

Implementointiprojekti täytyy ensin oikeuttaa eli tuoda esille, kuinka ERP-järjestelmä hyödyttää yritystä. Tämä voidaan tehdä analysoimalla realistisesti sen tuomia etuja niihin resursseihin, joita sen implementointi vaatii [Sumner, 2000]. Monet yritykset implementoivat ERP-järjestelmän 90-luvun loppupuolella 2000-ongelman vuoksi. Niiden piti uusia tietojärjestelmänsä ja useimmille paras ratkaisu oli siirtyä käyttämään suoraan integroitua toiminnanohjausjärjestelmää, joka korvasi useat erilliset ja vanhat tietojärjestelmät.

Implementoidakseen ERP-järjestelmän yrityksen tulee olla riittävän vakava-rainen, jottei projektista synny sille kohtuutonta riskiä. Implementointi vie runsaasti aikaa ja vaikka se hyvin onnistuisikin, se maksaa sijoituksensa takaisin hitaasti [Vogt, 2002]. Vogt [2002] mainitsee myös, että usein yritykset potevat implementointiprojektin jälkeistä masennusta. Tämä johtuu siitä, että vie aikaa ennen kuin työntekijät oppivat käyttämään uutta järjestelmää tehokkaasti hyväkseen, joten aluksi usein näyttää siltä, ettei ERP:stä olisikaan täyttämään sille asetettuja odotuksia.

Organisaation ja ERP-järjestelmän huono yhteensopivuus, projektin huono johtaminen, heikko johdon tuki, organisaation vastustus, heikko tiedon laatu ja huono yrityskulttuurin tuntemus ovat mm. asioita, jotka aiheuttavat vakavan uhan implementointiprojektin onnistumiselle. Seuraavassa on käyty läpi yleisimpiä implementointiin liittyviä ongelmia.

2.1. ERP-järjestelmän ja organisaation prosessien vastaavuus

ERP-järjestelmät on luotu siten, että niihin on ohjelmoituina ns. parhaat prosessit eli sen tietojärjestelmät on luotu toimimaan tiettyjen liiketoimintaprosessien mukaisesti, jotka kokemuksen ja tutkimuksen kautta on havaittu hyviksi ja sopiviksi mahdollisimman erilaisille yrityksille. Yrityksillä, jotka ERP-järjestelmää ovat hankkimassa, on yleensä ollut käytössä yrityksen sisäisesti kehittämiä

tietojärjestelmiä. Ne on luotu yrityksen prosesseja vastaaviksi ja tästä johtuen yritykset ovat tottuneet siihen, että tietojärjestelmät vastaavat niiden prosesseja eikä toisinpäin.

Tämä on eräs suurimpia ongelman aiheuttajia ERP-järjestelmien implementoinnissa. Mikäli hyvää yhteensopivuutta ERP-järjestelmään ohjelmoitujen prosessien ja yrityksen prosessien välillä ei saada aikaan, implementointi varmasti epäonnistuu.

Davenport [2000] ilmaisee ongelmatekijäksi etenkin sen, että yritysjohto näkee liiketoiminnan ja sen prosessit täysin erillisenä asiana tietojärjestelmästä, vaikka ne todellisuudessa ovat hyvin kiinteästi toistensa kanssa tekemisissä. Usein yritysjohton harhaoletuksena onkin, että integroitu tietojärjestelmäkokonaisuus saadaan yksinkertaisesti aikaan implementoimalla ERP-järjestelmä, ilman että tehtäisiin muutoksia liiketoimintaprosesseihin [Davenport, 2000]. ERP-järjestelmä tulisikin nähdä paremman ja tehokkaamman liiketoiminnan mahdollistajana eikä liiketoiminnasta erillisenä itseisarvona. Holland ja muut [1999] tähdentävätkin, että ERP:n liiketoimintaprosessit ovat yleensä parempia kuin yrityksen vanhat prosessit, joten niiden omaksuminen on tältä osin järkevää.

Konsultit ja ERP-toimittajat ovat vahvasti sitä mieltä, että yrityksen tulisi suorittaa BPR (business process re-engineering) eli liiketoimintaprosessien uudelleenjärjestely tietojärjestelmäpaketin prosesseihin sopiviksi ja itse ERP-järjestelmän ohjelmakoodiin tulisi tehdä mahdollisimman vähän muutoksia [Shanks *et al.*, 2000]. ERP-järjestelmän muokkaamiseen liittyy aina suuri riski, sillä ensinnäkin se aiheuttaa yleensä merkittäviä lisäkustannuksia uuden ohjelmakoodin muodossa ja lisäksi kokonaisjärjestelmän integraatio voi kärsiä. Kiireessä tehty ohjelmakoodi voi myös sisältää virheitä, mikä voi aiheuttaa ongelmia koko toiminnanohjausjärjestelmän tasolla [Vogt, 2002]. ERP-järjestelmän ylläpito ja päivitys voivat myös olla hankalia suorittaa, mikäli ohjelmakoodi sisältää muutoksia [Hong and Kim, 2002].

2.2. Projektin johtaminen

Implementointiprojektin johtaminen vaatii paljon siihen osallistuvan projektitiimin jäseniltä. Projektitiimi tulisi koostaa tasaisesti sekä liiketoiminta- että IT-alan asiantuntijoista, jotta sekä teknologinen että liiketoiminnallinen muutos saataisiin mahdollisimman hyvin läpiviedyksi [Shanks *et al.*, 2000]. Projektitiimin jäsenien olisi hyvä koostua paitsi ulkoisista asiantuntijoista (konsultit) myös oman yrityksen työntekijöistä, koska myös yrityksen henkilöstöstä olisi hyvä löytyä ERP-järjestelmien osaajia implementointiprojektin päätyttyä. Ulkoisten ja sisäisten asiantuntijoiden yhteensovittaminen voi olla ongelmal-

lista eri näkökulmien vuoksi, eikä yrityksen sisäistä asiantuntijuutta tietojärjestelmäprojektiin välttämättä edes löydy [Sumner, 2000].

Projektitiimien koko ja määrä voi vaihdella kulloisenkin tarpeen mukaan. Mikäli tuleva ERP-järjestelmä on hyvin keskitetty, täytyy myös projektin johdon olla keskitetty [Sumner, 2000]. Joustava projektin hallinta ja riittävän koulutuksen järjestäminen projektitiimin jäsenille on tärkeää onnistumisen kannalta [Scott and Vessey, 2000]. Projektitiimin jäsenille olisi suotava työrauha ERP-implementoinnin parissa ja delegoida heidän normaalit työnsä muille.

Aikataulussa pysyminen projektissa on vaikeaa ja vain todella harvoin siinä onnistutaan. Projektin alussa onkin tärkeää aikatauluttaa eri projektin vaiheet mahdollisimman tarkasti, jottei yllätyksiä tule. Implementoinnin aikana esiin nousee monia konfigurointiin liittyviä ongelmia, jotka vaativat päätöksentekoa. Davenport [2000] sanookin, että päätöksenteon viivästyminen on eräs suurimmista projektin aikataulua viivyttävistä asioista ja ehdottaa jokaiselle eteen tulevalle päätökselle tiukkoja aikarajoja, jotta aikataulussa pysyminen olisi mahdollista. Ajan käyttö täytyy suunnitella jokaisen implementointivaiheen mukaan siten, että mitä vaativammasta osasta on kyse sitä enemmän aikaa sille on varattava [Kemppainen, 2004]. Ajan käytön huono suunnittelu johtaa ongelmiin implementoinnin aikarajaan pyrittäessä. Projektin aikarajan venyminen maksaa koko ajan yritykselle lisää siihen sitoutettujen resurssien muodossa.

Ongelmien ilmetessä ne olisi pyrittävä selvittämään mahdollisimman nopeasti, koska projektin alussa ilmenneet pienetkin ongelmat kerääntyvät implementoinnin edetessä ja voivat aiheuttaa myöhemmin vakavampia ongelmia [Liang and Xue, 2004]. Projektitiimin on tärkeää pysyä koko organisaatiota koskevassa datan integraation suunnitelmassa, eikä lipsua siitä [Sumner, 2000].

Implementoinnin aikana on oltava selvä tavoite, johon siinä pyritään. Projektin loputtua tuloksia voidaan verrata alussa asetettuihin tavoitteisiin ja katsoa saavutettiin ne. Ilman selkeää implementointistrategiaa ja selkeän tavoitteen asettamista implementoinnin onnistumista voi olla mahdotonta arvioida.

2.3. Johdon tuki

Yrityksen johdolla on projektin onnistumisen kannalta suuri vastuu, sillä he ensi sijassa valitsevat projektitiimin jäsenet. Davenport [2000], Sumner[2000], Kemppainen [2004] sekä Shanks ja muut [2000] ovat kaikki vahvasti vakuuttuneita siitä, että implementointi ei voi onnistua ilman johdon vahvaa tukea projektille.

Johdon tehtävänä on asettaa strategiset tavoitteet, jotka ERP-järjestelmän implementoinnilla halutaan saavuttaa ja tehtävä etukäteissuunnitelma niiden toteuttamiseksi [Davenport, 2000].

Organisaatiossa ilmenee aina vastustusta ERP-järjestelmän implementointia kohtaan ja mikäli edes johto ei ole selvästi projektin takana tai johdon kesken on ristiriitaisuutta, vastustus vain lisääntyy. Mikäli vastustusta ilmenee, on johdon puututtava siihen, koska muuten projekti viivästyy tarpeettomasti.

2.4. Organisaatiokulttuuri

ERP-järjestelmän implementointi aiheuttaa suuria muutoksia organisaatiokulttuurissa. Se aiheuttaa muutoksia liiketoimintatavoissa ja -prosesseissa ja näin ollen muokkaa koko organisaation tapaa toimia.

Monien työntekijöiden työnkuva muuttuu tai jopa häviää kokonaan uuden järjestelmän myötä ja tämä herättää usein vastustusta organisaatiossa. Kompleksisuutensa takia ERP-järjestelmät ovat varsinkin alussa vaikeita käyttää, joten haluttomuutta niiden hyväksikäyttämiseen ilmenee.

Etenkin liiketoimintayksiköiden johtajille, joilla aiemmin on ollut suhteellisen vapaa asema valitessaan tietojärjestelmiään ja liiketoimintaprosessejaan, vastustavat usein standardoidun yhtenäisjärjestelmän asentamista, koska tällöin ne menettävät ison osan autonomiastaan sekä datan ja prosessien hallinnasta [Kemppainen, 2004]. Vapauksien antaminen tuo kuitenkin huomattavia lisäkustannuksia sekä implementointiin että tietojärjestelmien ylläpitoon ja vaikuttaa negatiivisesti organisaation tiedon ja prosessien integraatioon [Davenport, 2000].

ERP-järjestelmä vaatii kurinalaisen organisaatiokulttuurin liittyen tietoon, prosesseihin ja järjestelmien käyttöön [Davenport, 2000], koska integroitu tietojärjestelmäympäristö kärsii laajasti yksittäisistä virheistä, esim. tiedon syöttämisestä väärin järjestelmään.

Organisaatiokulttuurin muutoksen johtaminen ja hallitseminen on tärkeä osa implementointia. Voimakas vastustus ja haluttomuus käyttää uutta ERP-järjestelmää viivyttää implementointiprojektin läpivientiä ja uhkaa lopulta koko järjestelmän mielekkyyttä, jos sitä ei haluta käyttää [Davenport, 2000].

Scott and Vessey [2000] painottavat avoimen ilmapiirin luomista projektiin, jossa myös loppukäyttäjien kanssa voidaan keskustella ongelmista ja yrittää ratkoa niitä. Integroidun, standardin yhtenäisjärjestelmän tarve olisi perusteltava hyvin sen tuleville käyttäjille organisaation vastustuksen lieventämiseksi [Davenport, 2000]. Loppukäyttäjien huomioon ottaminen koko implementointiprojektin ajan edesauttaa ERP-järjestelmän hyväksyntää ja sen hyväksikäyttöä implementoinnin päätyttyä.

Pelkkä teknisen järjestelmän muutos ei riitä onnistuneeseen implementointiin, vaan organisaation sosiaalinen muutos on myös kyettävä hallitsemaan. Yrityksen vakiintuneet prosessit ovat pitkän historian ja kokemuksen kautta syntyneitä, eikä niistä luopuminen ole helppo asia, vaan kuten jo edellä

mainittua se aiheuttaa varmasti vastustusta. Vastustukseen pystytään siis kuitenkin vaikuttamaan ja siihen pitää vaikuttaa projektin onnistumiseksi. Loppukäyttäjille on kyettävä tarjoamaan riittävästi koulutusta ja informaatiota uudesta ERP-järjestelmästä ja sen käytöstä sekä perustelemaan sen paremmuus aiempaan tietojärjestelmään verrattuna.

2.5. Inhimilliset resurssit

Toiminnanohjausjärjestelmän implementointi saattaa etenkin suurien yritysten kohdalla kestää useita vuosia, joten sen on sitoutettava projektiin paljon omia työntekijöitään ulkoisten konsulttien ym. ERP-asiantuntijoiden lisäksi. Yleensä nämä työntekijät ovat yrityksen parhaimmista ja tästä johtuen seuraakin ongelmia, koska he eivät ehdi hoitamaan normaaleja työtehtäviään ERP-projektin lisäksi. Vogt [2002] onkin sitä mieltä, ettei projektitiimin jäsenille kannata antaa muita työtehtäviä implementointiprojektin aikana eikä myöskään heti sen jälkeen, koska heidän tulisi saada rauhassa keskittyä projektiin täysipainoisesti. Jos he joutuvat tekemään myös tavalliset työnsä ERP-projektin lisäksi, se voi johtaa oikomiseen ja huolimattomuuteen implementoinnissa joka edelleen johtaa ongelmatilanteiden syntyyn.

Projektitiimiin kuuluneet yrityksen omat työntekijät joudutaan yleensä sitouttamaan ERP-järjestelmän pariin vielä implementoinnin päätyttyäkin, koska he ovat ainoita joilla on paljon tietoa järjestelmästä ja näin ollen he voivat joutua vielä pitkään antamaan konsultointiapua järjestelmää ja sen käyttöä koskevista ongelmista.

Yrityksen johdolla voi olla myös liian optimistiset oletukset siitä, miten ERP-järjestelmä osaa luoda ja käyttää tietoa hyväkseen [Kemppainen, 2004]. He unohtavat, että ihmiset ovat kuitenkin edelleen niitä, jotka sitä tietoa käyttävät hyväkseen. Käyttäjien koulutukseen on siis panostettava voimakkaasti, jotta ERP-järjestelmää osattaisiin ja haluttaisiin käyttää mahdollisimman paljon hyväksi [Shanks et al. , 2000].

Jo liiketoimintaprosessien uudelleen suunnittelussa tulisi reflektoida loppukäyttäjää mahdollisimman paljon, jotta ongelma-alueet saataisiin jo aikaisessa vaiheessa selville [Liang and Xue, 2004]. Käyttäjää pitää lähestyä mahdollisimman selkokielellisesti, kun halutaan saada heidän näkökulmansa järjestelmään selville. Liian tekninen lähestymistapa ei tuo esille ongelmakohtia ja voi näin ollen aiheuttaa ongelmia myöhemmissä implementoinnin vaiheissa tai viimeistään siinä vaiheessa kun ERP-järjestelmä on jo käytössä [Agarwal and Ghosh, 2000].

Ylläpitokustannuksien lisäksi nimenomaan henkilöstön koulutuskustannukset on useimmiten aliarvioitu implementoinnin budjettia suunniteltaessa

[Sumner, 2000]. Tämä kertoo inhimillisten resurssien väheksymisestä ERP-projektin onnistumisen kannalta.

2.6. Tiedon laatu

Eräs ERP-järjestelmän tavoitteista on tarjota laadukasta tietoa liikkeenjohdon päätöksen tueksi. Siihen koodatut prosessit käsittelevät vain niiden prosesseihin kuuluvia tietoja, joten aivan kuten liiketoimintaprosessit myös niiden sisältämä tieto täytyy standardisoida järjestelmään sopivaksi. ERP-järjestelmän tehokkaan toimivuuden kannalta kriittisen tärkeää, että sen sisältämä tieto on laadukasta.

Aiemmista tietojärjestelmistä saatavan tiedon on oltava laadukasta [Shanks *et al.*, 2000] tai muuten se täytyy määritellä uudelleen, jotta sitä voitaisiin käyttää ERP-järjestelmässä. Kemppainen [2004] kertoo case-tutkimuksessaan, että ERP-järjestelmään tulevaa tietoa ei etukäteen ollut huomattu määritellä lainkaan, vaan ongelma tuli eteen kesken implementoinnin, joka aiheutti merkittävää viivästymistä koko projektin läpiviennille. Laadukas tieto siis on tärkeää onnistumisen kannalta ja järjestelmään tuleva tieto pitää myös etukäteen valikoida, jottei sen takia tule viivästymisiä implementointiin.

Tiedon standardoimisessa on myös sen laadukkuuden lisäksi toisenlainen ongelma. Aiemmissa, erillisissä tietojärjestelmissä kullakin liiketoimintayksiköllä tai toimipaikalla on ollut heidän liiketoimintaprosesseilleen ominaista ja tärkeää tietoa. Voi olla vaikeaa määrittää mitä tietoa yhtenäiseen ERP-järjestelmään sisällytetään, koska jotkin tiedot ovat joillekin liiketoimintayksiköille tai toimipaikoille tärkeitä, mutta toisille ei [Kemppainen, 2004]. Tästä syntyy ristiriitaisuutta, koska jokainen yksikkö pitää omia tietojaan muita tärkeämpinä eikä haluaisi luopua niistä.

2.7. Tekniset ongelmat

ERP-järjestelmät muodostavat laajan tietoteknisen infrastruktuurin yritykseen. Ne koostuvat ohjelmistokomponenteista eli moduuleista, joita yhdistelemällä saadaan kullekin yritykselle soveltuva ERP-ratkaisu aikaiseksi. Kun sopivat moduulit on valittu, järjestelmä konfiguroidaan yrityksen tarpeisiin sopivaksi.

Koska ERP-järjestelmät on suunniteltu soveltumaan mahdollisimman monelle eri toimialalle sekä erilaisille yrityksille, ne ovat hyvin suurelta osin muokattavissa erilaisiin tarpeisiin. Standardoituneisuus on ongelma, koska ne pyritään sopeuttamaan erilaisiin tarpeisiin sopiviksi laajan konfiguroinnin avulla. Tämä taas johtaa suureen kompleksisuuteen eli ERP-järjestelmien implementointi ja käyttäminen on vaikeaa [Soh *et al.*, 2003]. Jo valmiiksi eri toimialoille selkeästi suunnitellut ERP-järjestelmät olisivat helpommin imple-

mentoitavissa, mutta toimialasuuntautuneisuutta on ainakin vielä vain vähän olemassa.

Tiukat implementointiaikataulut johtavat usein siihen, että järjestelmän teknisessä konfiguroinnissa oiotaan monilta osin, koska teknisiä ongelmia ei saada ratkaistuksi riittävän nopeasti [Kemppainen, 2004]. Tämä taas aiheuttaa ongelmia myöhemmissä implementoinnin vaiheissa tai valmiissa ERP-järjestelmässä.

2.8. Kulttuuriset erot

Eräs vähemmän ERP-tutkimuksissa esillä ollut ongelma liittyy kulttuurisiin eroihin eri maiden välillä. Tämä näkökulma on hyvinkin aiheellinen etenkin globaalien yritysten ERP-järjestelmähankkeissa, koska niiden toimipaikat voivat sijaita hyvin erilaisissa kulttuuriympäristöissä. Se missä kulttuurissa eletään, vaikuttaa vahvasti myös sen maan organisaatioiden johtamiskulttuuriin ja koko sen tapaan toimia.

Kuten edellä mainittua, toiminnanohjausjärjestelmän implementointi muokkaa myös organisaatiokulttuuria. Eri kulttuureissa on myös erilainen organisaatiokulttuuri ja siitä johtuen niissä myös suhtaudutaan eri tavalla siihen kohdistuviin muutoksiin. Shanks ja muut [2000] ovat tutkineet kulttuurisia eroja ERP-järjestelmien implementointiin liittyvissä ongelmissa, esimerkiksi Kiina ja Australia. He ovat huomanneet hyvinkin suurta vaihtelua siinä, kuinka ERP-projekteihin yrityksissä suhtaudutaan. Kiinassa, jossa johtamismalli on hyvin hierarkkinen, implementointi ei aiheuta kovin suurta vastustusta mikäli yrityksen johto on selvästi sen kannalla. Australiassa yritykset ovat vähemmän hierarkkisia kuin Kiinassa ja siellä vastustus on selvästi voimakkaampaa tältä osin. [Shanks *et al.* 2000]

Myös se, onko kulttuuri kollektiivinen vai individualistinen vaikuttaa suhtautumistapaan. Kollektiivisessa Kiinassa ajatellaan enemmän yrityksen parasta kun Australiassa, joten siellä standardiin järjestelmään mukautumista ei niin voimakkaasti mielletä itsenäisyyden menettämiseksi. [Shanks *et al.* , 2000]

Epätietoisuuden sietokyky ja maskuliininen tai feminiininen ongelmanratkaisutapa vaikuttavat myös vahvasti yrityksen suhtautumiseen ERP-järjestelmää kohtaan. Mikäli epätietoisuuden sietokyky on heikkoa eli riskejä ei siedetä, ei ERP-järjestelmää mielellään implementoida, koska se sisältää aina taloudellisen riskin. [Shanks *et al.* , 2000]

Suuret ERP-toimittajat kuten SAP, Baan, Oracle, PeopleSoft ja JD Edwards ovat kaikki länsimaisia yrityksiä, joten niiden ERP-paketit on suunniteltu lähinnä länsimaisille yrityksille sopiviksi. Niiden tarjoamille ERP-järjestelmille on

yhteistä se, että ne kaikki vaativat implementoivalta yritykseltä voimakasta liiketoimintaprosessien standardoimista. Esimerkiksi Kiinassa yritysten uniikkien liiketoimintaprosessien standardoimista ei siedetä niin hyvin kuin länsimaissa, joten siellä ERP-järjestelmien täytyy olla voimakkaammin kustomoitavissa yrityksen omien liiketoimintaprosessien mukaisiksi. [Shanks *et al.*, 2000]

ERP-järjestelmätoimittajien tulisikin tarkemmin perehtyä eri kulttuurien erityispiirteisiin murtautuakseen laajemmille ERP-markkinoille. Kiinassa markkinoita hallitsevat paikalliset tietojärjestelmätoimittajat, joiden ERP-järjestelmät soveltuvat länsimaisia ERP-järjestelmiä paremmin kiinalaiseen yrityskulttuuriin. Ne ovat valmiiksi suunniteltuja ottamaan huomioon kiinalaisten yritysten erityispiirteitä ja ne pystytään voimakkaasti muokkaamaan vastaamaan implementoivan yrityksen prosesseja. [Shanks *et al.*, 2000]

Globaalien yritysten kohdalla standardista koko organisaation kattavasta järjestelmästä voi olla pakko jonkin verran tinkiä, koska eri maiden toimipaikkojen tietojärjestelmät on usein lokalisoitava eli muokattava vastaamaan sen maan kulttuurin johtamispiirteitä vastaavaksi [Liang and Xue, 2004]. Sopivan tasapainon löytäminen globaalien yhtenäisjärjestelmän ja toisaalta maakohtaisten spesifien asetusten välillä on vaikeaa, koska integroidusta kokonaisuudesta ei kuitenkaan täysin voida tinkiä [Nandhakumar *et al.*, 2004].

3. Yhteenveto

ERP-järjestelmien implementointiin liittyy monia eri ongelmia ja monia eri näkökulmia niiden käsittelyyn. Tutkimusaineisto koostuu suurelta osin case-tutkimuksista, joten eri implementointiprojekteissa ilmenneitä ongelmia yhdistämällä sai melko kattavan kuvan siitä ongelmakentästä, jonka suuressa tietojärjestelmäprojektissa voi kohdata.

Tutkimuksessa kävi esille, että huono valmistautuminen projektiin johti lähes varmasti ongelmatilanteisiin. Huolellinen projektisuunnitelman tekeminen, jossa on eriteltynä eri implementointivaiheet on välttämätön onnistumiselle. Järjestelmään tuleva tieto täytyy olla valmiiksi määriteltynä jo ennen implementoinnin aloittamista, koska se aiheuttaa ylimääräistä viivettä, jos se joudutaan suorittamaan projektin aikana.

Liian kunnianhimoiset aikataulut implementoinnin läpiviemiseksi kostaavat usein heikosti toimivana ERP-järjestelmänä. Aikataulu voi tästä syystä venyä tarpeettoman pitkäksi, koska ongelmat jotka olisi ollut helppo korjata jo implementoinnin aikana, ovat vaikeasti korjattavia jo implementoidusta järjestelmästä. Tästä syystä vaiheittainen implementointi, jossa jää enemmän aikaa oppia virheistä ja huomata sekä korjata ongelmia, tuottaa yleensä myös laadukkaamman ERP-järjestelmän.

Vaikka ERP-järjestelmien implementoinnin ongelmista olisikin tietoinen etukäteen, ei niitä kaikkia voi implementointiprojektin aikana välttää. Tasapainoinen ja asiantunteva projektitiimi voi kuitenkin jo etukäteen varautua tiettyihin ongelmiin, sen sijaan että ne tulisivat yllätyksenä. Kesken implementoinnin yllättävät ongelmat joista ei osattu odottaa, viivästyttävät implementointia ja näin ollen luovat merkittäviä lisäkustannuksia yritykselle.

ERP-järjestelmän näkeminen uudenlaisen liiketoiminnan mahdollistajana eikä liiketoiminnasta erillisenä kokonaisuutena, on tärkeää projektin onnistumiselle. Vaikka tekninen implementointi onnistuisikin, niin mikäli ERP-järjestelmä ei tuota yrityksen liiketoiminnalle minkäänlaista lisäarvoa, sitä yleensä voidaan pitää epäonnistuneena investointina [Davenport, 2000].

Viiteluettelo

- [Agarwal and Ghosh, 2000] Rakesh Agarwal and Arup Ratan Raha Bhaskar Ghosh, Our experience and learning in ERP implementation. *ACM SIGSOFT Software Engineering Notes* **25** (2000), 31-34.
- [Davenport, 2000] Thomas H. Davenport, *Mission Critical: Realizing the Promise of Enterprise Systems*. Harvard Business School Press, 2000.
- [Holland et al. , 1999] C.P. Holland, B. Light and N. Gibson, A critical success factors model for enterprise resource planning implementation. In: *Proc. of the Seventh European Conference on Information Systems*, 273-287.
- [Hong and Kim, 2002] Kyung-Kwon Hong and Young-Gul Kim, The critical success factors for ERP implementation: an organizational fit perspective. *Information and Management* **40** (2002), 25-40.
- [Kemppainen, 2004] Ilkka Kemppainen, Change management perspectives in an ERP implementation. In: *Proc. of the Twelfth European Conference on Information Systems*, Turku School of Economics and Business Administration, Turku, Finland. Available: <http://www.tucs.fi/publications/attachment.php?fname=inpKemppainen04a.pdf>
- [Liang and Xue, 2004] Huigang Liang and Yajiong Xue, Coping with ERP-related contextual issues in SMEs: a vendor's perspective. *Journal of Strategic Information Systems* **13** (2004), 399-415.
- [Nandhakumar et al. 2004] J. Nandhakumar, J.M. Talvinen and M. Rossi, ERP revelations: the dynamics of contextual forces of ERP implementations. In: *Proc. of the Twelfth European Conference on Information Systems*, Turku School of Economics and Business Administration, Turku, Finland. Available: <http://is.lse.ac.uk/asp/aspecis/20040117.pdf>

- [Scott and Vessey, 2000] Judy E. Scott and Iris Vessey, Implementing enterprise resource planning systems: the role of learning from failure. *Information Systems Frontiers* **2**, 2 (2000) 213 - 232.
- [Shanks *et al.* , 2000] G. Shanks, A. Parr, B. Hu, B.J. Corbitt, T. Thanasankit and P.B. Seddon, Differences in critical success factors in ERP systems implementations in Australia and China: a cultural analysis. In: *Proc. of the Eighth European Conference on Information Systems*, 537-544.
- [Soh *et al.* , 2003] C. Soh, S.K. Sia, W.F. Boh and M. Tang, Misalignments in ERP implementation: a dialectic perspective. *International Journal of Human-Computer Interaction* **16** (2003), 81-100.
- [Sumner, 2000] M. Sumner, Risk factors in enterprise-wide/ERP projects. *Journal of Information Technology* **15** (2000), 317-327.
- [Vogt, 2002] Christian Vogt, Intractable ERP: a comprehensive analysis of failed enterprise-resource-planning projects. *ACM SIGSOFT Software Engineering Notes* **27** (2002).

Inhimillinen tekijä tietoturvallisuudessa

Jari Kivelä

Tiivistelmä.

Tässä tutkielmassa käsitellään tietoturvallisuutta painottuen inhimillisen tekijän vaikutuksiin sekä tietoturvallisuuden menetelmien menestymisessä että niiden epäonnistumisessa. Tietoturvallisuuden lähtökohtana on ollut jakaa turvallisuus erilaisiin ala-osioihin, kuten esimerkiksi tietoliikenneturvallisuus, fyysinen turvallisuus ja henkilöstöturvallisuus. Lisäksi osa-alueiden sisällä on pyritty jakamaan tietoturvallisuuden toimintatapaa akselilla tekninen – inhimillinen. Useimmiten inhimillinen tekijä jätetään kuitenkin vähemmälle huomiolle. Tämä saattaa olla osasyynä sille, miksi tietoturvallisuus ei ole kehittyneistä menetelmistään ja jaotteluistaan huolimatta aivan pystynyt lunastamaan sille asetettuja vaatimuksia inhimillisen tekijän vaikutusten arvioijana ja paremman tietoturvallisuuden takaajana. Teknologia ei siis ole antanut vastausta tietoturvallisuuden hallintaan. Tässä tutkimuksessa selvitetään olemassa olevia käsityksiä inhimillisestä tekijästä kartoittamalla miten tietoturvastandardit jaottelevat tietoturvallisuuden ja millaista problematiikkaa niiden eri osa-alueisiin sisältyy. Lisäksi käsitellään inhimillisen toiminnan monimuotoisuutta ja moraalikäsityksiä analyoimalla mm. hakkereiden motiiveja.

Avainsanat ja -sanonnat: Tietoturvallisuus, inhimillinen tekijä, hakkerit, moraalit, tietoturvastandardit.

CR-luokat: K.4.1, H.2.0, D.4.6

1. Johdanto

”Jos luulet että teknologia poistaa tietoturvaongelmat, et joko ymmärrä ongelmaa tai et ymmärrä teknologiaa”

Bruce Schneier

Tietoturvallisuus määritellään tavoitetilaksi, jossa tiedot, tietojärjestelmät ja palvelut saavat asianmukaista suojaa niin, että niiden luottamuksellisuuteen, eheyteen ja käytettävyyteen kohdistuvat uhat eivät aiheuta merkittävää vahinkoa yhteiskunnalle ja sen jäsenille [Vahti, 2003]. Määrittelyssä mainitut luottamuksellisuus, eheys ja käytettävyys muodostavat tietoturvallisuuden kulmakivet. Paavilainen [1998] määrittelee käsitteet seuraavasti:

Luottamuksellisuudella tarkoitetaan, että tiedot ovat vain niihin oikeutettujen henkilöiden ja organisaatioiden saatavilla, eikä niitä paljasteta muille.

Eheydellä tarkoitetaan, että tiedot eivät muutu tai tuhoudu laitteisto- tai järjestelmävian tai inhimillisen toiminnan tms. vuoksi.

Käytettävyydellä tarkoitetaan, että tiedot ja niiden muodostamat palvelut ovat niihin oikeutettujen henkilöiden käytettävissä tai saatavilla.

Tietoturvallisuuden toimenpiteet ja menetelmät keskittyvät näin ollen toteuttamaan yllä mainitut vaatimukset kaikissa mahdollisissa olosuhteissa. Luottamuksellisuudella on lakiin perustuva pohjansa ja luottamuksellisuutta vaalimaan on mm. määritelty Laki yrityssalaisuudesta [RL]. Eheys ja käytettävyys ovat myös tällä hetkellä käytännössä välttämättömiä, vaikka niiden laiminlyönnille ei olekaan määritelty suoraa rikosoikeudellista rangaistusta. Organisaatiolle näiden vaatimusten laiminlyönti merkitsee joka tapauksessa merkittäviä liiketoiminnallisia menetyksiä. Laiminlyönti voi aiheuttaa myös rikosoikeudellista vastuuta, eli esimerkiksi toimihenkilöiden ja johtajien vastuusta huolehtia siitä, että tiedot yrityksestä ovat luotettavia ja antavat oikean kuvan yrityksen taloudellisesta tilanteesta. Esimerkiksi tilinpäätöstietojen eheyden menettäminen, eli lukujen virheellisyys, voi aiheuttaa korvausvaatimuksia sijoittajilta.

2. Ohjeet ja standardit

Tietoturvallisuuden vaatimukset ovat mainittujen kolmen tekijän, luottamuksellisuuden, eheyden ja käytettävyyden, muodossa määritelty. Koska vaatimukset ovat kuitenkin sellaisenaan luonteeltaan hyvin yleispiirteisiä ja vaikeasti hahmotettavia, on niitä täsmennetty erilaisissa ohjeissa ja standardeissa.

Tietotekniikka ja informaatiotekniikka eivät ole luonteeltaan yhtä regulatiivisia kuin monet muut toimialat, kuten esimerkiksi terveydenhuolto ja pankkitoiminta. Tietotekniikka ei näin ollen ole ensisijaisesti itse lähtenyt määrittämään turvallisia toimintatapoja, vaan paljon menettelyistä on tullut sellaisten toimialojen asettamien vaatimusten kautta, joissa tietotekniikalla on tärkeä rooli toiminnan mahdollistajana ja tehostajana. Tietotekniikalle on olemassa vaatimuksia erityisesti tehokkuuteen ja tietoturvallisuuteen liittyen.

Tietoturvallisuuteen liittyviä ohjeita ja standardeja on useita, joista tässä on käsitelty kaksi Suomessa tällä hetkellä laajemmin käytössä olevaa, Vahti ja ISO/IEC 17799. Kumpikin standardi on esitelty luokitteluineen, eli miten kyseinen standardi jakaa tietoturvallisuuden eri osa-alueisiin. Lisäksi tarkastellaan erityisesti sitä, minkälaista problematiikkaa luokitteluun liittyy ja mitä ongelmia erilaisten menettelyiden täytäntöönpanossa esiintyy.

2.1. VAHTI

Vahti (Valtionhallinnon tietoturvallisuuden johtoryhmä) ohjeisto on Valtiovarainministeriön tuottama tietoturvallisuuden ohjekokoelma, joka on käytössä erityisesti julkishallinnossa ja erilaisilla viranomaisilla [Vahti]. Ohjeisto sisältää sekä yksityiskohtaisia ohjeita, kuten esimerkiksi ”Valtionhallinnon sähköpostien käsittelyohje” [Vahti, 2005] että laajempia katsantoja tietoturvallisuuteen valtionhallinnossa, kuten esimerkiksi ohjeessa ”Valtionhallinnon tietoturvallisuuden kehitysohjelma 2004–2006” [Vahti, 2004a]. Ohjeita lisätään ja päivitetään tarpeen mukaan, mutta ohjeilla ei ole varsinaista käyttökohteen mukaista hierarkiaa, joka helpottaisi tietyn tarpeen sovittamista ohjeeseen. Kaikille ohjeille yhteistä on kuitenkin käytössä oleva tietoturvallisuuden jaottelu eri osaluokkiin. Tässä käsiteltävä jaottelu käy hyvin esiin mm. ohjeesta ”Valtionhallinnon keskeisten tietojärjestelmien turvaaminen” [Vahti, 2004b].

Hallinnollinen turvallisuus pitää sisällään tietoturvallisuuteen tähtäävät hallinnolliset keinot, kuten organisaatiojärjestelyt, tehtävien ja vastuiden määrittely sekä henkilöstön ohjeistus, koulutus ja valvonta [Vahti, 2004b].

Hallinnolliseen turvallisuuteen liittyvä käsite johdon vastuusta on eräs organisaatioiden tietoturvallisuuden tärkeimmistä lausumista. Tällä tarkoitetaan, että viime kädessä on organisaation johdon tehtävä huolehtia, että tietoturvallisuuden vaatimukset tulevat täytetyiksi. Yleensä johto vastuuttaa tehtävät muille asiantuntijoille, mikä toisaalta myös hämärtää vastuun rajoja. Johdon esimerkki tietoturvallisuuteen suhtautumisesta antaa käsityksen koko organisaation tietoturvallisuuden tasosta.

Henkilöstöturvallisuudella tarkoitetaan henkilöstöön liittyvien tietoriskien hallintaa henkilöstön soveltuvuuden, toimenkuvien, sijaisuuksien, tiedonsaanti- ja käyttöoikeuksien, suojaamisen, turvallisuuskoulutuksen ja valvonnan osalta [Vahti, 2004b].

Henkilöön liittyvät tietoturvallisuuskysymykset alkavat jo ennen kuin henkilöstä tulee edes organisaation työntekijä. Taustatarkistukset ja -selvitykset antavat kuvaa henkilön luonteesta ja moraalista. Mitä enemmän henkilöstä tiedetään ennen palkkausta, sitä luottavaisempia voidaan olla sen suhteen, että henkilön moraalit on soveltuva myös palkkaavan organisaation tietoturvallisuusvaatimuksiin.

Fyysinen turvallisuus on henkilöiden, laitteiden, aineistojen, postilähetysten, toimitilojen ja varastojen suojaamista tuhoja ja vahinkoja vastaan. Fyysinen turvallisuus sisältää muun muassa kulun- ja tilojen valvonnan, vartiointin, palo-, vesi-, sähkö-, ilmastointi ja murtovahinkojen torjunnan sekä kuriirien ja tietoaineistoja sisältävien lähetysten turvallisuuden [Vahti, 2004b].

Fyysisen turvallisuuden katsotaan yleensä olevan ensimmäinen suojakeino vahinkoa vastaan. Sen tavoitteena on toisaalta suojata henkilöitä ja varoja fyysisiltä uhilta ja toisaalta suojata sekä varoja että ihmisiä ulkopuolisilta henkilöiltä. Osa suojamenetelmistä sopii sekä fyysisten varojen suojaamiseen että henkilöiden suojaamiseen, mutta mikäli nämä ovat ristiriidassa, niin henkilöiden suojaaminen menee aina etusijalle. Tätä periaatetta voi mahdollinen murtautuja hyväksikäyttää, koska hänen tavoitteenaan ei välttämättä ole suojella ihmisiä. Esimerkiksi kulunvalvontaan liittyviä suojamekanismeja on useita ja osan käyttöön liittyy enemmän ongelmia kuin mitä ne tuovat lisäturvaa. Esimerkiksi yhden henkilön kerrallaan päästävät lukittavat ovijärjestelyt (ns. man-trap) hidastavat työntekijöiden päivittäistä liikkumista, mutta ne eivät kuitenkaan tuo vastineena merkittävää lisäturvaa, koska murtautuja valitsee yleensä heikoimman lenkin, eli esimerkiksi avaimen varastamisen.

Tietoliikenneturvallisuuteen tähtääviä keinoja ovat mm. laitteistojen ja siirtoyhteyksien ylläpito ja niiden kokoonpanojen hallinta, verkonhallinta, pääsynvalvonta, tietoliikenteen käytön valvonta ja tarkkailu, ongelmatilanteiden kirjaaminen ja selvittäminen, viestinnän salaaminen ja varmistaminen sekä tietoliikenneohjelmien testaus ja hyväksyminen [Vahti, 2004b].

Tietoliikenneturvallisuutta tarkastellaan yleensä hyvin teknisenä tietoturvallisuuden osa-alueena. Osa tietoliikenneturvallisuudesta voidaan hoitaa teknisin menetelmin, mutta suuri osa tietoliikenneturvallisuudenkin suojaamiskyvykkyydestä nojaa menetelmiin, käytäntöihin ja niiden valvontaan, eli viimekädessä henkilöiden tunnollisuuteen. Esimerkiksi tietomassojen ja lokien ollessa massiivisia, ei voida olettaa, että lokeja seurattaisiin aktiivisesti. Niitä kuitenkin kerätään, joka toisaalta ohjaa käytökseen, että suojamekanismista ei tarvitse välittää.

Laitteistoturvallisuus liittyy tietojenkäsittely- ja tietoliikennelaitteiden ja tilojen käytettävyyteen, toimivuuteen, kokoonpanojen määrittelyyn ja pääsynvalvontaan sekä varaosien ja tarvikkeiden saatavuuteen [Vahti, 2004b].

Tietoliikenneturvallisuuden tavoin laitteistoturvallisuutta käsitellään teknisenä suojauksen osa-alueena. Esimerkiksi puhtaat laitteiden ja tilojen toimivuuteen liittyvät tekijät on helppo hoitaa fyysisin suojauskeinoin, mutta laitteistoturvallisuuteen liittyvät myös laitteiden omistajuus ja säännöllinen huolto, joiden oikea toteuttaminen on hyvin henkilöriippuvaista. Huolto voidaan esimerkiksi ulkoistaa, jolloin riskiä siirretään ulkopuoliselle toimijalle, jonka motiivina on raha tai sen menettäminen uhkasakon muodossa.

Ohjelmistoturvallisuudella tarkoitetaan käyttöjärjestelmiin ja muihin ohjelmistoihin kohdistuvia toimia, kuten ohjelmistojen tunnistamis-, eristämis-,

pääsynvalvonta- ja varmistusmenettelyitä, tarkkailu- ja paljastustoimia, lokimennettelyitä ja laadunvarmistusta sekä ohjelmistojen ylläpitoon ja päivitykseen liittyviä toimia tietoturvallisuuden parantamiseksi [Vahti, 2004b].

Ohjelmistot ovat inhimillisen työn tulosta ja näin ollen alttiita virheille. Ohjelmistoturvallisuus pitää sisällään useita menettelyitä, joiden avulla ohjelmistojen laatua ja samalla myös tietoturvallisuutta voidaan parantaa. Osa toimenpiteistä on luonteeltaan ennaltaehkäiseviä ja osa jälkikäteen havaitsevia. Menetelmien käyttö vaatii kuitenkin koulutusta ja jatkuvaa panostusta, mikä ei ole ohjelmien nopeasta julkaisutahdista johtuen useinkaan mahdollista. Vielä ei olla tilanteessa, jossa ohjelmistoyritykselle olisi erityisen hyödyllistä ylläpitää korkeaa laatua, koska kilpailijoiden tilanne on useimmiten sama. Laatuun ei siis kannata sijoittaa, koska muutkaan eivät sitä tee.

Tietoaineistoturvallisuuteen katsotaan kuuluvan tietoturvallisuuteen tähtäävät toimet asiakirjojen, tiedostojen ja muiden tietoaineistojen käytettävyyden, eheyden ja luottamuksellisuuden ylläpitämiseksi keinoina muun muassa tietoaineistojen luettelointi ja luokitus sekä tietovälineiden ohjeistettu hallinta, käsittely, säilytys ja hävittäminen [Vahti, 2004b].

Tietoaineistoturvallisuuteen liittyviä teknisiä suoja mekanismeja ovat mm. fyysinen suojaaminen ja salaus. Suurin osa muista toimenpiteistä, mm. aineiston omistajuus ja turvallisuusluokittelu, nojaa niistä vastuussa olevan henkilön osaamiseen ja huolellisuuteen. Eräs vaihtoehto on esimerkiksi luokitella kaikki organisaation asiakirjat luottamuksellisiksi, mutta tämän tuloksena voi olla, että luokittelu menettää merkityksensä, koska kaikkia asiakirjoja ei kuitenkaan osata tai viitsitä suojata.

Käyttöturvallisuus liittyy tietotekniikan käyttöön, käyttöympäristöön, tietojenkäsittelyyn ja sen jatkuvuuteen sekä tuki-, ylläpito-, kehittämis- ja huolto-toimintoihin liittyviin keinoihin tietoturvallisuuden parantamiseksi [Vahti, 2004b].

Käyttöturvallisuuteen liittyy nykyisin oleellisena osana ylläpitosopimukset sekä kaiken kattavat ulkoistussopimukset. Tällaisten sopimusten solmiminen on ollut välttämätöntä kustannustehokkuusvaatimusten kasvaessa. Ulkoistaminen ei kuitenkaan ulkoista vastuuta, vaan se säilyy edelleen ulkoistuksen tehneellä organisaatiolla. Vastuukysymykset saattavat kuitenkin ongelmatilanteissa hämärtyä. Lisäksi tulee huomioda, että vaikka jokin toiminnallinen kokonaisuus olisikin ulkoistettu toiselle organisaatiolle, on tällä ulkoistuspalvelua harjoittavalla organisaatiolla ratkaistavana samat kysymykset, kuin mitkä ulkoistava organisaatio on halunnut ratkaista, eli kysymykset liittyen mm. henkilöiden osaamiseen ja moraaliin.

2.2. ISO/IEC 17799

ISO/IEC 17799 on ISO:n ja IEC:n yhteistyössä laatima tietoturvallisuusstandardi [ISO/IEC1], joka on kehittynyt alkujaan Britannian standardointiinstituutin standardista BS7799. ISO/IEC 17799 standardia on tällä hetkellä saatavana kahtena eri osiona, joista ensimmäinen osio, eli ISO/IEC 17799:2005 antaa suositukset hyvän tietoturvallisuuden saavuttamiseksi ja toinen osa ISO/IEC 27001:2005 [ISO/IEC2] asettaa sertifioitavissa olevat vaatimukset tietoturvalle järjestelmälle. ISO/IEC:n rakenne ja tietoturvallisuuden painopisteet ovat vuosien aikana muuttuneet, mikä osoittaa, että tietotekniikkaan ja erityisesti tietoturvaluuteen liittyvän standardin täytyy mukautua myös kasvaviin vaatimuksiin. Alla ISO/IEC:n tietoturvallisuuden jaottelu ja mitä ongelmia kyseiseen osa-alueeseen liittyy (suomennokset kirjoittajan).

Tietoturvapoliittikka (security policy) muodostaa pohjan tietoturvallisuuden hallinnalle ja on samalla johdon osoitus tietoturvaluuteen sitoutumisesta.

Tietoturvapoliittikan jalkauttaminen koko organisaatioon on edelleen yksi tietoturvallisuuden haastavimmista tehtävistä. Poliittikka voidaan lausumana kirjoittaa ja jaella koko henkilöstölle, mutta tahtotilan sisäistäminen kaikessa laajuudessaan on erittäin haastava tehtävä koko organisaatiolle. Tietoturvapoliittikka voi esimerkiksi sisältää termistöä, joka on vierasta tavalliselle työntekijälle. Ei voida vaatia, että käyttäjä osaisi toimia oikein ohjeen kaikessa laajuudessa, ellei hän osaa yhdistää poliittikassa olevia käsitteitä, esimerkiksi luottamuksellisuudesta, jokapäiväiseen työhönsä. Poliittikkaa ja käsitteitä voidaan opettaa, mutta mikäli kyseisiin asioihin ei törmää päivittäin, unohtuvat ne pian normaalityöskentelystä.

Tietoturvallisuuden organisointi (organizing information security) ohjeistaa tietoturvallisuuden hallintajärjestelmän organisoinnissa, jonka avulla mm. vastuutetaan tietoturvaluuteen liittyvät työtehtävät.

Tietoturvallisuus voidaan vastuuttaa, mutta toisinaan unohdetaan resursointi, eli jokapäiväinen tietoturvallisuuden hoito on vastuutettu, mutta aikaa tai henkilöresursseja menettelyiden käyttöönottoon tai valvontaan ei olekaan. Asiantuntijoilla on yleensä tavoitteena parempi tietoturva, mutta esimerkiksi resurssien vähyyden vuoksi voidaan suoja mekanismeja ottaa käyttöön vain osittain tai tilapäisasetuksilla. Erityisesti teknisten suoja mekanismeiden monimutkaisuus hidastaa entisestään resurssivajaita käyttöönottoja.

Varojen hallinta (asset management) määrittää tiedoille ja järjestelmille omistajat ja turvallisuusluokituksen. Omistajuuden käsite on ollut tietoturvaluudessa pitkään. Tämä on hyvä periaate vastuuttamisen kannalta, mutta aina järjestelmän tai tiedon omistajalla ei ole riittävä osaamista arvioida kohteensa tietoturvaluuden tasoa tai siihen kohdistuvia riskejä. IT- tai tietoturvaluuteen

osaamista voidaan tässä kohtaa hyödyntää, mutta samalla hämärtyy omistajuus, eli käytännön päätöksiä saattaakin tehdä joku muu kuin varsinainen omistaja.

Henkilöstöturvallisuus (human resources security) huolehditaan mm. taustatarkistuksista, salassapidosta ja riittävästä koulutuksesta. Ennen työhön-ottoa tapahtuva taustatarkistus on oleellinen osa palkkausprosessia. Tällä hetkellä taustatarkistuksen ja virallisemmän turvallisuusselvityksen tekeminen ei ole varsinkaan yksityisille organisaatioille helppoa. Julkisten organisaatioiden palkkauspolitiikka mahdollistaa hieman laajemmat turvallisuusselvitykset, mutta yleensä selvitykset on kohdistettava sellaisiin henkilöihin, joilla on esimerkiksi työtehtäviensä kautta pääsy yhteiskunnalle kriittisiin tietoihin tai järjestelmiin. Yrityksen kannalta on yhtä tärkeää myös se, miten henkilö käsittelee yritykselle kriittisiä tietoja ja järjestelmiä.

Fyysinen- ja ympäristöturvallisuus (physical and environmental security) varmistuu järjestelmien ja tietojen fyysisestä suojauksesta, sekä inhimillisiä että luonnollisia uhkia vastaan.

Osa fyysisistä suojamekanismeista on ollut olemassa jo niin pitkään, että niitä on totuttu käyttämään, eikä niitä näin ollen yritetä esimerkiksi mukavuussyistä kiertää. Suojatuissa tiloissa liikkumiseen liittyy kuitenkin aina riskejä, koska niihin voidaan päästää tarpeettomia henkilöitä pelkästään sillä perusteella, että yleensä on tapana auttaa muita ihmisiä, eli päästää samasta ovesta ilman, että tarkistetaan kulkulupaa. Ihmiselle luonteenomaisen käyttäytymisen ollessa ristiriidassa suojamekanismien vaatimusten kanssa voidaan olettaa, että tietoturvarikkomuksia syntyy.

Käytönhallinta (communications and operations management). Pitää sisälleen kaikki käytönhallintaan liittyvät suojamekanismit mm. varmistukset, median käsittelyn ja sähköisen kaupan suojamekanismit.

Käytönhallintaan liittyy mm. tehtävien eriyttäminen, eli että samalla henkilöllä ei ole kahta sellaista työtehtävää, joiden hoidossa voisi ilmetä eturistiriitoja. Tällainen voisi olla esimerkiksi lupa tehdä muutos järjestelmässä oleviin palkkatietoihin sekä lupa hyväksyä palkkamuutospyynnöt saman järjestelmän avulla. Mikäli menettelyitä ei pystytä teknisesti pakottamaan, niin on mahdollista, että resurssisyistä henkilöllä on useampia tehtäviä, joita lisäksi ei päästä tehokkaasti valvomaan. Esimerkiksi järjestelmien ylläpitäjillä saattaa olla oikeus tehdä muutoksia järjestelmään sekä lisäksi poistaa lokitietoja, joista muutokset kävisivät ilmi. Tehtävien eriyttäminen tuo haasteita, koska esimerkiksi tiukassa taloudellisessa tilanteessa organisaatio pyrkii säästämään myös henkilömenoissa, ilman että vastaavia organisaation toimintoja kuitenkaan kokonaan

lakkautettaisiin. Tästä seuraa luonnollisesti, että pois lähtevien työntekijöiden tehtäviä jaetaan muille, jolloin vaarallisten tehtävähdistelmien riski kasvaa.

Pääsvalvonta (access control) valvoo pääsyä tietoon, järjestelmiin ja liiketoimintaprosesseihin. Se sisältää mm. tietoliikenteen turvallisuuden.

Järjestelmiin tulisi olla selkeästi määrätyt oikeudet kullekin käyttäjälle. Mikäli käyttöönottoprosessi on hyvin toteutettu, näin tapahtuukin, viimeistään siinä vaiheessa, kun järjestelmä siirtyy pilotointivaiheesta tuotantoon ja käyttäjämäärä lisääntyy. Joskus kuitenkin järjestelmään jää pilotti- ja testivaiheen käyttäjiä, joiden oikeudet saattavat olla tarpeettoman suuria suhteessa työtehtäviin. Toinen mahdollisuus on, että kun huomataan että järjestelmä ei toimikaan aivan halutulla tavalla käyttöönoton jälkeen, lisätäänkin käyttäjän oikeuksia, jotta ongelmia ei esiintyisi tai että ne voitaisiin kiertää. Vaikka järjestelmä myöhemmin saataisiinkin toimimaan halutulla tavalla, ei liian suuria oikeuksia muistetakaan enää poistaa. Lisäksi käyttöoikeuksia tulkitaan organisaatiossa 'saavutetuksi eduksi' eli niiden vähentäminen antaa vaikutelman siitä kuin henkilöön ei luotettaisi tai että hänen tärkeytensä tai asemansa olisikin aiempaa matalampi.

Tietojärjestelmien hankinta, kehittäminen ja ylläpito (information systems acquisition, development and maintenance) sisältää tietoturvallisuuden ylläpitämisen järjestelmän koko elinkaaren aikana.

Tietojärjestelmien kehittämisessä, vastaavasti kuten Vahti-ohjeistuksen ohjelmistoturvallisuudessakin, inhimillinen tekijä on merkittävä heti kehittämisprojektin alusta lähtien. Tietojärjestelmäsuunnittelun tulee perustua realistisiin arvioihin kehitettävästä järjestelmästä. Liian hätiköidysti toteutettu suunnittelu saattaa aiheuttaa mm. sen että esimerkiksi vaatimusten yllättäen muuttuessa toteutusvaiheessa, ei projektille pystytäkään enää allokoimaan riittävästi resursseja. Tämä voi johtaa esimerkiksi siihen, että alun perin suunnitellut tietoturvallisuutta vahvistavat kooditarkistukset ja virheenkäsittelyt jätetään vähälle huomiolle. Lisäksi kiire saattaa tulla testausvaiheessa, josta seurauksena on, että asiakkaalle toimitettava järjestelmä on keskeneräinen vaarantaen samalla luottamuksellisuuden, eheyden ja käytettävyyden.

Tietoturvallisuusloukkauksien hallinta (information security incident management) huolehtii, että tietoturvallisuusloukkaukset raportoidaan ja korjaavat toimenpiteet toteutetaan.

Tietoturvaloukkauksista raportointia ei koeta kovin mielekkäänä. Raportointia halutaan välttää erityisesti, jos sen esilletuonti vaikuttaa omaan asemaan tai jos muille osapuolille tulee vaikutelma omasta epäpätevyydestä asioiden hoidossa. Ja vaikka tietoturvaloukkaukset olisi sanktioitu esimerkiksi tietoturvapoliitikassa, ei sanktioiden täytäntöön paneminen ole ongelmatonta joh-

tuen esimerkiksi henkilösuhteista ja siitä että sanktioinnin koetaan rikkovan työyhteisön yhtenäisyyttä ja lisäävän valvoja-valvottava -asetelmaa.

Jatkuvuudenhallinta (business continuity management) huolehtii liiketoiminnan jatkuvuudesta myös poikkeusolojen aikana.

Riskienhallinnan tehtävänä on arvioida organisaatiolle aiheutuvia uhkia ja suositella toimenpiteitä riskien pienentämiseksi. Jatkuvuudenhallinta liittyy yleensä suurista uhkista suojautumiseen, oli sitten kyseessä tulva, tulipalo tai poikkeustila. Jos uhkat ovat suuria, tulee suojamekanismien olla vastaavia. Tästä aiheutuu väistämättä kustannuksia. Jo pelkästään tietojärjestelmien kahdentaminen esimerkiksi toiseen toimitilaan on niin merkittävä kustannus suurimmalle osalle organisaatioita, että sitä ei voida todellisuudessa toteuttaa. Useimmiten jäljellä jää vain vaihtoehto riskien siirtämisestä esimerkiksi vakuutusten kautta. Vaikka vakuutus korvaisi menetetyt järjestelmät, kärsii organisaatio kuitenkin yleensä toiminnan katkoksista ja normaaliolosuhteisiin palaminen saattaa kestää hyvinkin pitkään. Jos katkos on liian pitkä, on mahdollista että organisaatio ei pysty siitä toipumaan, vaan edessä on toiminnan supistaminen tai lopettaminen.

Lainmukaisuus (compliance) huolehditaan siitä että liiketoiminta on lakeja, säädöksiä ja sopimuksia noudattava. Lakien ja säädösten mukainen toiminta on yleensä organisaatioille ensiarvoisen tärkeää. Suurimmalla osalla organisaatioiden toimintaa säätelevistä laeista on pitkät perinteet ja niitä on totuttu noudattamaan ja myös valvomaan. Uudet lait, kuten esimerkiksi Laki yksityisytyden suojasta työelämässä [Laki 759/2004] ja Sähköisen viestinnän tietosuojalaki [SVTSL 516/2004] vaativat kuitenkin sekä lainopillista että teknistä osaamista ja näiden yhteensovittaminen ei ole ongelmatonta. Ongelmaa ei vähennä se, että lain laatijatkaan eivät välttämättä osaa arvioida kaikkia lain mukanaan tuomia teknisiä ja sosiaalisia vaikutuksia.

3. Ihminen toimijana

Standardit eivät yleensä käsittele omissa tietoturvallisuuden jaotteluissaan ihmistä millään tavalla erillisenä toimijana, vaan enemmänkin hallittavana osana jotakin muuta järjestelmää. Perinteinen uhka → suojausskenaario olettaa, että tiettyyn ulkopuoliseen uhkaan on aina olemassa tai ainakin mahdollista toteuttaa tietty suojakeino, joka estää tai havaitsee uhan. Tämä on sellaisenaan siirretty koskemaan myös inhimillistä toimintaa. Tästä kertovat monet erilaiset suojausmenetelmät, jotka olettavat, että ihminen toimii haitallisella tavalla X, jonka estämiseksi tai havaitsemiseksi löytyy suojakeino Y. Kaikkia suojausmenetelmiä ei tietenkään suoraan kohdisteta koskemaan ihmisen toimintaa, koska tämä koetaan liiallisena valvontana. Osa suojamekanismeista saattaakin

pääsääntöisesti tarkistaa järjestelmän toimintaa, mutta sen oheistuotteena pystytään varmistumaan myös käyttäjän toiminnan oikeellisuudesta. Esimerkiksi keskitetysti hallittu virustorjuntaohjelma voi tarkistaa omien ajettavien ohjelmien toimintaa virusten varalta, mutta samalla voidaan varmistua siitä, että käyttäjä ei ole sammuttanut virustorjuntaohjelmaa kiertääkseen kyseisen suojamenetelmän.

Ihmisen käyttöön tietojärjestelmän käyttäjänä ja organisaation osana liittyy aina jonkin verran satunnaisuutta, aiheutuu ihmisen toimintaan reagoivasta suojausmenettelystä se, että suojamekanismien määrä kasvaa koko ajan. Tästä taas seuraa eräänlainen varustelukierre, eli aina uuden käyttömallin tai virheellisen toimintatavan ilmentyessä toteutetaan sitä estämään uusi suojamekanismi. Koska inhimillisen käyttäytymisen kirjo on loputon, ei tälle kiertellekään ole nähtävissä loppua.

Suurin osa tietojärjestelmien käyttäjistä on tavallisia ihmisiä, joilla ei ole suurempia motiiveja kuin saada työtehtävät menestyksellisesti tehtyä. Heidän motiivinsa eivät ole siis lähtökohtaisesti haitallisia tai organisaation kannalta kyseenalaisia. Mikäli tietoturvallisuuteen liittyviä tapahtumia kuitenkin sattuu, voidaan ne yleensä jakaa kahteen eri luokkaan: tahallisiin ja tahattomiin haitallisiin tekoihin.

3.1. Tahalliset ja tahattomat haitalliset teot

Tahattomasta haitallisesta teosta voidaan puhua, kun esimerkiksi käyttäjä syöttää tietojärjestelmään jonkin parametrin vahingossa väärin ja tästä aiheutuu tietokannan eheyden menetys. Käyttäjällä on siis tehnyt inhimillisen virheen, jolla on organisaation laajuinen vaikutus. Tahallisesta haitallisesta teosta voidaan puhua, kun käyttäjä tekee tietokantaohjelmaan toiminnon tai kyselyn, jonka avulla hän esimerkiksi pääsee näkemään muiden työntekijöiden henkilötietoja.

Tahallisen ja tahattoman haitallisen teon erottavana tekijänä on siis henkilön motiivi. Teko ja sen lopputulos voivat jopa olla täysin samat, esimerkiksi virheellinen parametri ja siitä seuraava tietokannan korruptoituminen. Tahattomassa teossa ei ole motiivia, vaan kyseessä on puhdas vahinko, johtuen esimerkiksi kiireestä tai osaamattomuudesta. Tahallisessa teossa motiivina voi olla esimerkiksi kosto tai luottamuksellisen tiedon tavoittelu. Tällaisen tapahtuman sanktioiminen on kuitenkin ongelmallista, koska voi olla vaikea osoittaa, onko itse asiassa kyse tahallisesta vai tahattomasta teosta. Rikosoikeudellisessa mielessä osoitusvastuu on organisaatiolla, jonka tulisi pystyä osoittamaan se, onko ylipäätään tapahtunut jotakin haitallista, miten se on tapahtunut ja mahdollisesti etsiä tietoja myös motiivista. Todistusaineistona voidaan käyttää esimerkiksi järjestelmistä kerättyä lokiaineistoa, jonka pääasiallinen funktio

saattaakin olla työntekijän toimien oikeellisuuden valvonta, eikä niinkään järjestelmän toiminnan seuranta. Todistusaineisto voi paljastaa mm. aiempia tietoturvarikkomuksia, jolloin voidaan arvioida tapahtuneen teon motiivia uudestaan. Mikäli on näyttöä useita rikkomuksista, voidaan olettaa, että teot ovat olleet tarkoituksellisesti haitallisia ja henkilö on luokiteltavissa tietomurtojen tekijäksi eli hakkeriksi.

3.2. Hakkerit

Hakkeri on perinteisen käsityksen mukaan tietokoneista kiinnostunut harrastaja ja asiantuntija, joka toimii lain puitteissa. Krakkerilla on tarkoitettu omaa etua tavoittelevaa edistynyt käyttäjä, joka murtautuu järjestelmiin. Näiden termien käyttö on kuitenkin sekoittunut ja tällä hetkellä hakkeri-termiä käytetään yleisnimityksenä kaikille asiantuntijoille ja alan harrastajille, joiden toiminnan päätavoitteena on tietojärjestelmille haitallisten toimintojen suorittaminen, joka organisaation sisäpuolelta tai ulkopuolelta. Hakkeri voi siis toisaalta olla virusten kirjoittaja ja toisaalta taitava tietojärjestelmiin murtautuja.

Hakkereihin liittyvä asiantuntijuuden ja "tietokoneneron" leima on kuitenkin vuosien varrella karissut, kuten Skibell [2002] esittää. Suurin osa hakkeriksi itseään kutsuvista on nuoria tietokoneharrastajia, jota käyttävät apunaan Internetistä löytämiään ohjelmia, joiden avulla sekä virusten kirjoittaminen että tietojärjestelmiin hyökkääminen voidaan automatisoida. Osaamista ei näin ollen vaadita enää läheskään siinä määrin kuin aiemmin tietokoneiden ollessa harvinaisempia.

Käsitys moraalista on muotoutunut uudestaan hakkerien keskuudessa. Hakkerin motiivina voi toimia "ammatillinen" intohimo, taloudellinen hyöty tai esimerkiksi verkkoaktivismi. Verkon anonymiteetti tuo suojaa mahdollisia oikeustoimia vastaan, ja toisaalta taas erilaiset hakkereiden verkkoyhteisöt tuovat tavoiteltavaa kunnioitusta. Verkossa oma tavoitettavuus ja näkyvyys voidaan pitkälti itse valita, mikä lisää vallan tunnetta, joka vuorostaan lisää mm. itsetuntoa. Hakkerin toimintaa kuvastaa voimakas haasteellisuus, jonka tavoittelu ylittää moraalin [Kivelä, 2002]. Tietojärjestelmät ja tietoverkot ovat heille maailmaa, jossa ei ole lakeja, koska ne ovat liian helposti rikottavissa. Näin ollen tietojärjestelmiin murtautumisessa oletuksena voi olla haasteellisuuden tavoittelu, joten sopivan suojamekanismin löytäminen on mahdotonta, koska tavoitteena on juuri löytää aukkoja kaikista mahdollisista suojamekanismeista. Suojamekanismi itsessään on haaste, eikä täytä näin ollen sille asetettua suojaavaa tarkoitusta. Lisäksi murtautujilla on yleensä puolellaan merkittävästi resursseja, eli paljon aikaa ja tukenaan muita vastaavasta toiminnasta kiinnostuneita alan harrastajia.

4. Yhteenveto

Inhimillinen toiminta aiheuttaa merkittävää haastetta jokapäiväisessä tietoturvatyössä. Toiminta varioi vahingossa poistetuista tiedostoista aina hakkereiden toteuttamiin palvelunestohyökkäyksiin ja sekunneissa leviäviin maailmanlaajuisiin virusepidemioihin. Tällä hetkellä tietoturvastandardit ottavat kyllä huomioon inhimillisen tekijän olemassaolon, mutta menetelmät ja suojamekanismit tämän tekijän kontrolloimiseksi ovat haastavia. Inhimillinen tekijä voidaan ottaa myös liian kontrolloimattomana ja epävarmana, kuten Gonzalez ja Sawicka [2002] esittävät.

Inhimillisen toiminnan kontrolloimiseksi ei voida pelkästään tehdä suojamekanismeja ja olettaa, että ne sellaisenaan toimivat. Parhaan suojauksen inhimillistä toimintaa vastaan tarjoaisi luonnollisesti toinen ihminen, eli ihminen on tietoturvallisuudessa samanaikaisesti sekä heikoin että vahvin lenkki [Egan, 2005]. Nykyisessä tietoverkkoyhteiskunnassa ja nykyisillä tietomäärillä inhimillisen suojamekanismin, eli ihmisen käyttö tilanteen arvioijana on aina vain vaikeampaa. Tietoturvastandardit ja niiden tarjoamat tekniset tietoturvamekanismit tuovat osan ratkaisusta, mutta todellinen tietoturvallisuus saavutetaan vasta sitten kun suojamekanismit eivät ole reaktioita tietoturvarikkomuksiin vaan ovat mukana järjestelmäkehityksen alusta aina loppukäyttäjiin ja heidän koulutukseensa saakka.

Viiteluettelo

- [Egan, 2005] Mark Egan, Information security and the human factor, *Information System and Control Association JournalOnline*, 2005.
- [Gonzalez and Sawicka, 2002] Jose J Gonzalez and Agata Sawicka. A Framework for human factors in information security. Presented in WSEAS Int. Conf. on Information Security, 2002.
- [Kivelä, 2002] Jari Kivelä, Krakkerietiikka, Edutech Business Forum seminaari Tietoturva ja Etiikka, Tampereen teknillinen korkeakoulu, 2002.
- [Laki 759/2004] Laki yksityisyyden suojasta työelämässä 13.8.2004/759.
- [Paavilainen, 1998] Juhani Paavilainen, *Tietoturva*, Suomen Atk-kustannus, 1998.
- [RL] Rikoslaki 30:11§.
- [Schneier, 2000] Bruce Schneier, *Secrets and Lies: Digital Security in a Networked World*, John Wiley & Sons Inc., 2000.
- [Skibell, 2002] Reid Skibell, The myth of the computer hacker, *Information, Communication & Society* 5, 3 (2002), 336-356.
- [SVTSL 516/2004] Sähköisen viestinnän tietosuojalaki 16.6.2004/516.

- [Vahti] Valtiovarainministeriö, www.vm.fi/vahti, 20.12.2005.
- [Vahti, 2003] Valtionhallinnon tietoturvakäsitteistö, Edita Prima Oy, 2003.
- [Vahti, 2004a] Valtionhallinnon tietoturvallisuuden kehitysohjelma 2004–2006, Edita Prima Oy, 2004.
- [Vahti, 2004b] Valtionhallinnon keskeisten tietojärjestelmien turvaaminen, Edita Prima Oy, 2004.
- [Vahti, 2005] Valtionhallinnon sähköpostien käsittelyohje, Edita Prima Oy, 2005.

Käytettävyyden rooli verkko-opetuksessa

Suvi Leander

Tiivistelmä

Tässä tutkielmassa perehdytään käytettävyyden merkitykseen verkko-opetuksen suunnittelussa. Verkko-opetusohjelman hyvä käytettävyys mahdollistaa oppilaille hyvän käyttö- ja oppimiskokemuksen ja auttaa häntä oppimistavoitteiden saavuttamisessa. Tässä tutkielmassa esitellään verkko-opetuksen ja käytettävyyden peruskäsitteitä. Käytettävyys tulee ottaa huomioon jo verkko-opetusohjelman suunnittelun varhaisessa vaiheessa. Käyttäjäkeskeinen suunnittelu ja käytettävyyden testaus erilaisilla menetelmillä kuten heuristiikoilla ja käytettävyydesteillä, mahdollistaa verkko-opetusohjelmalle hyvän käytettävyyden.

Avainsanat ja -sanonnat: Verkko-opetus, käytettävyys, heuristiikat.

CR-luokat: K.3.1, H.1.2

1. Johdanto

Verkko-opetus on koulutuksen nykypäivää, sillä Internet tarjoaa monia koulutusmahdollisuuksia, jotka ovat ajasta ja paikasta riippumattomia. Opiskelijat ympäri maailmaa voivat osallistua samalle verkkokurssille ja jakaa tietoaan muiden kanssa. Oppiminen tapahtuu oppilaan omilla ehdoilla ja hänen oman aikataulunsa mukaan. Nykyään osa yliopistojen kurseista on mahdollista suorittaa verkkokursseina. On perustettu myös virtuaaliyliopistoja ja muita koulutuslaitoksia, joissa opetus tapahtuu pelkästään verkko-opetuksen kautta.

Internet on hyvä kanava verkko-opetukseen, sillä sen käyttö on monille ihmisille mahdollista. Se on lisäksi tehokas ja helppokäyttöinen. Jotkut tutkijat ovat sitä mieltä, että World Wide Web aktiivisena oppimisympäristönä rohkaisee luovuuteen. Web-pohjaisissa opetusohjelmissa on mahdollisuus käyttää monia eri medioita ja tapoja esittää opetusmateriaali. Se rohkaisee tutkimiseen ja tiedon etsimiseen selailun kautta ja näin ollen liittyy oleellisesti oppimiseen. [Karoulis and Pombortsis, 2003].

Internet mahdollistaa avoimien oppimisympäristöjen käytön. Se on halpa, tehokas ja helppokäyttöinen ja ennen kaikkea suhteellisen suurella osalla ihmisistä on mahdollisuus käyttää sitä. Internetin (World Wide Webin) hypermedia-rakenne tukee oppilaan kognitiivista ajattelua [Karoulis and Pombortsis, 2003]. Aktiivisena ja avoimena oppimisympäristönä WWW tarjoaa oppilaalle monia mahdollisuuksia etsiä tietoa, jakaa tietoa ja julkaista myös omia tuotoksiaan

muiden nähtävillä. Tiedon erilaiset esitysmuodot sekä helppo tiedon löytäminen auttavat oppijaa oppimistehtävissään.

WWW:n hypermediarakenne eli se, että se koostuu sivuista, jotka ovat linkitettyinä toisiinsa, tukee oppimista. Oppija voi selailta sivuja tietoa etsiessään omien mieltymyksiensä ja tavoitteidensa mukaan. Internet rohkaisee oppijaa tiedon etsimiseen ja selailuun, mikä taas edesauttaa oppimista. Internet tarjoaa monia mahdollisuuksia oppimiseen, mutta se on kuitenkin vain väline, joka oikein käytettynä tukee oppimista [Karoulis and Pombortsis, 2003]. Oppiminen prosessina vaatii aina oppijan omaa motivaatiota ja kiinnostusta aiheeseen. Oppijan asettamat tavoitteet luovat oppimiselle tarkoituksen. Motivaation lisäksi tärkeitä tekijöitä ovat myös oppijan aikaisempi kokemus sekä hänen oppimisstrategiansa. Mikään oppimisympäristö ei korvaa näitä tärkeitä tekijöitä [Karoulis and Pombortsis, 2003].

Etäopetus on teknologian kehittyessä ja Internetin myötä saanut aivan uuden muodon. Variaatioiden suuri määrä mahdollistaa verkko-opetuksen monipuolisuuden. Erilaisia opetustapoja on lukuisia, joten jokainen oppija voi löytää itselleen sopivan tavan. Verkko-opetus voi olla synkronista eli samanaikaista tai asynkronista eli eriaikaista. Yksi tehokkaan verkko-opetuksen haaste onkin oikeanlaisen synkronisen ja asynkronisen opetuksen yhdistelmän löytäminen. Muita haasteita ovat luokkahuoneaktiiviteettien sekä opettajan ja oppilaiden roolien miettiminen. [Neal and Miller, 2005].

Verkko-opetus on erittäin suosittua yliopistomaailmassa ja yleensäkin aikuisopiskelussa. Eräänä syynä siihen on se, että aikuisten tekniset taidot ovat kehittyneitä ja he pystyvät opiskelemaan itsenäisesti. Vuorovaikutteinen, kasvokkain tapahtuva opetus on useimmiten lapsille ja nuorille parempi vaihtoehto, koska he eivät ole vielä tarpeeksi kypsiä itseohjautuvaan opiskeluun eikä heillä ole verkko-opetuksessa tarvittavaa itsekuria. Lapsi voi kokea itsensä myös yksinäiseksi ja eristyneeksi jos hän opiskelee yksin tietokoneen ääressä. Vertaisryhmän seura ja sosiaaliset suhteet luokkatovereiden kanssa ovat tärkeitä asioita, jotka vaikuttavat myös oppimiseen. [Neal and Miller, 2005].

Verkko-opetuksen yhdistäminen luokkahuoneopetuksen kanssa tarjoaa lapsille uusia, mielenkiintoisia ja jännittäviä oppimistapoja. Tällaisia avoimia lapsille suunnattuja oppimisympäristöjä tullaan suunnittelemaan tulevaisuudessa yhä enemmän. Verkko-opetuksen muiden hyötyjen lisäksi tämä antaa lapsille mahdollisuuden tutustua eri kulttuureihin ja kansainvälisyyteen.

Markkinoilla on saatavilla paljon erilaisia työkaluja verkko-opetuksen toteutukseen. Usein niissä on kuitenkin keskitytty liikaa teknologiaan ja käytettävyyden on unohdettu liian pieneen rooliin. Käytettävyyden tarkoittaa ohjelman

hyvää laatua ja käyttäjän sekä hänen tarpeidensa huomioonottamista [Zaharias, 2000]. Zahariaksen [2000] mukaan käytettävyys on verkko-opetuksen laadun perusmittari. Hyvä käytettävyys on tärkeää erityisesti opiskelun onnistumisen kannalta. Jos opetusohjelmaa on vaikea tai epämiellyttävä käyttää, se estää oppilasta keskittymään itse oppimistehtävään ja näin hän helposti turhautuu.

Suuret verkko-opetuksen valmistajat eivät tee käytettävyystestejä tuotteilleen, eikä verkko-opetuksen käytettävyystestistä osata edes kunnolla keskustella, vaikka verkko-opetukseen investoidaan koko ajan yhä enemmän rahaa [Feldstein, 2002]. Käytettävyys näyttää olevan melko vieras käsitys verkko-opetuksen piirissä. Tärkeää on siis ymmärtää, mitä käytettävyys tarkoittaa juuri verkko-opetuksessa ja miten sitä tulisi soveltaa käytäntöön. Suunnittelussa keskitytään pääosin opetusohjelman sisältöön, eikä niinkään siihen, miten sisältö tulisi esittää, jotta oppijat omaksuisivat esitetyn tiedon mahdollisimman tehokkaasti.

Käytettävyyttä voi mitata esimerkiksi heuristiikkojen avulla. Verkko-opetuksen kannalta tulisi löytää sopiva mittari juuri verkko-opetusohjelmia varten, sillä muiden ohjelmien suunnitteluperiaatteet eivät päde välttämättä verkko-ohjelmien suunnitteluun. Tulisi löytää myös jonkinlainen mittari sille, miten opetusohjelman sisällöstä saa hyvin opittavan [Feldstein, 2002]. Eräs keino on suunnitella juuri verkko-opetusohjelmille räätälöityjä tarkistuslistoja ja pedagogisia suunnitteluohjeita.

2. Käytettävyys

Käytettyydyllä tarkoitetaan yleisesti sitä, kuinka helposti käyttäjä pystyy käyttämään jotakin tuotetta, tässä tapauksessa tietokoneohjelmaa tarkoituksenmukaisesti jonkin tehtävän suorittamisessa. Kun tuotteella on hyvä käytettävyys, käyttäjän tulee tietää koko ajan, miten hän pääsee tavoitteeseensa ja onnistuu tekemään haluamansa tehtävät ilman suurempia ongelmia. Tuotteen huono käytettävyys johtaa ongelmiin, saa käyttäjän turhautumaan ja voi tehdä tehtävän suorittamisesta jopa mahdotonta.

Käyttäjakeskeinen suunnittelu on erityisen tärkeää hyvin käytettävän ohjelman suunnittelussa. Käyttäjakeskeisessä suunnittelussa käyttäjä ja hänen tarpeensa ovat keskipisteessä [Zaharias, 2004]. Käyttäjakeskeisessä suunnittelussa pyritään siihen, että tuotetta on helppo ja mukava käyttää. Sovellus tukee käyttäjän toimia ehkäisemällä virheitä ja olemalla tehokas käyttäjien haluamien tehtävien suorittamisessa.

Verkko-opetuksen käytettyydyttä tarkasteltaessa on tärkeää ottaa huomioon myös verkko-oppimistyökalun tai -ympäristön ”opittavuus” [Feldstein, 2002]. Kuten Norman on sanonut, ” verkko-opetuksessa ei käytettyydyys ole pääasia,

vaan opittavuus” [Feldstein, 2002]. On mietittävä mikä tekee verkko-opetusympäristön sisällöstä opittavan? Käytettävien metodien, moodien ja työkalujen tulee olla helposti opittavia, jotta ne auttavat tiedon omaksumisessa. Näiden seikkojen lisäksi erilaiset kommunikaatiokanavat, jotka mahdollistavat vuorovaikutuksen muiden kanssa, ovat hyvin tärkeitä [Karoulis and Pombortsis, 2003]. Sovellus on käytettävyydeltään hyvä, silloin kun se täyttää viisi perussääntöä [Karoulis and Pombortsis, 2003]:

1. **Helppo oppia:** Käyttäjä pystyy suoriutumaan tehtävistään nopeasti.
2. **Tehokasta käyttöä:** Kun käyttäjä on oppinut käyttämään sovellusta, voi hän yltää suuremman tuotteliaisuuteen.
3. **Helppo muistaa:** Käyttäjän ei tarvitse muistella uudelleen systeemin eri toimintoja kun hän käyttää sitä uudelleen jonkin ajan kuluttua.
4. **Vähän virheitä:** Käyttäjä ei tee paljoa virheitä käyttäessään systeemiä ja jos virheitä ilmenee, niistä toivutaan nopeasti.
5. **Käytön miellyttävyys:** Käyttäjä on tyytyväinen käyttäessään systeemiä, sitä on mukava käyttää

Käyttäjän tulee voida käyttää sovellusta haluamallaan tavalla ja siten, että saavuttaa tavoitteensa helposti ja tehokkaasti. Teknologisen tehokkuuden sijasta käyttäjä on suunnittelun keskipisteenä. Sovelluksesta pyritään tekemään miellyttävä ja helppokäyttöinen juuri käyttäjää ja hänen tavoitteitaan ajatellen.

2.1 Käyttöliittymän hyvä käytettävyys

Verkko-opetusohjelman käyttöliittymän tulee olla selkeä, jotta oppijan on helppo löytää haluamansa asiat nopeasti ja tehokkaasti. Verkko-opetusohjelmassa on oppiminen pääasia ja käyttöliittymän tulisi tukea sitä yksinkertaisuudella ja selkeydellä. Normanin [1991] mukaan hyvään käyttöliittymään kuuluu

1. **Hyvä käsitelmä,** joka auttaa käyttäjää hahmottamaan tuotteen käytön toiminnan ja logiikan.
2. **Hyvä näkyvyys,** jolloin sovelluksessa on luonnollisia visuaalisia viestejä kertomassa eri toimintamahdollisuuksista.
3. **Selkeät kytkimet,** jotka ovat helposti ymmärrettäviä. Käyttäjä pystyy hahmottamaan helposti, mikä kytkin liittyy mihinkin toimintoon.
4. **Riittävä palaute** kertoo käyttäjälle hänen tekemistään toiminnoista ja näin ollen hän pystyy esimerkiksi havaitsemaan tapahtuneen virheen.
5. **Virheisiin varautuminen** ja virheiden ehkäiseminen.

Käyttäjakeskeisessä suunnittelussa otetaan käyttäjäryhmä huomioon. Käyttäjäryhmän ominaispiirteet ja tarpeet analysoidaan jo suunnitteluvaiheessa. Verkossa oleva yliopistokurssi on täysin erilainen kuin ala-asteikäisille suunnattu

kurssi. Kohderyhmän käyttäjäprofiilit määrittelevät pitkälti sen minkälaiseksi käyttöliittymä muodostuu.

Käytettävyys voidaan nähdä myös käyttäjän omana subjektiivisesti koetuna kokemuksena, joka liittyy käyttäjän tuntemuksiin sovellusta käytettäessä. Käyttäjän kokemusta kartoittaessa voidaan tarkastella esimerkiksi seuraavia tekijöitä: käyttöön liittyvät asenteet, käytön tyydyttävyys, käytön havaittu hyödyllisyys, havaittu käytön helppous, käytön henkinen ja fyysinen rasittavuus, turhautuminen ja suorituksen vaatavuus [Tervakari et al., 2002]. Kun puhutaan käytön miellyttävyydestä, voidaan tarkastella myös sovelluksen viihdyttävyyttä, aistillisuutta, sovelluksen tuottamaa mielihyvää sekä käyttäjän sisäistä motivaatiota. [Tervakari et al., 2002]

3. Verkko-oppiminen

Ihmisessä tapahtuva oppiminen on kokonaisvaltainen prosessi. Oppimisteorioita on monenlaisia ja vuosien aikana vallitsevat näkemykset oppimisesta ovat muuttuneet yhteiskunnan mukana. Perinteisissä oppimiskäsityksissä, kuten behaviorismissa, oppija on passiivinen tiedon vastaanottaja. Kognitiivisen oppimisteorian mukaan ihminen ohjaa itse omaa oppimistaan käyttämällä kognitiivisia taitoja kuten ajattelua ja muistamista muodostaen oppimastaan ja aikaisemmin tietämästään tiedosta jäsenyntyneitä rakenteita. Konstruktivistisen oppimiskäsityksen mukaan oppija konstruoi omaa tietoaan kokemusten kautta. Uudempi verkko-opetukseen liittyvä oppimisen näkemys on kokemuksellis-kokonaisvaltainen oppimiskäsitys, jonka mukaan oppiminen on jatkuva oppijan omiin kokemuksiin perustuva prosessi. [The Theory Into Practice Database, 2005]

Kokemuksellisen oppimiskäsityksen mukaan oppiminen tapahtuu itse tekemällä ja oppijan omien kokemusten kautta. Verkko-oppimisympäristö tarjoaa hyvän mahdollisuuden itsenäiseen tiedonhakuun ja omien kokemusten kartuttamiseen. Kokemuksellisen oppimisen teoria perustuu siihen, että kokemus muuttuu tekemisellä tietämykseksi [Conole et al., 2004]. Verkko-opetusohjelmissa oppilas voi erilaisten animaatioiden, simulaatioiden ja oppimisobjektien avulla itse harjoitella oppimiaan tietoja käytännössä.

3.1 Verkko-opetus

Verkko-opetus on 1900- luvun lopun ja 2000- luvun alun koulutustrendi, joka on koettu hyvin hyödylliseksi. On arvioitu, että vuosina 1997-1998 n. 1.6 miljoonaa oppilasta osallistui verkkokursseille [Harper et al., 2004]. Verkko-opetuksesta on tullut myös taloudellisesti kannattavaa toimintaa, johon investoidaan vuosittain yhä enemmän. Yliopistojen ja eri koulutuslaitosten lisäksi

verkko-opetusta käytetään yritysten henkilöstön koulutuksessa. Näyttää siltä, että verkko-opiskelun tarjonta tulee kasvamaan myös tulevaisuudessa [Harper et al., 2004].

Nykyajan Internet tarjoaa uudenlaisen tavan etäopiskeluun. Pelkän tiedon vastaanottamisen lisäksi opiskelija voi myös olla vuorovaikutuksessa opettajan tai muiden opiskelijoiden kanssa. Internetin kautta opiskelija voi lähettää opiskelumateriaalia opettajalle tarkistettavaksi ja keskustella muiden opiskelijoiden sekä opettajan kanssa.

Verkko-opetuksen tärkeitä elementtejä ovat ne eri kanavat, joiden kautta tietoa voidaan välittää. Kuva, ääni, video ja teksti tekevät opetuksesta rikkaamman ja mielenkiintoisemman. Kun näihin elementteihin lisätään kommunikointi muiden opiskelijoiden ja tutoreiden kanssa, saadaan oppimiskokemuksesta vielä parempi. Verkko-opetuksen tärkein hyötytekijä on sen antama vapaus opiskelijalle. Opiskelija saa itse valita ajan ja paikan sekä oman opiskelutahtinsa opiskellessaan verkkokurssilla.

Verkko-opetus tarjoaa opiskelijalle mahdollisuuden osallistua videokonferensseihin, ryhmäkeskusteluihin verkossa sekä erilaisiin on-line -yhteisöihin, joilla on sama kiinnostuksen tai opiskelun kohde [Desanctis et al., 2003]. Verkossa tapahtuva opiskelu antaa mahdollisuuden tiedon etsimiseen, tiedon tuottamiseen sekä suhteiden luomiseen osallistujien välille [Descantis et al., 2003].

Tulevaisuudessa verkko-kurseille osallistuu yhä enemmän ihmisiä eri maista ja kulttuureista. Se tuo oman rikkautensa opiskeluun. Suunnittelussa tulee huomioida opiskelijoiden erilaiset taustat, esimerkiksi kulttuurierot. Verkko-opetus ylittää maantieteelliset rajat, mutta sen lisäksi se voi rikkoa myös kulttuurisia sääntöjä, normeja ja opetustapoja [Harper et al., 2004].

Verkko-opiskelun onnistuminen vaatii opiskelijalta myönteistä asennetta ja tavoitteellista opiskelua. Opettajien ja tutoreiden tulee valmistella verkko-kurssit huolella, jotta niistä olisi paras mahdollinen hyöty opiskelijoille. Kurssien huolellinen suunnittelu antaa opiskelun ja oppimisen mahdollisuuden. Jos opiskelijat kokevat verkkokurssin teknologian hyödyttömäksi tai hankalaksi, kokemus omasta oppimisesta jää huonoksi.

Oppilaiden tulisi aina sitoutua siihen, että he suorittavat verkkokurssit sovituksessa aikataulussa [Harper et al., 2004]. Vaikka verkkokurssi antaakin ajasta ja paikasta riippumattoman vapauden, vaaditaan opiskelijalta myös päättävyyttä ja sitoutumista, jotta hän saa verkko-kurssin suoritettua. Verkko-opetuksessa, kuten muussakin etäopetuksessa juuri oppijan rooli korostuu. Perinteisessä opetuksessa opettajan rooli on erittäin tärkeä, mutta verkko-opetuksessa opettaja katoaa miltei kokonaan [Neal and Miller, 2005]. Tämän vuoksi vuoro-

vaikutus jää vähäiseksi ja opiskelija saattaa tuntea ajoittain yksinäisyyttä oppimistehtäviensä kanssa.

Sisällöllisten seikkojen lisäksi verkkokurssien suunnittelussa tulee aina kiinnittää huomiota myös esimerkiksi kurssiaikatauluun, oppilaan ohjaukseen, sisällön kehittämiseen, tekniikan ja välineiden valitsemiseen sekä muihin tärkeisiin seikkoihin [Harper et al., 2004]. On myös tärkeää huomioida, että kaikilla opiskelijoilla ei ole samanlaisia teknisiä taitoja. Tämän takia hyvän käytettävyyden merkitys on suuri, jotta opetusohjelmien käyttö olisi mahdollisimman helppoa ja yksinkertaista.

4. Verkko-opetuksen käytettävyys

Verkko-opetuksen käyttöliittymän eli käyttäjälle avautuvan näkymän WWW-sivusta tulee vastata käyttäjän tarpeita. Verkkosivuston käyttöliittymän tulee tukea opiskelua tarjoamalla helppokäyttöinen väline digitaalisen oppimateriaalin esittämiseen. Kun verkko-oppimissivusto on käytettävyydeltään hyvä, se ei kuormita käyttäjän kognitiivisia prosesseja vaan auttaa informaation jäsentämisessä ja käsittelemisessä. [Tervakari et al., 2002]

Käytettävyys on tuotteen mitattava ominaisuus, mutta sitä ei ole kuitenkaan standardisoitu [Notess, 2001]. Sen vuoksi ei voida sanoa esimerkiksi, että ohjelmisto on 80 %:sti käytettävä. Erityisesti verkko-opetuksen käytettävyyden määrittelyssä sen mittaus voi olla hankalaa. Sen sijaan voidaan kuitenkin luoda käyttäjäprofiili kuten millainen käyttäjä on, ikä, status, teknologian käyttökokemus, tehtävät eli oppimistavoitteet ja konteksti esimerkiksi kotona, koulussa, työpaikalla [Notess, 2001]. Lisäksi voidaan mitata tehtävän suoritukseen kulunut aika ja suorituksen taso, virheiden määrä sekä käyttäjän tyytyväisyys [Notess, 2001]. Nämä ovat jo merkittäviä mittareita joiden avulla voidaan pyrkiä johonkin tiettyyn käytettävyydestavoitteeseen. Muita mittareita voivat olla myös esimerkiksi se, miten paljon opetusohjelmaa voidaan käyttää ilman avuntarvetta tai vaikka sitä, kuinka moni suoritti verkkokurssin loppuun [Notess, 2001].

Verkko-opetuksen käytettävyydessä käyttäjän eli oppijan rooli korostuu vielä enemmän kuin ns. tavallisessa sovelluksessa. Yleisesti sovellusten avulla suoritetaan jokin tehtävä, esimerkiksi lasketaan yrityksen kuukausitulot, ohjataan jotakin konetta tai suoritetaan laskuoperaatio. Opetusohjelmassa sen sijaan tavoiteltu muutos tapahtuu itse käyttäjässä: hän oppii jotain. Tämän vuoksi hyvä käytettävyys ja näin ollen käyttäjän tarpeiden huomioiminen on erittäin tärkeää. Suunnittelussa tulee ennen kaikkea selvittää, millainen käyttäjä on, mitkä ovat hänen ominaispiirteensä, motivaationsa ja missä ympäristössä hän

toimii [Notess, 2001]. Käyttäjäprofiilin tarkka määrittely auttaa opetusohjelman tehokkaassa suunnittelussa.

Tervakarin ja muiden [2002] mukaan opetusohjelmien suunnittelussa tulee aina ottaa huomioon yksilön kognitiiviset toiminnot. Käytettävyydestä tutkimuksella ja kognitiivisella psykologialla on yhteiset juuret, sillä käytettävyyssuunnittelu liittyy olennaisesti ihmisen ajatteluun, muistiin, havaitsemiseen, ongelmanratkaisuun ja oppimiseen. Havaitseminen, tarkkavaisuus, muisti ja ongelmanratkaisu liittyvät erityisesti sovelluksen näkyvyyteen. Käyttäjän tulee havaita, miten tehty toiminto vaikuttaa järjestelmässä, eikä kuormittaa muistiaan esimerkiksi muistelemalla sitä, mitä nappia hän juuri painoi. Käytettävyyden kannalta sovellus ei koskaan saisi kuormittaa liikaa käyttäjän muistia vaan olla niin selkeä, että sitä on kokoajan helppo käyttää.

Käyttöliittymän ja toimintojen yhdenmukaisuus sekä niissä käytetyt tutut tavat vähentävät käyttäjän muistiprosessien määrää. Tarkkaavaisuus taas liittyy siihen, että huomaa ja vastaanottaa sovelluksen antaman palautteen. Sovelluksessa mahdollisesti ilmaantuvien ongelmien aikana käyttäjä tarvitsee ongelmanratkaisukykyään. [Tervakari et al., 2002]

Ongelmatilanteissakin sovelluksen tulee auttaa käyttäjää toipumaan virheistä ja parhaassa tapauksessa estää käyttäjää tekemästä niitä. Mainittujen esimerkkien lisäksi käyttäjä käyttää koko ajan melkein pä kaikkia kognitiivisia toimintojaan opetusohjelmaa käyttäessään. Sen vuoksi kognitiivisen psykologian tarjoamaa tietoa ihmisen aivojen toiminnoista voidaan käyttää hyväksi opetusohjelmien suunnittelussa.

4.1 Verkko-opetusohjelman suunnitteluperiaatteet

Verkko-opetusohjelmaa suunniteltaessa keskeinen asia on se, että ohjelmaa on miellyttävää käyttää. Koska sovelluksen käytön tavoitteena on oppiminen, pääasia suunnittelussa on käyttäjä ja hänen tarpeensa [Notess, 2001]. Käyttäjän osallistuminen suunnitteluun antamalla mielipiteitä, kritiikkiä ja ideoita on tärkeää.

Opetusohjelmaa suunniteltaessa ei suunnitella vain tuotetta vaan oppimiskokemusta oppijalle [Notess, 2001]. Opetusohjelmiin käytetään samoja suunnitteluperiaatteita kuin muidenkin ohjelmien suunnitteluun. Verkko-opetusohjelma ei ole käyttökontekstiltään sama asia kuin ns. tavallinen ohjelma [Notess, 2001]. Tehokkuuden lisäksi ohjelman tulee vastata oppimistavoitteisiin ja oppijan tulee saada oppimiskokemuksia sen kautta.

Suunnittelun kannalta paras tapa saada tietoa opiskelijoista on tehdä kenttätutkimusta [Notess, 2001]. Opiskelijoiden ja opettajien kanssa keskusteleminen sekä heidän toiveitaan kuuntelemalla saadaan hyvää suunnitteluma-

teriaalia. Sen lisäksi aikaisemmat oppimiskokemukset antavat tärkeää tietoa siitä, mikä on hyvää ja mikä huonoa opetusohjelmissa ja verkkokursseissa.

Kun suunnitellaan verkko-opetuksen materiaalia, fokus siirtyy oppimiseen erinäisten tehtävien sijaan [Notess, 2001]. Oppimistuloksien saavuttaminen hyvin, tehokkaasti ja mielekkäästi ovat tärkeitä kysymyksiä suunnitteluvaiheessa [Notess, 2001]. Chydenius-Instituutin Verkostoyliopiston sisältötuotantohankkeissa verkko-opetuksen suunnittelun ja toteutuksen periaatteiksi on valittu mielekkyys, monimuotoisuus, joustavuus, käytettävyys ja visuaalisuus. Nämä periaatteet ovat kertyneet verkkokurssien suunnittelu- ja toteutusprosessien, kokemusten ja arviointitutkimusten kautta [Lehto, 2003]. Juuri kokemuksen ja käyttäjiltä saadut arviot sekä kritiikki ovat tärkeitä asioita uusien verkko-opetuskokonaisuuksien suunnittelussa.

Opiskelijan tulee olla jatkuvasti suunnittelun keskipisteessä, sillä ohjelmaa suunnitellaan juuri hänen käyttöönsä. Lehdon [2003] mukaan keskeisimmät suunnittelun pedagogisessa käsikirjoituksessa käsiteltävät asiat ovat kohderyhmän erikoispiirteiden kartoittaminen, verkko-opetuksen tavoitteiden ja sisältöjen suunnittelu sekä oppimateriaalin määrittely.

4.2 Pedagoginen käytettävyys

Verkko-opetusohjelman tulee käytettävyyden lisäksi olla hyödyllinen. Verkko-opetuksessa hyödyllisyys muodostuu ensinnäkin Internetin käytön oppimiselle ja opetukselle tuomasta lisäarvosta sekä verkko-opetusohjelman pedagogisesta käytettävyydestä [Tervakari et al., 2002]. Pedagogisessa käytettävyydessä kiinnitetään erityisesti huomiota verkko-opetusohjelman rakenteeseen, sisältöön sekä opiskeluprosessin tukemiseen. Hyödyllisyys ja käytettävyys yhdessä muodostavat sovelluksen käyttökelpoisuuden.

Tervakarin [2002] mukaan pedagogiseen käytettävyyteen kuuluu muun muassa oppimistavoitteiden tukeminen, opiskeluprosessin tukeminen, opiskelijan autonomian tukeminen, vuorovaikutustaitojen tukeminen sekä opiskelutaitojen kehittymisen tukeminen. Pedagogisella käytettävyydellä tarkoitetaan sitä, miten hyvin verkko-oppimisympäristön käyttöliittymä, oppimistehtävät ja muut tehtävät, sisältö sekä työkalut motivoivat ja tukevat erilaisten opiskelijoiden oppimista tietyssä opiskelukontekstissa ja tiettyjen pedagogisten tavoitteiden mukaisesti [Tervakari et al., 2002].

Horilan ja muiden [2002] mukaan pedagogisella käytettävyydellä tarkoitetaan käytettävyyttä oppimisen ja opettamisen näkökulmasta. Sen lisäksi painotetaan oppimateriaalin käytön soveltuvuutta eri oppimistilanteisiin, oppimateriaalin tehokkuutta, tavoitteellisuutta sekä lisäarvoa oppimiselle verrattuna normaaleihin oppimateriaaleihin. Verkko-opetuksen oppimateriaalin käytettä-

vyyttä on arvioitu paljon. Lähtökohtana ovat usein olleet tekniset ominaisuudet sekä oppimateriaalin sisältö.

4.3 Verkko-opetuksen visuaalinen suunnittelu

Verkko-opetus mahdollistaa erilaisten ja monipuolisten opetuskeinojen kuten kuva, äänen, tekstin ja videon käytön. Erityisesti opetusohjelman visuaalisuus ja visuaalinen selkeys ovat erittäin tärkeitä asioita, jotka tulee ottaa suunnittelussa huomioon. Selkeä ja rauhallinen yleisilme auttaa käyttäjää hahmottamaan keskeisen tiedon sekä eri elementit sivulla [Tervakari et al., 2002]. Esteettisyyden lisäksi asettelu, värien, muotojen ja tyhjän tilan käyttö liittyvät ohjelman visuaaliseen suunnitteluun [Tervakari et al., 2002]. Yleisilmeen selkeys ja yhtenäisyys ja se on tärkeää myös käytön tehokkuuden ja opittavuuden kannalta.

Grafiikan avulla voidaan opiskelijalle visuaalisesti ilmaista esimerkiksi siirtymiseen eli navigointiin liittyviä asioita. Hahmottelun ja värien avulla voidaan luoda oppijalle haluttuja mielikuvia ja erotella osioita toisistaan. Jopa väri itsessään on viesti, joka sisältää assosiatiiivisia ja symbolisia merkityksiä, kuten esimerkiksi tiettyjen aatteiden värisymboliikka. [Stång, 2003]

Toisaalta myös mahdolliset rajoitteet kuten värisokeus, heikkonäköisyys sekä tekniset taidot pitää aina ottaa suunnittelussa huomioon. Myös kulttuuriset erot esimerkiksi värien käytössä on tärkeä ottaa huomioon.

5. Verkko-opetusohjelmien käytettävyyden arviointi

Tässä luvussa esitellään erilaisia verkko-opetuksen käytettävyyden arviointimenetelmiä. Tärkeimpiä työkaluja ovat erilaiset heuristiikat, eli sääntölistat, joiden avulla voidaan arvioida sovelluksen käytettävyyttä. Heuristiikkojen käyttämistä arvioinnissa pidetään erityisen hyvänä keinona verkko-opetuksen arvioinnissa, sillä se on nopeaa, tehokasta ja halpaa [Feldstein, 2002].

Heuristiikkojen lisäksi käytettävyydestit ovat tärkeitä, sillä ne antavat tietoa oikeasta käyttötilanteesta ja siinä esiintyvistä ongelmista. Käytettävyydestejä suositellaan tehtäväksi mahdollisille loppukäyttäjille tai ainakin kohderyhmään sopiville henkilöille, jotta saadaan mahdollisimman tarkkaa tietoa sovelluksesta.

Verkko-opetuksen käytettävyyden suurin mittari on ”opittavuus” eli se, että saavutetaan tietty tarkoin määritelty ja konkreettinen oppimistavoite [Feldstein, 2002]. Tästä näkökulmasta keskeisiä ovat kognitiiviset tavoitteet eli se, miten oppija oppii ja ymmärtää verkko-opetuksessa esitetyt asiat. Kognitiivisen psykologian eri teorioita hyväksi käyttäen pystytään esittämään tieto sellaisessa muodossa, että se on helppo ymmärtää ja oppia. Kognitiivinen tavoite määritel-

lään siten, että se muistuttaa läheisesti sitä tavoitetta, jonka oppija yrittää saavuttaa [Feldstein, 2000]. Kognitiiviset teoriat ovat tärkeässä asemassa silloin, kun pyritään selittämään aivojemme toimintaa oppimistilanteessa sen sijaan, että miettisimme, mitä opettelemme [Feldstein, 2000].

Käytettävyyden testaamisen ja erilaisten läpikäyntien ensisijaisena tavoitteena on pyrkiä löytämään mahdolliset käytettävyysongelmat mahdollisimman aikaisessa vaiheessa [Tervakari et al., 2002].

5.1 Heuristiikat

Nielsen [2000] on koonnut kymmenen hyvän käytettävyyden sääntöä eli heuristiikkaa, joiden avulla tietokoneohjelman käytettävyyttä voidaan arvioida. Heuristiikat tarjoavat eräänlaisen mittarin, joita voidaan käyttää apuna ohjelman suunnittelussa ja arvioinnissa. Käytettävyydestauksen lisäksi heuristinen arviointi on tärkeää ohjelman käytettävyyden arvioinnissa. Nielsenin heuristiikat antavat hyvän pohjan ohjelman arvioinnille. Säännöt käydään kohta kohdalta läpi ja katsotaan vastaako ohjelma niiden vaatimuksiin. Nielsenin [2000] heuristiikat ovat seuraavat:

1. **Näkyvyys.** Käyttäjän tulee olla koko ajan tietoinen mitä sovelluksessa tapahtuu. Sovelluksen tulee antaa riittävää palautetta koko ajan.
2. **Yhteensopivuus systeemin ja todellisen maailman välillä.** Sovelluksessa ei saa käyttää käyttäjälle vierasta kieltä, outoa tai liian teknistä sanastoa. Sisältö pitää esittää luonnollisesti ja loogisessa järjestyksessä.
3. **Käyttäjän kontrolli ja vapaus.** Käyttäjällä tulee aina olla mahdollisuus päästä pois epämiellyttävästä tilasta tai palata askel taaksepäin.
4. **Jatkuvuus ja standardit.** Sovelluksessa ei saa esittää samaa asiaa monella eri tavalla. Esimerkiksi tietty kuvake tekee aina tietyn toiminnon. Suunnittelussa pitää noudattaa tiettyjä ennalta määrättyjä standardeja.
5. **Virheiden ehkäisy.** Monien virheilmoitusten sijaan käyttäjää pitäisi tarkan suunnittelun avulla ehkäistä tekemästä virheitä esimerkiksi kysymällä käyttäjältä, haluaako hän varmasti tehdä jonkin toiminnon.
6. **Muistikuormituksen minimointi.** Käyttöliittymien osien tulee olla näkyviä ja selkeitä, jotta käyttäjän ei tarvitsisi yrittää muistaa, mitä eri objektit tarkoittavat. Käyttäjän muistia ei saa kuormittaa ja ohjeiden tulee olla helposti näkyvillä ja saatavilla.
7. **Käytön tehokkuus ja joustavuus.** Käyttöliittymän tulisi tarjota ekspertti-käyttäjille oikopolkuja eri toimintoihin. Oikopolut eivät kuitenkaan saa sekoittaa aloittelevaa käyttäjää. Näin käyttöliittymä on monipuolisempi erilaisille käyttäjille.

8. **Esteettinen ja minimalistinen suunnittelu.** Näytöt eivät saa sisältää tarpeetonta tai epäolennaista tietoa, sillä kaikki ylimääräinen tieto kilpailee olennaisen tiedon kanssa ja näin ollen vähentää sen näkyvyyttä.
9. **Virheistä toipuminen.** Virheilmoitukset tulee ilmaista selkeällä kielellä ja siten, että ne kuvaavat suoraan kyseisen ongelman sekä ehdottavat ratkaisua.
10. **Ohjeet.** Sovelluksen tulisi mieluiten olla käytettävissä ilman apua, mutta usein ohjeet ovat välttämättömiä. Ohjeiden tulisi olla näkyvillä, helposti saatavilla ja niiden tulisi kohdistua suoraan käyttäjän ongelmaan. Ohjeiden tulee ohjata käyttäjää ongelman ratkaisussa. Ohjeet eivät saa olla liian pitkiä.

Nielsenin heuristiikat ovat hyvänä pohjana kaikkien sovelluksien suunnittelussa, mutta verkko-opetusohjelmilla on tiettyjä oppimistavoitteiden kannalta tärkeitä seikkoja, jotka tulisi myös ottaa huomioon. Suunnitteluun tulee kehittää myös pedagogisia ohjeita, jotta ne sopivat yhteen heuristiikkojen kanssa [Zaharias, 2000].

Benson ja muut [2001] ovat laajentaneet Nielsenin heuristiikoista verkko-opetukseen sopivan arviointimenetelmän. Tärkeimmät ongelmat tunnistettiin ja etsittiin niihin ratkaisut. Arvioinnissa on tärkeintä antaa tarkat ohjeet siitä, miten opetusohjelman käytettävyyttä tulisi parantaa. Nielsenin kymmenen heuristiikan sijaan Benson ja muut määrittelivät yhteensä 20 heuristisen arvioinnin sääntöä. Ensimmäiset kymmenen myötäilevät Nielsenin heuristiikkoja. Niiden lisäksi arvioidaan verkko-opetusohjelmien interaktiivisuutta, information esittämisen muotoa, opetuksen suunnittelua, opetuksen arviointia, media-integraatiota, tietolähteiden käyttöä, tukityökaluja, opiskelijan itsearviointia, palautetta ja sisältöjä. [Benson et al., 2001.]

Erilaisten tarkistuslistojen käyttöä on kritisoitu, sillä tarkistuslistojen mukainen arviointi ei vastaa kuvaa todellisesta käyttötilanteesta eli sovelluksen kautta oppimisesta. Lisäksi vain tarkistuslistojen perusteella tehty arviointi ei ole riittävä, sillä se ei ota huomioon, että oppiminen on myös sosiaalinen ja konstruktivinen tapahtuma [Karoulis and Pombortsis, 2003]. Oppiminen tapahtuu ihmisessä itsessään ja sovellus on siinä vain välineenä. Suunnittelussa pitää siis erityisesti ottaa huomioon pedagogiset seikat, kuten tavat, joilla materiaali esitetään, tehtävien muodot ja vuorovaikutuksen laatu. Vaikka opiskelija saa opiskella ajasta ja paikasta riippumatta, ei ole tarkoitus, että hän eristäytyy ja opiskelee aivan yksin. Vuorovaikutus muiden opiskelijoiden sekä opettajan kannalta on oppimisen kannalta tärkeää.

Jotta verkko-oppimisympäristöjä voitaisiin arvioida monipuolisemmin, tarvitaan myös muita tapoja kuten haastatteluja, kenttätutkimuksia ja käytettävyydestejä. Tärkeää on huomioida opettajien kokemukset ohjelmien käytöstä sekä myös opiskelijoiden omat mielipiteet ohjelman toimivuudesta. Verkko-oppimisympäristön tulee olla käytettävä oppijoille, mutta sen lisäksi sen tulee olla tehokas ja helppo käyttää myös opettajille, jotka rakentavat ja vetävät verkkokursseja.

5.2 Käytettävyydestit

Käytettävyyden testaamisessa hyödynnetään sovelluksien todellisia käyttäjiä testaamalla ohjelmistoa testihenkilöiden avulla käytettävyysslaboratoriossa [Tervakari et al., 2002]. Testiympäristö on kuten normaali työhuoneympäristö, ainoana lisänä on se, että kamerat ja mikrofonit tallentavat testihenkilön toiminnan ja puheen, jotta tilanne voidaan analysoida tarkasti myöhemmin. Käytettävyydestissä pyritään käyttämään testattavaa sovellusta sillä tavalla kuin sitä tullaan käyttämään normaalissa yhteydessä. Testihenkilöiksi pyritään saaman sellaisia henkilöitä, jotka vastaavat sovelluksen kohderyhmää.

Testissä testihenkilö saa tehtäväkseen yksinkertaisia tehtäviä, jotka hänen tulee itsenäisesti suorittaa. Tehtävän tekemisen aikana hänen tulee ääneen kertoa, mitä on tekemässä. Näin ollen ongelmien ilmetessä saadaan arvokasta tietoa siitä, mikä sovelluksen käyttämisessä on vaikeaa, epämiellyttävää tai epäloogista.

Verkko-opetuksen arvioinnissa käytettävyydestit ovat nopea ja tehokas tapa huomata suunnitteluvirheet ja jopa 5-6 testiä antaa kattavat tulokset [Notess, 2001]. Testit ovat hyviä selvittämään pääasian eli sen, tietävätkö käyttäjät, mitä ja miten heidän tulee tehdä [Notess, 2001]. Kuitenkaan käytettävyydestit eivät anna aivan todellista kuvaa oikeasta käyttötilanteesta. Vaikka oikeaa käyttötilannetta ja sen ympäristöä simuloidaan tarkkaan, tietää käyttäjä kuitenkin, että kyse on testitilanteesta. Ääneen puhuminen ei tunnu kaikista käyttäjistä luonnolliselta eikä hän välttämättä huomaa tai uskalla kertoa kaikkia ajatuksiaan, joita tilanne hänessä herättää.

Eräs hyvä keino kerätä materiaalia on tehdä kenttätutkimusta ja mennä esimerkiksi kouluihin tarkkailemaan oppimistilanteita. Myös erilaiset haastattelut ja kyselyt ovat hyviä tapoja kerätä syvällisempää tietoa käyttäjän kokemuksista.

6. Yhteenveto

Parhaimmillaan oppija voi kokea huippuelämyksiä ja seikkailuntunnetta hyvin suunnitellussa ja toteutetussa verkko-opetuksessa [Lehto, 2003]. Jotta verkko-

opetuksesta voitaisiin saada hyviä kokemuksia, tulee opetusohjelman toimia tarkoitettulla tavalla käyttäjän tarpeet huomioon ottaen. Kun verkko-opetusohjelman hyvä käytettävyys varmistetaan jo suunnitteluvaiheessa, tulee opetusohjelmasta laadukas ja miellyttävä käyttää.

Erilaiset käytettävyyden testauksen menetelmät, kuten heuristiikkojen käyttäminen sekä käytettävyydestien tekeminen, varmistavat sovelluksen hyvän käytettävyyden. Verkko-opetusohjelmat tarvitsivat myös juuri niille räätälöityjä tarkistuslistoja, sillä opetusohjelmien erityiset käytettävyyshaasteet tulee ottaa huomioon. Tavallisten ohjelmistojen testauksessa käytettävät tarkistuslistat eivät ole kattavia verkko-opetusohjelmien testauksessa. Testaajan tulee ottaa huomioon kaikki ohjelman erilaiset vuorovaikutteiset osiot ja niiden tarkoituksenmukaisuus ja pedagogisesti oikea toiminta. Ei riitä, että oppilas saa tavanomaisesti palautetta ohjelmalta, vaan hänen tulee saada sellaista palautetta, joka auttaa ymmärtämään mahdolliset virheet ja antaa ratkaisuvaihtoehtoja.

Verkko-opetuksen suunnittelussa keskitytään usein liian paljon teknologiaan ja unohdetaan käyttäjäkeskeinen suunnittelu. Opetusohjelmien tekijät eivät välttämättä panosta käytettävyyteen, sillä ohjelmia halutaan tehdä nopeasti ja tehokkaasti. Silloin ongelmat ilmenevätkin vasta loppukäyttäjän käyttäessä ohjelmaa. Tietenkin opetusohjelman materiaali eli sen sisältö on tärkeässä asemassa, mutta myös sen esitystavalla on merkitystä oppimisen kannalta.

Viiteluettelo

- [Benson et al., 2001] Lisa Benson, Dean Elliot, Michael Grant, Doug Holschuh, Beaumie Kim, Hyeonjin Kim, Erick Lauber, Sebastian Loh and Tom Reeves, Heuristic Evaluation Instrument and Protocol for E-Learning Programs. Draft September 5, 2001. Available as <http://it.coe.uga.edu/~treeves/edit8350/HEIPEP.html>
- [Conole et al., 2004] G. Conole, M. Dyke, M. Oliver and J. Seale, Mapping pedagogy and tools for effective learning design. *Computers & Education* **43** (2004), 17-33.
- [DeSanctis et al., 2003] Gerardine DeSanctis, Anne-Laure Fayard, Michael Roach and Lu Jiang, Learning in online forums. *European Management Journal* **21**, 5 (October 2003), 565-577.
- [Feldstein, 2002] Michael Feldstein, What is "usable" e-learning? *eLearn Magazine* (August 2002).
- [Harper et al., 2004] Kimberly C. Harper, Kuanchin Chen and David C. Yen,

- Distance learning, virtual classrooms, and teaching pedagogy in the Internet environment. *Technology in Society* **26**, 4 (November 2004), 585-598.
- [Horila et al., 2002] M. Horila, P. Nokelainen, A. Syvänen ja J. Överlund, Pedagogisen käytettävyyden kriteerit: kokemuksia OPIT-oppimisympäristön käytöstä Hämeenlinnan normaalikoulussa syksyllä 2001. Hämeenlinna: Hämeen ammattikorkeakoulu
- [Karoulis and Pombortsis, 2003] Athanasis Karoulis and Andreas Pombortsis, On heuristic evaluation of web-based ODL program. In: Claude Ghaoui (ed.), *Usability Evaluation of Online Learning Programs*. Information Science Publishing, 2003, 88-109.
- [Lehto, 2003] Sini Lehto, Verkkokurssi – elämänmakuinen oppimiseikkailu? *Chynetti nro 27*, Chydenius-Instituutti, Jyväskylän yliopisto, 2003.
- [Neal and Miller, 2005] Lisa Neal and Diane Miller, The basics of e-learning. *eLearn Magazine* (July 2005).
- [Nielsen, 2000] Jakob Nielsen, Ten Usability Heuristic. November 22 2005
Available at http://www.useit.com/papers/heuristic/heuristic_list.html
- [Norman, 1991] Donald Norman, *Miten avata mahdottomia ovia? Tuotesuunnittelun salakarit*. Weilin+Göös, 1991.
- [Notess, 2001] Mark Notess, Usability, user Experience, and learner experience. *eLearn Magazine* (August 2001).
- [Stång, 2003] Virpi Stång, Turhaa koristelua? Visuaalisuus verkko-opetuksessa. *Chynetti 29* Chydenius-Instituutti, Jyväskylän yliopisto, 2003.
- [Tervakari et al., 2002] Anne-Maritta Tervakari, Pekka Ranta ja Heidi Kaartokallio, *Tietoverkkoavusteisen opetuksen käyttökelpoisuus Käytettävyys ja pedagoginen käytettävyys opetuksen organisoinnin näkökulmasta*. Tampereen teknillinen yliopisto DMI/Hypermedialaboratorio, 2002.
- [The Theory Into Practice Database, 2005] Available at <http://www.gwu.edu/~tip/theories.html>
- [Zaharias, 2004] Panagiotis Zaharias, Usability and E-learning. *eLearn Magazine* (May 2005).

Information Foraging

Juha Leino

Abstract.

This paper strives to give a complete and in-depth overview of the information foraging theory, an adaptationist psychological theory concerned with human behavior in the context of information gathering for some purpose, in the context of the Internet. Developed by Pirolli and Card, the theory has quickly risen to a prominent status in the Human-Information Interaction¹ (HII) research in addition to becoming a household name in the usability circles. The research has already developed computational cognitive models for simulating human information foraging behavior, and the practical tools utilizing them are beginning to emerge. The information foraging theory and its concepts concern everybody involved in HII and usability.

Keywords: Information foraging, information scent, information diet, information patches, computational cognitive models, LSA, ACT-R

CR Classification: H.1.2

1. Introduction

“Information foraging is the most important concept to emerge from Human-Computer Interaction research since 1993” – Jakob Nielsen [2003]

Information foraging theory is an adaptationist psychological theory that attempts to understand and provide theoretical framework for how the human information seeking mechanisms and strategies adapt to the information environment where they operate [Pirolli, 2004]. According to it, we behave in the Internet while searching for information like, in the words of Nielsen [2003], “wild beasts in the jungle”. While this may sound like a joke and is certainly an oversimplification, it does vividly describe something essential about the information foraging theory and its implications.

Animals have highly optimized formulas as to where, when and what to eat. These formulas have developed through evolution, and have resulted in optimal foraging behavior that is still present in our intuitive approach to the information environments that typically bear close resemblances to the real

¹ According to Lucas [2000], Nahum Gershon coined the term Human-Information Interaction to denote how people interact with, relate to and process information regardless of the medium between the two in 1995. Computers become less visible in the interaction, and finally, invisible. We will ultimately interact with information objects.

food-foraging environment [Nielsen, 2003; Pirolli and Card, 1999]. Consequently, as informavores, we try to optimize our gain from the interaction. We want to minimize our costs in relation to the valuable information we gain [Pirolli, 2004].

The information foraging theory applies many concepts of the optimal foraging theory to today's information environments that give rise to the human information foraging behavior [Olston and Chi, 2003; Pirolli, 2004]. It divides the information ecology into information patches, and uses the concept of information scent to explain our cost-value evaluations based on which we exploit the current patch, follow the scent within the patch or move to another patch. Information scent is the subjective cost-evaluation we make based on prior experience and proximal snippets about the distal object. [Pirolli and Card, 1999] The relationship of the proximal cue to the distal object is merely probabilistic, so we cannot trust it explicitly to give us a complete and faithful picture or description of the distal information object [Pirolli, 2005].

In addition, we must evaluate the current patch in relation to other patches in the environment. We do not necessarily know much about the other patches, so leaving the current patch to pursue another involves costs, namely opportunity cost and resource cost. Consequently, the informavore forages in a probabilistically textured environment where we use the environment itself as one factor when forming subjective cost-value evaluations [Pirolli, 2004].

The information foraging research has already developed computational cognitive models to simulate human-information interaction behavior [Pirolli and Fu, 2003; Chi et al., 2001]. The research continues, and the researchers already vision automated tools for Internet site information scent and usability evaluations. In addition, the researchers are envisioning automated tools into the browser development and even the development of adaptive information environments where the tools we use adapt to our foraging behavior [Chi et al., 2001; Pirolli, 2004; 2005].

While it is possible to apply the information foraging theory to most information seeking situations, this paper concentrates on its implications in the context of the Internet. Using Internet as an example allows us to make the theoretical concepts concrete. We foremost discuss at the information foraging theory in detail and introduce its most important concepts. However, we also look at the current state of the information foraging research and discuss its implications for the Internet-related information searching, designing and usability.

2. Background of the information foraging theory

Peter Pirolli and Stuart Card of Xerox Palo Alto Research Group developed information foraging theory in mid-1990 [Nielsen, 2003; Gattis, 2002]. Information foraging theory addresses Human-Information Interaction involving the modern information technologies, such as World Wide Web. It is “concerned with human behavior and technology involved in information gathering for some purpose” [Pirolli, 2005]. Information foraging theory is concerned with humans who must develop adaptive methods of getting and using external information to gain knowledge in order to help them in decision making and problem solving in addition to emphasizing the influence of the information landscape and the adaptation to it [Pirolli, 2005; Pirolli and Card, 1999].

Pirolli and Card based the information foraging theory on the foraging models of human behavioral ecology studied by anthropology since mid-1970 [Gattis, 2002]. The information foraging theory also builds on the evolutionary-ecological theory in biology that underlies the adaptationist psychology and the theories related to it [Pirolli 2004]. Adaptationist psychology framework combines the analysis of the structure of the environment with the analysis of the design rationale of the human psychological mechanisms and knowledge to understand how humans adapt to their environmental conditions [Pirolli 2004].

Thus, to understand a cognitive mechanism underneath our behavior we have to start by “defining its function in relation to the environment” [Pirolli, 2004]. A certain type of environment results in certain types of psychological mechanisms that are efficient if not optimal for surviving in that environment, and those psychological mechanisms are traceable back to the environment. This reverse engineering approach means that we ask what environment-related problem the psychological mechanism overcomes, why the mechanism is a good way to overcome the problem and how the mechanism realizes the solution [Pirolli 2004]. The effects of this thinking are evident in the reverse engineering thinking of IUNIS (Inferring User Need by Information Scent) that we discuss later.

One evolutionary-ecological theory that has had a significant effect on the information foraging theory is the optimal foraging theory. Optimal foraging theory brings into the information foraging theory the assumption of ecological rationality. Ecological rationality means simply that the actions of an organism in an environment are rational, that is, work for the survival and well being of the organism in that environment. [Pirolli and Card, 1999; Pirolli 2004]

However, the rationality does not mean to imply that each individual organism always and under all conditions is somehow able to make the most

rational selection even if such could somehow be defined. Rather, it means that the selective forces of the environment eventually select the better-suited biological and behavioral models. [Pirolli, 2005]

Optimal foraging theory attempts to explain the adaptation of an organism structure and behavior to the challenges and food foraging constraints present in its environment [Pirolli and Card, 1999]. Originated in ethological studies of food seeking and prey selection of animals, the optimal foraging theory has been used to explain dietary choices, habitat choices, time allocation and other aspects of a hunter-gatherer society [Pirolli and Card, 1999]. We meet these concepts later again when we discuss the central concepts of information foraging theory.

The information foraging theory places an equal emphasis on the users and their information environments. Pirolli [2004] says that Brunswik's probabilistic functionalism has inspired this symmetrical approach. Furthermore, Brunswik's Lens Model has been a partial inspiration of the concept of information scent [Pirolli, 2004].

Information foraging theory uses a lot of foraging-related vocabulary, which makes its concepts vivid and easy to understand. Pirolli and Card [1999] say that they would like to think that the information foraging adaptations evident in today's information foragers are exaptations of the foraging behavior evolved for food foraging in the past, but they feel that it is unlikely that we can trace this evolution.

3. Information cost

Ecological rationality assumption of the information foraging theory means that we try to get as much as possible for as little as possible. The less effort we have to exert ourselves to get food, the better it is for our survival. Thus, we are under evolutionary pressure to optimize our Internet usage to get maximum benefit for minimal effort. [Nielsen, 2003; Pirolli 2004] This relation is evident also in the studies that have researched what information is read and cited most often. No surprisingly, it is the information for which access cost is the lowest. [Pirolli and Card, 1999]

There are different kinds of costs involved in searching information from the Internet. The most obvious one is the opportunity cost. Opportunity cost means that the resources used in doing something are not available to do something else. Therefore, we judge what we choose to do with some resources more valuable than what we choose not to do with the same resources. When we talk about the resources used in seeking information for some purpose in the Internet, the primary ones are attention and time. The time we spend on

something, such as the time we spend accessing, rendering and interpreting certain information-carrying items [Katz and Byrne, 2003] in the Internet, is the time not available for another information carrying item or something completely else as is with attention. When we pay attention to something, we are not paying attention to something else.

Long waits, such as waiting for a page to load over 10 seconds, in the Internet (AKA World Wide Wait) have been proven to result in reduced search time at the expense of decision quality in addition to lost business opportunities [Dennis and Nolan, 2005]. Smoothness of the interaction is essential, or the users will leave us for the better foraging grounds [Nielsen, 2003]. If the cost is high, the informavore adapts its behavior accordingly within the constraints.

The other type of cost involved is the resource cost [Pirolli and Card, 1999]. This cost includes the resources spent in acquiring the valuable information. In the context of the information foraging theory, these typically consist of financial costs of acquiring the information [Pirolli and Card, 1999]. For instance, if we pay for the Internet connection or access time, there is clearly a resource cost involved in using the Internet.

The amount of information available to us has grown exponentially thanks to the rapid development of Internet [Pirolli and Card, 1999]. While there has always been cost involved in the sense making of available information [Pirolli and Card, 1999; Chen et al., 2002], the current amount of available information involves its own costs in search times and sense-making activities. For instance, the huge number of results returned by a simple Google query leaves us to try to make some sense out of it.

In accordance with the ecological rationality, Pirolli [2004] asserts, "human-information interaction systems tend to maximize the value of external knowledge gained in relative to the cost of interaction". Pirolli [2004] further observes that this statement is consistent with Resnikoff's observation that natural and artificial information systems develop towards stable states that minimize the unit cost of valuable information.

The more useful information a cognitive strategy yields per unit cost, the more efficient it is [Pirolli and Card, 1999] and the more likely the users adopt it [Pirolli, 2004]. Several studies attest to the information cost having strong effects on the foraging behavior [Dennis and Taylor, 2005; Katz and Byrne, 2003]. Overall, the information foraging theory argues that there are costs, such as accessing, rendering and interpreting the information, involved in the information foraging and that the users adapt their foraging strategies to optimize and maximize their rate of obtaining valuable information while trying to keep the costs down. [Pirolli and Card 1999; Katz and Byrne, 2003]

The concept of information cost is of utmost importance when we discuss the information diet, information patches and information scent, and how they affect our Internet searching behavior in the next section.

4. Information diet

Information diet in the information foraging theory is closely related to the information cost. In any environment, there are various potential food sources. The predator must construct a suitable diet in relation to these potential sources available to optimize its gain of energy per unit cost. In nature, it appears that the predators even ignore low-gain prey to concentrate on higher-gain prey. The maximization calculation – how to maximize the gain per unit cost – is present all the time. [Pirolli and Card, 1999] Easiness of the prey is naturally part of this consideration – the lower the cost, the better gain ratio [Nielsen, 2003].

In the same way, the informavore or information forager must balance its diet in relation to its cost. The informavore must be able to rapidly categorize its prey types (web page type), assess their prevalence on the plain (web locality), assess the gain they offer (return in relation to the cost of pursuit) and determine which categories to pursue and which to ignore. [Pirolli et al., 1996]

Another consideration is that the information must be sufficiently relevant to the task [Pirolli and Card, 1999]. Furthermore, some dietary elements might be essential and thus not substitutable. For these items, the cost of information is a secondary consideration since we must have them. [Dennis and Taylor, 2005]

In the optimal foraging theory, the diet models are static. In contrast, the diet models in the information foraging theory are dynamic. The state of the forager, his or her information need, changes and develops through the interaction with the information, such as the consumption of certain information, and by other factors, such as looming deadline that restricts the time for foraging. This means that the information scent following model of the information foraging theory has to be dynamic as well. [Pirolli and Card, 1999]

Information scent, which will be discussed in depth later, refers to the subjective perception of the distal information at the end of a proximal link that offers us our limited “preview” of the information and influences our subjective perception of the costs involved in reaching that information. Based on this imperfect scent of the distal information, we must decide if we follow the link or not. [Olston and Chi, 2003; Pirolli, 2004] As the interaction with the information shapes our information needs, we are attracted to different scents [Pirolli and Card, 1999].

Information environment tends to be patchy in nature. This has relevance to the information cost and diet as the structure of the environment has a significant effect on our foraging strategies and decisions on what to feed. What and how we feed depends on the environment where we feed. [Pirolli and Card, 1999; Pirolli, 2004; Card et al., 2001] The availability of information, that is, the cost involved in obtaining the information, is an essential consideration when we ask ourselves what information we pursue.

5. Information environment

5.1. Information patches

As discussed, the information foraging theory maintains that the information environment structures our interactions with it [Pirolli, 2004]. With the theory, we essentially address the question how the user strategies and technologies involved in the information seeking, gathering and consuming are adapted to the flux of information in the given environment [Card et al., 2001]. With that in mind, let us take a closer look at the nature of the information environment.

The task environment of the informavore tends to be patchy in structure [Card et al., 2001]. The information ecology in the Internet, for instance, divides into patches that vary in many dimensions. These patches divide further into patches that form hierarchies. The patches are defined by family resemblance, such as topic of the content. [Card et al., 2001] Topical arrangement of the Internet is evident in the way the sites are interlinked. Greater textual similarity exists between sites that are linked to each other than between random sites. [Pirolli, 2005] This also supports the findings that browsing is an effective method of foraging for information relative to searching [Katz and Byrne, 2003].

Card *et al.* [2001] have identified the following structural patches in the Internet ecological environment:

- **Web** – Users often define their information goals in terms of pinpointing information or searching in the entire web.
- **Web site** – After locating a suitable web site, the users tend to search within the site. Web sites are large patches. They can be divided into subclasses, such as portal/index sites or search engines.
- **Page** – A web page contains link descriptors and content elements, among other elements. In a way, these elements further divide the page into patches. The pages can also be divided into various

subclasses, such as Hitlist page, Search engine page etc. The user forages for information on the page by visual search and scrolling.

The content elements in the pages are naturally the end point of a search in the Internet. It is the consumption of the content that bestows the user the reward for foraging. [Card et al., 2001]

The patchy nature of the information environment means that we need to allocate the foraging time between two behaviors. We have the within-patch (exploiting an information item) and between-patches time (searching for the next item for exploiting) [Pirolli and Card, 1999]. We can see the inherent problem easily in its biological context. Let us say that a bird eats berries growing on one type of bush. The bushes grow at uneven distance from each other. The bird is eating from one bush, but as the berries get less and less, the bird is at some point forced to move to the next bush. However, it does not know how far is the next bush and how much food it may offer. Now the optimization task for the bird is to decide when to leave the current bush and spent uncertain amount of time on searching for the next one.

Thus, how long should we stay in one page searching for information or consuming information (within-patch foraging) and when should we start to move to the next page or to another site altogether. The moving from one patch to another involves costs that we need to take into consideration. Following links takes time, and we need to orient to the next page before we get down to business -information foraging and consumption - there. Back-navigating takes time. Finding a new site with search engine using keywords takes significant amount of time. Consequently and in keeping with cost-based analysis, Pirolli [2004] observes that the users tend to move more within a site than from site to another.

Naturally, the lower the cost of moving to the next patch, the faster we are likely to dump a patches which information returns are diminishing [Pirolli and Card, 1999; Nielsen, 2003; Pirolli, 2005]. Consequently, Nielsen [2003] states that the better the search engines, the faster they offer us more relevant pages, the less time the users spent on one page.

Theoretically, each page in the Internet is competing against all the other pages in the Internet. Even if each page is not competing against all the other pages then at least against the pages of similar content. The better the search engines, the easier - or cost-effective in the parlance of the information foraging theory - it is for the users to leave a certain page in search of another one. Consequently, the web designers and content providers had better take care

that their web sites support the foraging behavior of today's informavores. [Nielsen, 2003]

This naturally leads to a question of on what the foragers base their decision of staying or leaving. The information foraging theory claims that the decisive factor in this cost-calculation that aims to optimize our information-consumption is the information scent [Pirolli and Card, 1999]. However, before discussing the information scent in depth, let us consider another environment relevant to the information foraging, that of the task environment.

5.2. Task environment

The interaction of the user with the information environment is subject to influence by the task environment. Task environment refers to the environment combined with the goal or problem that provides the essential motivation for the user to interact with the environment. The task defines the forager's viewpoint to the environment and thus allows the forager to interpret it according to his or her needs and concentrate on what is essential for him or her under the task environment. [Pirolli, 2004; Blackmon et al., 2003] This quintessentially means that the user sees the environment in terms of his or her needs, and this allows him or her to delimit the environment. Furthermore, the task environment includes the relevant aspect of the physical, virtual, social and cognitive environments that constitute and drive human behavior [Pirolli, 2004].

Naturally, this viewpoint is subjective and shaped by user's prior experiences and background [Blackmon, 2002; Pirolli and Fu, 2003]. Consequently, the user interaction with the environment is stochastic, not deterministic, by nature [Pirolli and Fu, 2003].

It is also important to keep in mind that when we are dealing with modern information environments, the means we use in the interaction bring their own twist to the tale. The technological systems work as "cognitive prostheses" for us when we search and view the information [Pirolli, 2004]. The way the environment is presented to us is largely determined by the technology we use. For instance, when interacting with the Internet information ecology, we typically do so with the help of a browser and search engines (themselves part of the environment). While allowing us the basic means of interaction in the first place, they simultaneously limit our interaction possibilities by placing constraints. Thus, the continuous development of the means of interaction is naturally of a great interest to the research and the audiences that use them [Pirolli et al., 1996; Pirolli, 1999; Chi et al., 2001] in addition to business interests involved.

An example of ongoing research based on the information foraging theory that aims to improve the perception of information scent in the web pages is ScentTrails that researches analyzing the distal information in order to be able to give the user clearer information scent based on user searches [Olston and Chi, 2003].

When we talk about the Internet environment, we are typically dealing with a need for further information rather than looking for a particular snippet of information that would answer one particular, clearly defined problem. The problems identified significant in everyday life usually require us to get more information in order to define our problems clearer and finally take action. We are affected by both what we know and what we do not know. [Pirolli, 2004]

Furthermore, the information foraging tasks in which we engage are typically embedded in the context of another, bigger task. We need the information for some purpose, and after the gathering and sense-making processes we turn it into action. Such ill-defined problems usually have a clear goal, but solving them requires large amounts of heterogeneous information. Thus, the cost and value structure of the information is related to this embedding task and changes over time. Again, our interaction with the information influences the equation. What we get to know dynamically redefines our problem and affects our task definition, thus underlining the dynamic nature of the information foraging. [Pirolli and Card, 1999]

When we discuss the task environment, it is good to bear in mind also the limitations of the information foraging theory. The theory presumes that the user is looking for information. When we discuss the Internet browsing as whole this might be a dangerous assumption. Occasionally, we browse the net purely to kill time or to amuse ourselves. Yes, we follow scents as well when browsing associatively from one point of shifting interest to another, but our goal is not any more clearly defined and thus it does not have such a strong effect on the process as when looking for information for a particular goal, as discussed earlier. We follow the scents as they rise and capture our interest, but there is no big, underlying goal directing this process.

This spells trouble under certain conditions for certain applications based on the information foraging theory. For instance, if we try to use log data to infer use need, as with the computational cognitive model IUNIS (Inferring Use Need by Information Scent), we have to be careful where we apply it. Information foraging theory works well on situations where the user is searching for information but we must exercise caution instead of applying it discriminately to all and any interaction in the Internet. Being aware of the user's task environment is important when interpreting his or her actions.

Overall, our goals and other factors of the task environment are in a state of flux as the process of information foraging and consumption results in constant re-evaluations of our information needs. They color and control our viewpoint to the information environment, and thus our interaction with the environment that, in turn, influences our behavior. Consequently, the information foraging is a very dynamic process that is stochastic by nature.

5.3. Enrichment

The traditional optimizing foraging theory and related patch models have presumed the environment to be unmalleable. Therefore, the forager has no other choice than to adapt to the environment. When it comes to the information forager, however, this is no longer true. The information forager can mold its environment to fit its strategies. This process of molding the environment to fit the foraging strategies is called enrichment. [Pirolli and Card, 1999]

The enrichment can be done in two ways. First, the forager can minimize the between-batch time, that is, decrease the cost involved in moving from one information patch to another. The optimizing question involved is how much time to devote to enrichment in relation to exploiting the patches. [Pirolli and Card, 1999] Enrichment, after all, does not directly give us information.

The second way is to improve the within-patch foraging gains to make foraging yield richer information harvest. Pirolli and Card [1999] mention such means as refining keyword queries as an example of improving the within-patch foraging. The trade-off here is the same as above: how much time should we allocate to the enrichment as opposed to the exploitation?

The reply to this dilemma consists of such considerations as the current information scent available, the task environment and the information environment. The environments define the constraints to our enrichment activities and contribute to the information scent that plays an important role in determining a solution to this cost-efficiency dilemma.

6. Information scent

Information scent is the foremost concept to arise from the information foraging theory. Quintessentially, the information scent is the imperfect, subjective perception of the value of the information at end of the scent and the costs involved in getting to it based on which we decide if we follow the scent or not [Olston and Chi, 2003; Pirolli, 2004].

When we browse the Internet, we judge the distal content based on the proximal cues available to us in the pages. What the proximal cues give is the

information scent, and based on this scent we decide if we pursue the prey – the information – or not. [Pirolli, 2004] The proximal cues in the context of the Internet refer to the text snippets and images that typically constitute links [Chi et al., 2000]. However, these snippets can also consist of and/or be augmented by many other elements, such as music, video and animations [Pirolli 2004].

However, it is good to keep in mind that the environment of these cues also matters. The other elements on a page and the link-location among them matter in finding and interpreting the links. First, contextual cues, such as the texts and heading, provide additional information for our sense-making mechanisms. Secondly, the page arrangement and design matters. [Card et al., 2001] Depending on the design of the page, prominent items can capture our attention and thus influence the attended and thus noticed information scents significantly [Katz and Byrne, 2003]. Moreover, we tend to understand and interpret the pages according to the genres or schemas we have. Our expectations rising from our understanding of the genre influence our visual search and sense making among the elements. [Norman, 1999] We do not delve here into the influence of genres in the context of information theory, as this appears to be a very under-researched area. However, this is an area where more research would be interesting, as its results would be interesting for designing purposes.

As mentioned, the information scent provides us the basis on which we decide if we follow a link or not. One inherent problem is how well the proximal cue represents the distal information. The relationship is only probabilistic [Pirolli, 2005]. According to Pirolli [2005], the division into proximal and distal cues comes from Brunswik's ecological theory of perception and judgment in a probabilistically textured environment. Because our perception is indirect, Brunswik emphasized analyzing the relationship between the observer and the proximal cues, the cognitive strategy, and the relationship between the proximal cue and the distal object, the ecological validity (Figure 1). [Pirolli, 2004]

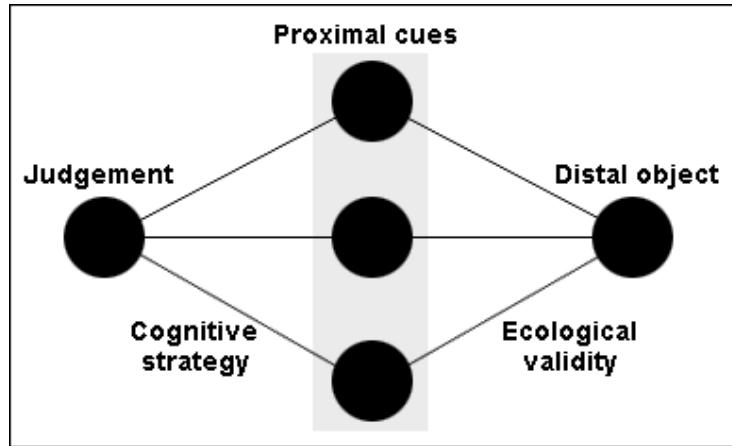


Figure 1: Brunswik's Lens Model. [Pirolli, 2004]

As mentioned, in a probabilistically textured environment the relationship between the proximal cue and the distal object cannot be trusted to be absolutely correct [Pirolli, 2004]. If it were, then the information scent would also be absolutely reliable. The accuracy of the information scent either increases the cost of foraging or decreases it by either increasing or decreasing the necessary browsing time [Pirolli, 2005]. Furthermore, the depth has a further impact on the equation. The further the target object and more branching the hierarchy, the more important the accuracy of the information scent [Pirolli, 2005]. Figure 2 shows the difference between perfect scent, random search and full search.

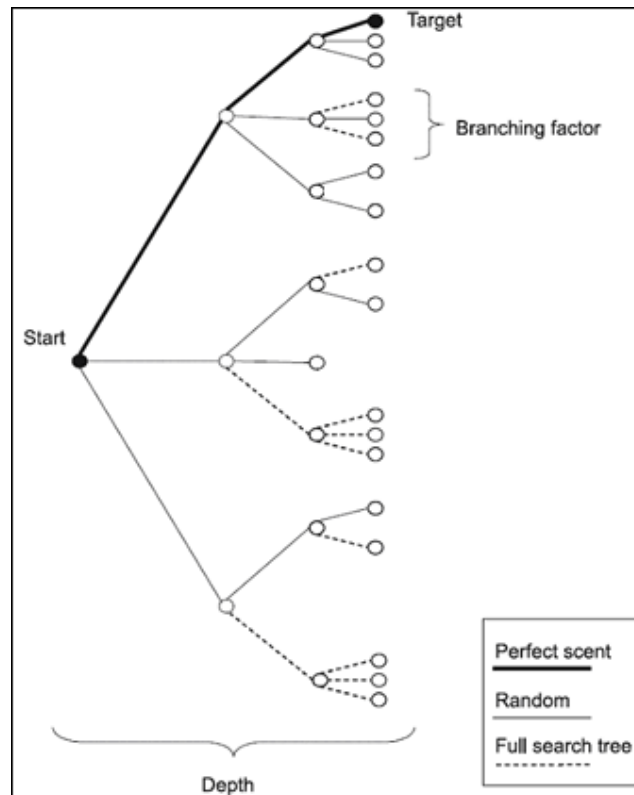


Figure 2: How perfect information scent decreases the information cost. [Pirolli, 2005]

Browsing in a hierarchy of any depth therefore involves performing hill-climbing with the information scent as heuristic for deciding the next step [Pirolli and Fu, 2003]. We judge if we are getting closer to the prey by seeing if the information scent is getting stronger². If the information scent gets stronger, we keep on clicking. If, on the other hand, the scent seems to wane, we have to re-evaluate the situation. [Nielsen, 2003]

In addition to following the information scent through links and menus, we use the general information scent of the page to judge if the information patch still offers us valuable information. If it appears that we have exhausted the valuable information from the patch or that the potential gains of continuing foraging the patch have diminished to such degree that other patches offer us more even with the involved between-patch cost considered, we move on. [Pirolli, 2004; Pirolli and Fu, 2003] Card *et al.* [2001] have used the term patch scent policy to describe this behavior as opposed to scent following behavior.

Reliability of the information has not been relevant in the tasks used by Pirolli and other developers of the information foraging theory. Much of the research has used click stream data and data mining in user log data [Blackmon et al., 2002] or using Xerox PARC personnel with data that they knew to be reliable. In all cases reviewed for this paper, the participants did not have to make any decisions as to the information reliability in the tasks. In the real world tasks, however, the reliability of the material tends to be more or less relevant when we search for information for our ill-defined problems. If, for instance, we are looking for medical information, the reliability of the information is of utmost importance. Reliability evaluation is certainly one part of the process that goes into the information scent evaluation. It affects both our scent following behavior and patch scent policy. If we judge the distal information unreliable, we are less likely to follow that scent, and if we judge the current patch information unreliable, we move on. Information reliability is one aspect of the information scent that should be studied in future.

One important question that rises from the concept of information scent is what does it mean in the cognitive terms as far as the human being is concerned. What is the process in our minds that takes place when we

² Incidentally, this throws the old, unsubstantiated “3-click-rule” out of the window. While the information scent keeps on getting stronger, the users do not appear to care about the number of clicks. They know by the scent that they are onto a good thing.

formulate the information scent and could be possible somehow to build cognitive models to simulate it?

6.1. Spreading activations and ACT-R architecture

The information foraging theory model of how humans judge the information scent is based on spreading activations [Pirolli and Fu, 2003]. The spreading activation models are neurally inspired and they have been used in the simulations of human memory [Pirolli, 2004].

The information goal of the user activates a set of chunks (cognitive structures) in his or her memory. The proximal snippets, such as links on the web page, then activate another set of chunks. Activation spreads from these chunks to the related chunks through a spreading activation network. The amount of activation accumulation on the goal-activated chunks (representing the needed information) and proximal cue-activated chunks indicates their reciprocity. [Pirolli and Fu, 2003]

The weighting values of the associations between the chunks determine the spreading of the activation between them. These weights determine the rate of activation flow between the chunks. If we consider this in the context of the Internet usage, our goal has activated a set of chunks, i.e. focus of attention, in our minds and the activation spreads through memory associations, often semantic in nature, to what we see on the page, typically words and images. These associations have strengths and weights that determine the quantity of activation that spreads from one chunk to another – the stronger the association, the bigger the amount of activation flow [Pirolli, 2004]. If, for instance, a link text has a strong association in our mind to our goal, we judge the information scent to be strong. [Pirolli and Fu, 2003] Consequently, the spreading of activations is based on our past experiences, prior knowledge and associations in our mind [Pirolli, 2004; 2005], and are thus individual and subjective.

In other words, the external information scent cues activate cognitive structures in our minds, and the activation spreads from these structures to related ones in a spreading activation network. The larger the amount of activation that accumulates on the information goal chunk, the stronger the information scent, that is, the likelihood that the distal object is what we need. [Pirolli, 2004]

This approach is based on ACT-R³ framework [Blackmon et al., 2002; Pirolli and Fu, 2003; Pirolli, 2004; Pirolli and Card, 1999]. ACT-R based models, such

³ ACT-R is a cognitive architecture for simulating and understanding human cognition [ACT-R, 2002-2003]. ACT stands for Adaptive Control of Thought [Ritter and Shiskowski, 2003]

as SNIF-ACT⁴, simulate spreading activations in the cognitive chunks and the following selection of production rules, quintessentially condition-action pairs. ACT-R consists of two memory components, declarative knowledge and procedural knowledge. Declarative knowledge contains the things that we are aware that we know and that we can describe to others. The chunks form the declarative memory in ACT-R. The chunks have activation values and the activations spread through associations in the declarative memory among the chunks as discussed earlier. [Pirolli, 2005; Pirolli and Fu, 2003]

Procedural knowledge, on the other hand, specifies how the declarative knowledge activations are translated into actions, i.e., behavior. The procedural knowledge is presented in the ACT-R architecture as production rules, that is, condition-action pairs. The condition of the production rule is matched against the user goal and the active chunks in the declarative memory. If a match is found, the action side of the production rule is executed. [Pirolli, 2005; Pirolli and Fu, 2003]

Pirolli and his associates, such as Ed Chi, James Pitkow, Wai-Tat Fu and Peter Card, have used this model of spreading activations in the development of their computational cognitive models, such as ACT-IF⁵, SNIF-ACT, WUFIS⁶ and IUNIS⁷, to simulate the user behavior in the Internet [PARC, 2002; Pirolli and Fu, 2003; Chi et al., 2001, Pirolli and Card, 1999].

While a more detailed look at these models is beyond the scope of this paper, we take a general look at them. All these models are based on spreading activations in addition to being extensions of the information foraging theory and information scent.

ACT-IF is “a computational cognitive model of information foraging” [Pirolli and Card, 1999] that is an extensive revision of ACT-R, a cognitive architecture for simulating and understanding human cognition [ACT-R, 2002-2003], that incorporates information foraging theory predictions [Pirolli and Card, 1999].

WUFIS and IUNIS, developed by Ed Chi, Peter Pirolli, Kim Chen and James Pitkow, are interesting computational models of the interaction between information goal and information scent, and how it affects navigation. WUFIS, Web User Flow by Information Scent, attempts to simulate the user actions based on their needs while IUNIS, Inferring User Need by Information Scent,

⁴ SNIF-ACT: “Scent-based Navigation and Information Foraging in the ACT architecture” [PARC, 2002]

⁵ ACT-IF: ACT stands for Adaptive Control of Thought [Ritter and Shiskowski, 2003] while IF stands for Information Foraging [Pirolli and Card, 1999]

⁶ WUFIS: Web User Flow by Information Scent [Chi et al., 2001]

⁷ IUNIS: Inferring User Need by Information Scent. [Chi et al., 2001]

attempts the same feat in reverse, namely inferring the user need based on user actions. According to their tests, both models work reasonably well when their predictions are compared to human user results. [Chi et al., 2001]

One aspect that IUNIS analysis of the user actions did not take into account is the time spent on the page. The weighting of the documents is based on giving more weight to the recently visited pages on the assumption that they better describe user needs, and by trying to give more weight to the content pages as opposed to the site splash/index page [Chi et al., 2001]. Site splash pages are identified by prior knowledge or by higher access statistics [Chi et al., 2001]. However, the time spent on page calculation could add more intelligence to the evaluation of the relevance of the page to the user need, and the future studies should address it as well.

SNIF-ACT, "Scent-based Navigation and Information Foraging in the ACT architecture" [PARC, 2002], models and simulates users performing unfamiliar information-seeking tasks in the Internet [Pirolli and Fu, 2003]. Pirolli and Fu [2003] have tested the system against log file information of actual human users, and it has shown good ability to stimulate real users even though scent-based utilities are stochastic and subject to variability due to the users and context.

6.2. Latent Semantic Analysis

While Pirolli and his associates have used spreading activation networks as the way semantic connections are evaluated for information scent, Blackmon *et al.* [2002] have used LSA (Latent Semantic Analysis) for determining the information scent in developing CWW (Cognitive walkthrough for the Web). LSA is a mathematical technique for estimating the semantic relatedness of texts based on a statistical analysis of large corpus [Blackmon et al., 2002]. Like spreading activation, LSA is related to neural network models [Landauer, 1998].

In fact, there is no great difference between the approaches in many aspects. In their research, Pirolli and his associates originally based their estimates of semantic relatedness of goals and proximal cues to the estimates derived from Tipster corpus, a database that contains the "statistics relevant to setting the base level activations of 200 million word tokens and the inter-word association strengths of 55 million word pairs" [Pirolli and Fu, 2003]. For SNIF-ACT, Pirolli and Fu [2003] supplemented the statistical database obtained from Tipster with word frequency and word co-occurrence statistics generated from the Internet itself using AltaVista search engine.

LSA analysis, also based on statistical analysis of large corpus, has one special feature. When doing semantic comparisons with LSA, one must choose a semantic space that corresponds the closest to the intended reading level of the user group. Blackmon *et al.* [2002] define semantic space as a “principled representation of the semantic knowledge of a given user population” that attempts to give an estimation of the selected population’s perception of semantic similarity. LSA web site⁸ offers five semantic spaces for American-educated English speakers based on average reading comprehension and background knowledge levels: first-year college student, grade 12, grade 9, grade 6 and grade 3. [Blackmon et al., 2002] Thus, LSA can reflect the differences in the different population groups instead of treating all as one, as the spreading activations approach currently does.

The main difference lies, in fact, in the underlying cognitive architectures to model human cognition used in the two approaches. While the cognitive spreading activations approach as based on ACT-R [Blackmon et al., 2002; Pirolli and Fu, 2003; Pirolli, 2004; Pirolli and Card, 1999], LSA is, according to Blackmon *et al.* [2002], based on CoLiDeS (Comprehension-based Linked model of Deliberate Search) model that, in turn, is an extension of Kintsch’s construction-integration model of text comprehension and problem solving.

While ACT-R based models, such as SNIF-ACT, simulate spreading activations in the cognitive chunks and following selection of production rules [Pirolli and Fu, 2003], CoLiDeS suggests that selecting the next action is a two-phase process. The first phase, attention phase, takes genres and schemas into consideration in that it maintains that during this phase, the user segments the Internet page into a collection of subregions. Then the user generates a brief description for each subregion based on his or her knowledge of the layout conventions of the web pages and the textual cues, such as the headings, on the page. The user attends then to the subregion which description best matches his or her current goal. [Blackmon et al., 2002]

During the second phase, action selection phase, the user concentrates on the control devices, such as textual and image links in the web page, in the selected subregion, generates descriptions for them and then acts on the one which description best matches his or her current goal. [Blackmon et al., 2002]

Both approaches appear to produce good results, and Blackmon *et al.* state that CWW is, in effect, meant to extend the WUFIS. In return, Pirolli [2005] acknowledges that Blackmon *et al.* have had success in applying LSA as a way of computing information scent. While information scent is a good, proven concept, the work to reproduce it with computational cognitive models

⁸ <http://lsa.colorado.edu/>

continues and many different approaches are necessary. Spreading activations model and LSA appear to be the most popular approaches currently.

7. Implications of the information foraging theory in the web

7.1. Making the web sites for the informavores

The information foraging theory is easy to apply to practical problems, in part because of its vivid, figurative terminology. Especially in the context of the Internet, the theory and the research it has spawned offer us many concrete means to evaluate our designing approaches and further develop them [Chi et al., 2001; Blackmon et al. 2002; Olston and Chi, 2003; Pirolli, 2005]. The theory gives us a clear understanding of the user behavior and needs, and that is the best starting point that a designer can have. In effect, Pirolli [2004; 2005] maintains that the information foraging theory has become a design concept in web site design and usability.

The foremost message the information foraging theory has for the designers is that the sites and the pages must support the information foraging behavior of the users. Evolutionary forces act upon the web pages, and it is the survival of the fittest, plain and simple.

While humans consume information, the information also needs to consume humans for its survival. The survival rates of web pages (as opposed to deletion) correlate strongly with the amount of visitors they get. [Pirolli and Card, 1999] As Nielsen [2003] so delicately puts it, "it's healthy to remember that users are selfish, lazy, and ruthless in applying their cost-benefit analyses [bolding removed]". To improve our sites, we need to make sure that they support the foraging behavior of the informavores.

The first step is actually something that the information foraging theory has not yet researched much as it has concentrated on the Human-Information Interaction. Because our interaction with the Internet information ecology takes place with the assistance of technical devices, typically browsers, and we typically search for information with search engines, such as Google and AltaVista, there is a healthy amount of Machine-Information Interaction involved as well. This part of the information foraging, the technical interaction aspect, is something that should be integrated into the theory in future. Consequently, our sites must be search engine and spider friendly in order to get the users to come in the first place. We must build the sites so that they support search engines and their rating systems, and that the information they present to the user in their result listings has as strong information scent as possible.

Once we have attracted the user to our site by giving strong scent first to the search engine and then through it to the user, we must keep him or her in our site. Here the evolution works for us. Since moving to another patch has a high price in comparison to the patch exploitation [Pirolli, 2004], the user is inclined to stay if we can give him or her a strong, unambiguous information scent. This is where the computational cognitive models help us.

The information scent has been researched a lot, and we already have good tools and strategies to help us make sure that the information scent of our site is in good order. First, we must respect the genre [Norman, 1999] and its conventions, and organize the content in such a way that it is easy for the user to segment the page into subregions that have clear purposes [Blackmon et al., 2002].

Good organization does not only apply to individual pages but also to the whole site. The content organization must support easy navigation and building good information scents that efficiently guide the users to the information content [Gailey, 2002].

Katz and Byrne [2003] found that the site information structure influenced strongly the decision to browse or to search. The better the information scent in the menu items and links, the more likely the users were to browse and the lower the scent, the more inclined they were to use search functions. Of course, the graphical design can direct the user attention to searching or browsing options, but the information scent plays a crucial role in this decision. This supports the information foraging concept that the information environment affects the user behavior. [Katz and Byrne, 2003]

Katz and Byrne [2003] further found that the users prefer using broad rather than deep menus. Giving a clear information scent throughout a deep menu can present problems that broader menus do not have.

Informavores want to maximize the returns for the effort exerted, and therefore we must make getting the valuable information as easy as possible for them [Nielsen, 2003; Pirolli 2004]. Therefore, we must make the content as readable as possible.

Writing for the Internet is different from writing for other media. Nielsen [1997] suggests that in the Internet people do *not* read, they scan. Consequently, our content must support this method of visually scanning the content. One of the most important advises Nielsen [1997] gives is the use of inverted pyramid style of writing, meaning that the important content and conclusions come first, and then come explanations and other content that is interesting only to a part of the audience. In addition, the text should be shorter than in other media. We

must write concisely and precisely. The headings must be clear, not cute or marketese. [Nielsen, 1997; Gailey, 2002]

Navigation and links are what makes hypertext what it is. They give us the ability to chunk information and create links between them. It is also in them that we must create the information scent for the users to follow to the juicy tidbits of information. The wording of the links and headings around them are very important in creating unambiguous information scent. First, the links and menu items must give a clear description of the distal content [Blackmon et al., 2002]. The user must have clear idea of what the distal content would be from the link.

Information scent is semantic by nature. Thus, when writing link texts we must find the words that represent the distal information object unambiguously to the user population. The words that create an especially lucid information scent in a certain context are called trigger words. Which words are trigger words depends on the target population and the content context [Spool, 2004]. They are the words that the goal has semantically primed in the user's mind. People who find the information they need are typically those who have given words that appear on the home page when describing their target before the searching task [Spool, 2004].

It is important to use words that are familiar to the user population [Blackmon et al., 2002]. This is the reason why for using semantic spaces in the LSA. If medical experts build a site for laymen to use, it is difficult for them to know what is understandable for their users. This is where user testing and use of systems such as LSA necessary. However, in order to use them, we have to know our user population.

Furthermore, the links and menu items must be unambiguous also in relation to each other. They have to be different enough to allow the user to differentiate their distal contents based on them [Blackmon et al., 2002].

Another aspect that the information foraging theory has not addressed thus far is the information scent after the visit to the site is already over. If our site has served the user needs and supported their foraging behavior, they might add our site to their bookmarks. The title texts that show in the bookmarks and bookmark icons are our tools to make our information scent evident to the user when he or she visually scans his or her bookmarks. The same rules concerning writing the links discussed above apply here, too.

A lot of what the information foraging theory tells us about the behavior of the users is something that other studies concerning usability have also discovered. However, information foraging theory provides these pieces of information a uniting cognitive theory about human adaptive foraging

behavior in the modern information environments. As discussed in the next section, the theory is also a springboard for technical development that allows us to present and use the information available to us more efficiently.

7.2. Technical tools and future development

The information foraging theory has given rise to a lot of research into its implication and how to utilize them in shaping our information environments, how to design them better and provide us better tools to use them. Below is just a short exemplary look at the current research and future visions.

The CWW (Cognitive walkthrough for the Web) [Blackmon et al., 2002] is a tool for overcoming semantic problems in the link and menu item text. It evaluates the texts for semantic problems while taking user population and distal objects into account.

SNIF-ACT, IUNIS and WUFIS also help us to understand the user goals and needs and allow us to make predictions as to what effects changing the design would have on the information scent of the site by simulating user behavior in addition to helping us evaluate our current designs [Chi et al., 2001; Pirolli and Fu, 2003]. While they are not widely available yet, they are likely to be integrated to many applications that ease the work of the designers [Pirolli and Fu, 2003; Pirolli 2004; Chi et al., 2001]. Pirolli [2004] even envisions that such automated usability tools could take the user goals as input and then simulate the user behavior in a given site. The systems could perform usability analysis and report on problem areas.

While the tools based on the concepts of the information foraging theory and awareness of user behavior it has brought to us influence our designing work, the theory also is likely to result in improvements in our technical tools, such as web browsers, for browsing the information environments that will adapt to our foraging behavior. The research also could provide us tools to evaluate the design of these tools automatically. [Pirolli, 2005]

The furthers reaching visions outline the development of information searching and browsing tools that automatically adapt to and optimize our foraging behavior in relation to our goals and needs and the information environment [Chi et al., 2001; Pirolli, 2004]

Overall, Pirolli [2005] considers the Human-Information Interaction improvements to lead to improvements in human intelligence. He expands Newell's proposal that intelligence is the ability to use one's whole knowledge in the service one's goals by stating that the better we can bring the right knowledge out at the right time to serve our goals, the more intelligent we are. The external world teems with knowledge, and what improves our ability to

bring out the right information out at the right time, improves our intelligence. [Pirolli, 2005]

8. Discussion

The information foraging theory, an adaptationist psychological theory, is concerned with the adaptive behavior of humans in Human-Information Interactions [Pirolli, 2004]. Based on the foraging models of human behavioral ecology and evolutionary-ecological theory [Gattis, 2002], it explains the human behavior in information seeking situation, such as using the Internet [Pirolli and Card, 1999]. The information foraging theory claims that the adaptive behaviors evident in today's information foragers are exaptations of the foraging behavior that evolved for food foraging in the past [Pirolli and Card, 1999].

Although a relatively new theory, the information foraging theory has found wide acceptance in the field. According to Pirolli [2004; 2005], the information foraging theory has already become a design concept in web site design and usability. The detractors are few and far between. There is, for instance, discussion about how to computationally represent information scent [Blackmon et al., 2002], but the concept itself appears to have found wide acceptance and it has been used in researched widely in many contexts.

The research has already produced computational cognitive models based on the information foraging theory and especially information scent [Blackmon et al., 2002, Pirolli and Fu, 2003, Chi et al., 2001]. The future development promises to give us automated usability analysis tools that simulate actual users, and help us improve the informationscape in many ways. Besides helping us better evaluate and design web sites, they bear a promise of assisting us in making better information browsers and perhaps even foraging tools that adapt to our foraging behavior and help us in optimizing it to the information environment. [Chi et al., 2001; Pirolli, 2004; 2005]

The information foraging theory appears to be here to stay. Its continued research promises to bring us better understanding not only concerning our information foraging behavior but also into the functioning of our cognitive system by modeling it better and better.

The information foraging theory shows the power of modern multi-scientific research. The psychological theory has borrowed from evolutionary and anthropological theories to formulate a theory that is bringing advances in information sciences and computer sciences. The better we understand ourselves as human beings, the better we understand our world and how to develop science that serves the human kind as whole and as individuals.

References

- [ACT-R, 2002-2003] Welcome to ACT-R (2002-2003). <http://act-r.psy.cmu.edu/>. Accessed: Dec. 14, 2005.
- [Blackmon et al., 2002] Marilyn Hughes Blackmon, Peter G. Polson, Muneo Kitajima and Clayton Lewis, Design Methods: Cognitive walkthrough for the web. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves* (2002), ACM Press, 463-470.
- [Card et al., 2001] Stuart K. Card, Peter Pirolli, Mija Van Der Wege, Julie B. Morrison, Robert W. Reeder, Pamela K. Schraedley and Jenea Boshart, Information scent as a driver of Web behavior graphs: results of a protocol analysis method for Web usability. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2001), ACM Press, 498-505.
- [Chen et al., 2002] Chaomei Chen, Timothy Cribbin, Jasna Kuljis and Robert MacrRedie, Footprints of information foragers: behaviour semantics of visual exploration. *International Journal of Human-Computer Studies* 57, 2 (Aug. 2002), 139-163.
- [Chi et al., 2000] Ed H. Chi, Peter Pirolli and James Pitkow, The scent of a site: a system for analyzing and predicting information scent, usage, and usability of a Web site. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2000), ACM Press, 161-168.
- [Dennis and Taylor, 2005] Alan R. Dennis and Nolan J. Taylor, Information foraging on the web: The effects of "acceptable" Internet delays on multi-page information search behavior. Article in press.
- [Gailey, 2002] Jeannine Gailey, Writing for the Web: Create Appealing Online Content (Feb. 2002). <http://www.microsoft.com/office/previous/frontpage/columns/sbcolumn04.asp>. Accessed: Dec. 15, 2005.
- [Gattis, 2002] Lyn F. Gattis, Planning and information foraging theories and their value to the novice technical communicator. In: *Proceedings of the 20th Annual International Conference on Computer Documentation* (2002), ACM Press, 39-43.
- [Katz and Byrne, 2003] Michael A. Katz and Michael D. Byrne, Effects of scent and breadth on use of site-specific search on e-commerce Web sites. *ACM Transactions on Computer-Human Interaction* 10, 3 (Sept. 2003), 198-220.
- [Landauer, 1998] Thomas K. Landauer, Peter W. Foltz and Darrell Laham, Introduction to Latent Semantic Analysis. *Discourse Processes* 25 (1998), 259-284.

- [Lucas, 2000] Peter Lucas, Pervasive information access and the rise of human-information interaction. *CHI '00 extended abstracts on Human factors in Computing Systems* (April 2000).
- [Nielsen, 1997] Jakob Nielsen, Jakob Nielsen's Alertbox: How Users Read on the Web (Oct. 1997). <http://www.useit.com/alertbox/9710a.html>. Accessed: Dec. 15, 2005.
- [Nielsen, 2003] Jakob Nielsen, Jakob Nielsen's Alertbox: Information Foraging: Why Google Makes People Leave Your Site Faster. (June 2003) <http://www.useit.com/alertbox/20030630.html>. Accessed: Dec. 13, 2005.
- [Norman, 1999] Donald A. Norman, Commentary: Banner Blindness, Human Cognition and Web Design (March 1999) <http://www.internettg.org/newsletter/mar99/commentary.html>. Accessed: Dec. 13, 2005.
- [Olston and Chi, 2003] Christopher Olston and Ed H. Chi, ScentTrails: Integrating browsing and searching on the Web. *ACM Transactions on Computer-Human Interaction* **10**, 3 (Sept. 2003), 177-197.
- [PARC, 2002] User Interface Research @ PARC: SNIF-ACT Home (2002) <http://www2.parc.com/istl/projects/uir/projects/snif-act/>. Accessed: Dec. 14, 2005.
- [Pirolli, 2004] Peter L. Pirolli, The use of proximal information scent to forage for distal content on the World Wide Web. To appear in Alex Kirlik (ed.), *Working with Technology in Mind: Brunswikian Resources for Cognitive Science and Engineering*. To be published.
- [Pirolli, 2005] Peter L. Pirolli, Rational analyses of information foraging on the web. *Cognitive Science* **29** (2005), 343-373.
- [Pirolli and Card, 1999] Peter L. Pirolli and Stuart K. Card, Information foraging. *Psychological Review* **106** (1999), 643-675.
- [Pirolli and Fu, 2003] Peter L. Pirolli and Wai-Tat Fu, SNIF-ACT: A model of information foraging on the World Wide Web. In: *Ninth International Conference on User Modeling* (2003), 45-54.
- [Pirolli et al., 1996] Peter Pirolli, James Pitkow and Ramana Rao, Silk from a sow's ear: extracting usable structures from the web. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1996), ACM Press, 118-125.
- [Ritter and Shiskowski, 2003] Frank E. Ritter and Natalie Shiskowski, ACT-R: Frequently Asked Questions List: What does ACT-R stand for? (July, 2003) <http://acs.ist.psu.edu/projects/act-r-faq/act-r-faq.html#G3>. Accessed: Dec. 14, 2005.

[Spool, 2004] Jared M. Spool, The Right Trigger Words (Nov. 2004)
http://www.uie.com/articles/trigger_words/. Accessed: Dec. 20, 2005.

L-systeemit: Kasvien mallintaminen

Juha Majanen

Tiivistelmä

L-systeemit ovat uudelleenkirjoitusjärjestelmään perustuva formaali kielioppi, jossa voidaan toteuttaa useita korvaavuussääntöjä rinnakkain. L-systeemeillä voidaan mallintaa mm. kasvien kehitystä, puiden kasvua ja generoida fraktaalikuvioita. Tässä tutkielmassa tarkastellaan ensin L-systeemejä yleisellä tasolla ja sen jälkeen keskitytään kasvien mallintamiseen soveltuviin L-systeemeihin ja niiden laajennuksiin.

Avainsanat: L-systeemit, kasvien mallintaminen

CR-luokka: F.4.3, I.3.3, I.6.5

1. Johdanto

L-systeemien keksijänä pidetään biologi Aristid Lindemeyeriä, joka kehitti L-systeemit alun perin mallintamaan soluja, kasveja ja niiden kehitystä. Pohjan L-systeemit saivat Chomskyn kielioppeista. L-systeemien perusajatuksena on se, että lähdetään liikkeelle tietystä aksioomasta, ja sovelletaan sen kaikkiin merkkeihin rinnakkain L-systeemin sääntöjä ja toistetaan tätä jokaisella iteraatiokierroksella määriteltyjen sääntöjen mukaisesti. Säännöissä määritellään korvaavat merkkijonot. L-systeemien ero muihin samantyyppisiin kielioppeihin on se, että sääntöjä toteutetaan rinnakkain. Tämän takia L-systeemit ovat erinomainen väline kasvien mallinnuksessa, koska rinnakkaisuuden periaatteella kasviin kohdistuvat muutokset saadaan tapahtumaan joka puolelle kasvia [Koli, 2001].

Käsittelen seuraavissa luvuissa erityyppisiä l-systeemejä ja sitä kuinka ne soveltuvat kasvien mallinnukseen.

2. L-systeemit

2.1 D0L-systeemit

Yksinkertaisin L-systeemiluokka on D0L-systeemit, jotka ovat deterministisiä ja kontekstittomia. Tämä tarkoittaa sitä, että jokaisella merkillä on yksikäsitteinen

korvaavuussääntö. Perusideana on se, että asetetaan jokin korvaavuussääntö jollekin tietylle merkille esim. $a \rightarrow ab$, joka tarkoittaa sitä, että merkki a korvataan merkkijonolla ab . Asetetaan sääntö $b \rightarrow a$, tällöin merkki b korvataan merkillä a . Uudelleenkirjoitus alkaa jonkun tietyn aksiooman perusteella. Otetaan tässä tapauksessa aksioomaksi b . L-systeemin johdot ovat muotoa $b \rightarrow a \rightarrow ab \rightarrow aba \rightarrow abaab \rightarrow abaababa \rightarrow \dots$

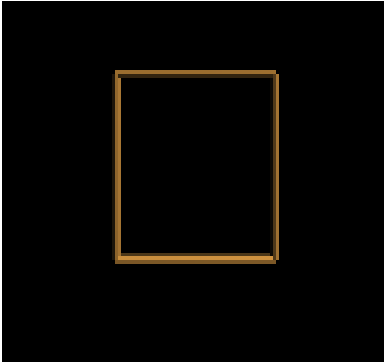
L-systeemien piirtämiseen on määritelty tietyt säännöt. Piirtämiseen käytetään eräänlaista kilpikonaa (Turtle interpretation of strings), jota voidaan verrata LOGO-tyyliseen kilpikonnaan [Prusinkiewicz and Lindenmayer, 1990]. Mikko Kolin [2001] tutkielmassa tämä on käännetty piirtopääksi, jota käytän myös tässä tutkielmassa.

Ideana on se, että piirtopään tila esitetään kolmikon (x, y, α) avulla, missä koordinaatit (x, y) kuvaavat piirtopään sijaintia ja α kulmaa, johon piirtopää osoittaa. Kun piirtopäälle annetaan askelpituus d ja kulman inkrementti δ , piirtopäälle voidaan antaa seuraavanlaisia komentoja [Prusinkiewicz and Lindenmayer, 1990]:

- F Liikutetaan piirtopäätä eteenpäin askelpituuden verran. Viiva pisteiden (x, y) välillä piirretään
- f Liikutetaan piirtopäätä eteenpäin askelpituuden verran piirtämättä mitään.
- + Käännyttään vasemmalle kulman δ verran. Piirtopään seuraava tila on $(x, y, \alpha + \delta)$.
- Käännyttään oikealle kulman δ verran. Piirtopään seuraava tila on $(x, y, \alpha - \delta)$.

Havainnollistan seuraavalla esimerkillä piirtopään käyttöä. Apuna käytän Drawing real trees -ohjelmaa, joka on toteutettu viime vuoden Projektityö -kurssilla.

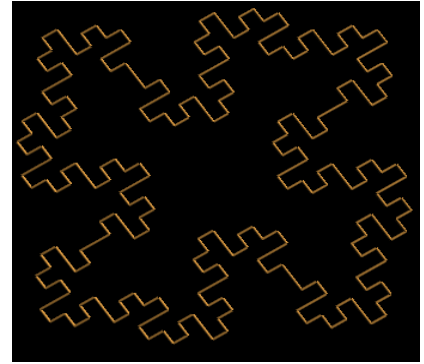
Otetaan esimerkiksi *Kochin saari*. Sen aksiooma on $F-F-F-F$ ja ainoa sääntö $F \rightarrow F-F+FF-F-F+F$. Kulman inkrementti on 90° . Kuvissa 1-3 on esitelty L-systeemin määräämät kuviot, kun sääntöä on sovellettu 0, 1 ja 2 kertaa.



Kuva 1.



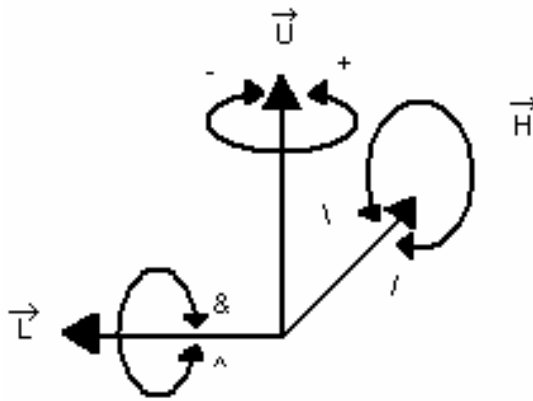
Kuva 2.



Kuva 3.

2.2 Kolmiulotteiset L-systeemit

L-systeemien piirtopään tulkinta on laajennettavissa kolmiulotteiseksi vektoreiden avulla. Kuvataan piirtopään suuntaus kolmella vektorilla H , L , U , osoittaen piirtopään suuntaa, suuntaamista vasemmalle ja suuntaamista ylös. Näillä vektoreilla on yksikköpituus, ne ovat suorakulmassa toisiinsa nähden ja ne toteuttavat yhtälön $H \times L = U$. Piirtopään rotaatiota voidaan kuvata yhtälöllä $[H' L' U'] = [H L U] \mathbf{R}$, jossa \mathbf{R} on (3×3) -rotaatiomatriisi [Prusinkiewicz and Lindenmayer, 1990]. Kuva 4 havainnollistaa vektoreiden merkitystä.



Kuva 4.

Edellisessä esimerkissä piirtopään ohjaus on kuvattu vektoreilla. Erona kolmiulotteisessa mallintamisessa kaksiulotteiseen mallintamiseen on se, että kolmiulotteisessa mallintamisessa otetaan myös kierto-liike avuksi. Seuraavaksi esittelen symbolit piirtopään ohjaamiseen hieman yksinkertaisemmin X-, Y-, ja Z -akseleilla. Periaate on kuitenkin sama kuin edellisessä esimerkissä.

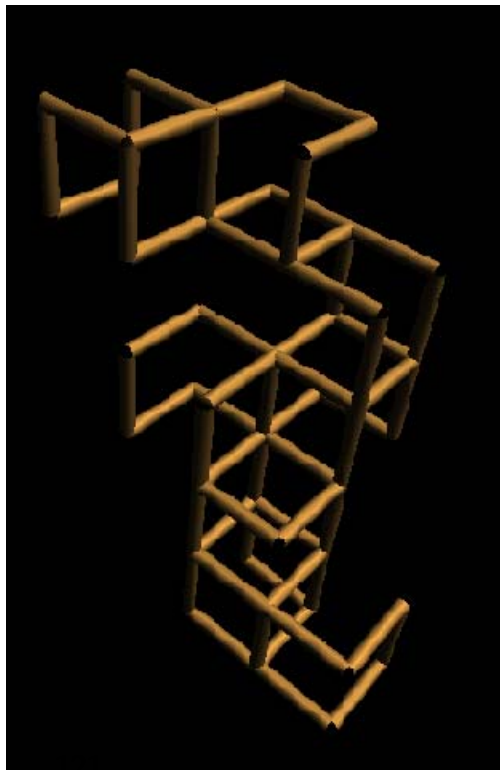
Seuraavia symboleita voidaan käyttää piirtopään ohjauksessa:

- + Kääntää vasemmalle Y akselin ympäri kulman δ verran
- Kääntää oikealle Y akselin ympäri kulman δ verran
- & Kääntää ylös X akselin ympäri kulman δ verran
- ^ Kääntää alas X akselin ympäri kulman δ verran
- \ pyörittää vasemmalle Z akselin ympäri kulman δ verran
- / Pyörittää oikealle Z akselin ympäri kulman δ verran
- | Kääntyy Y akselin ympäri 180°

Tarkastella seuraavaksi L-systeemiä, jonka aksiooma on F ja ainoa sääntö $F \rightarrow F + F \setminus F - \setminus F / F \wedge F \setminus F \setminus F$. Kuvissa 5 ja 6 on esitelty L-systeemin määräämät kuviot, kun sääntöä on sovellettu kerran ja kahdesti.



Kuva 5.



Kuva 6.

2.3 Haarautumisrakenteet

Haarautumisrakenteet tai toiselta nimeltään pinorakenteet ovat avainasioita kasvien mallintamisessa. Aiemmin esiteltyissä malleilla saadaan piirrettyä hienoja viivakuvioita, mutta kasvien mallintamiseen se ei riitä. Kasvien mallintamisessa keskeisiä elementtejä ovat esimerkiksi oksat, varret, lehdet jne., haarautumisrakenteiden avulla näiden mallintaminen on mahdollista. Periaatteena on se, että piirtopään tila (suuntaus) tallennetaan pinoon, josta se voidaan myöhemmin palauttaa [Prusinkiewicz and Lindenmayer, 1990].

Esitellään kaksi uutta symbolia haarojen määrittämiseen. Piirtopää tulkitsee ne seuraavalla tavalla:

- [Tallentaa piirtopään sen hetkisen tilan pinoon. Pinoon tallennettu informaatio sisältää piirtopään sijainnin ja suuntauksen, ja mahdolliset muut ominaisuudet kuten värin ja piirrettävän viivan leveyden.
-] Palauttaa tilan pinosta ja tekee siitä piirtopään nykyisen tilan. Mitään viivaa ei piirretä, vaikka yleisesti piirtopään sijainti muuttuu.

Otetaan muutama esimerkki haarautumisrakenteiden piirtämisestä kaksiulotteisesti:

Iteraatiot: 7

$\delta=7$

Aksiooma: X

Sääntö 1: $X \rightarrow F[+X][-X]FX$

Sääntö 2: $F \rightarrow FF$

Iteraatiot: 4

$\delta=22,5$

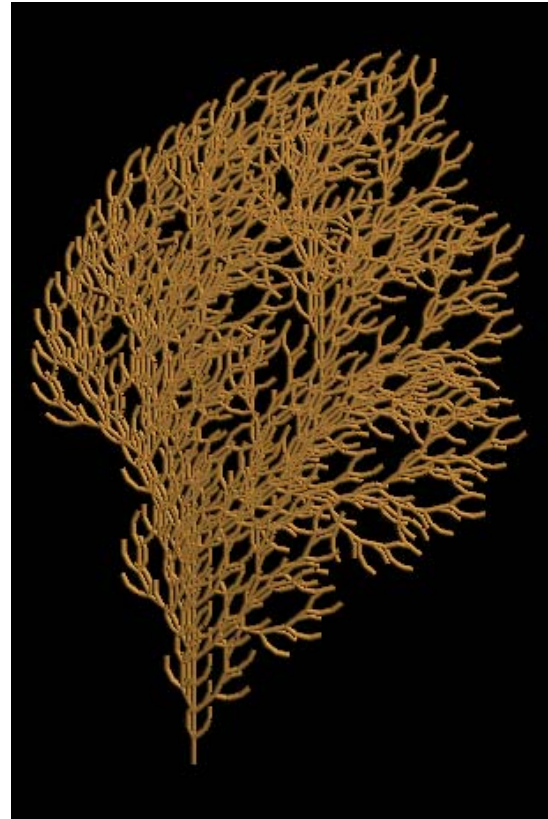
Aksiooma: F

Sääntö 1: $F \rightarrow FF[-F+F+F][+F-F-F]$

Näiden arvojen määräämät kuviot on esitetty kuvissa 7 ja 8.



Kuva 7.



Kuva 8.

Esitellään vielä muutama uusi symboli kolmiulotteisen kuvan piirtämistä varten. Otetaan malliin mukaan myös lehti.

{	Aloitetaan lehden piirtäminen
}	Lopetetaan lehden piirtäminen
f	Tehdään askel ilman piirtoa
! ja '	Näitä merkkejä käytetään vähentämään segmenttien läpimittaa ja lisäämään nykyinen indeksi värikartalle.

Tarkastellaan esimerkkinä kolmiulotteista kasvimallia, jonka aksiooma on A ja jossa on seuraavat neljä sääntöä:

Sääntö 1: A -> [&FLA]/////[&FLA]////////[&FLA]

Sääntö 2: F -> !S/////F

Sääntö 3: S -> FL

Sääntö 4 (lehti): L -> [""^^{-f+f+f-|-f+f+f}]

Soveltamalla sääntöjä seitsemän kertaa, saadaan kuvan 9 kuvio.



Kuva 9.

2.4 Stokastiset L-systeemit

Kaikki samalla deterministisellä L-systeemillä luodut kasvimallit ovat identtisiä. Yhdistelemällä näitä rinnakkain samaan kuvaan ei saada aikaiseksi kovin realistisen näköistä kasvimaata tai kukka-asetelmaa. Kasvimalleihin saadaan vaihtelua stokastisilla L-systeemeillä. Kasvin perusominaisuudet pysyvät samana, mutta yksityiskohdat muuttuvat sattumanvaraisesti. Muutos voidaan toteuttaa satunnaistamalla piirtopään liikkeitä, L-systeemi tai molemmat [Prusinkiewicz and Lindenmeyer, 1990].

Satunnaisuus luodaan siten, että L-systeemin sääntöjä toteutetaan jollain tietyllä todennäköisyydellä niin, että todennäköisyyksien summa on 1.

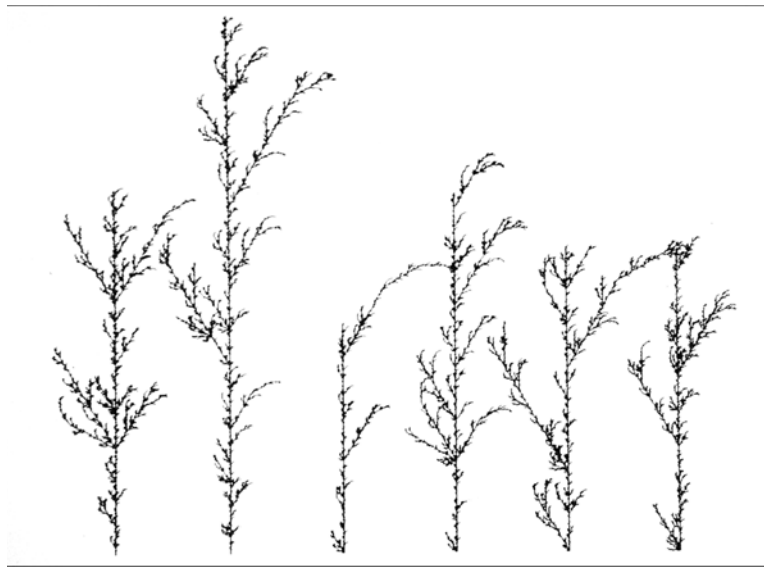
Esimerkkinä on L-systeemi, jonka aksioma on F ja jonka kolmella F-säännöllä on sama todennäköisyys:

Sääntö 1: $F \rightarrow (0.33)F[+F]F[-F]F$

Sääntö 2: $F \rightarrow (0.33)F[+F]F$

Sääntö 3: $F \rightarrow (0.34)F[-F]F$

Drawing real trees -ohjelmassa ei ole mahdollista ottaa huomioon satunnaisuutta, joten kuva 10 on suoraan Prusinkiewitzin ja Lindenmeyerin [1990] kirjasta.



Kuva 10.

2.5 Kontekstiset L-systeemit

Kontekstiset L-systeemit ovat harppaus astetta syvällisempään kasvien mallinnukseen. Pyrittäessä mahdollisimman realistiseen kasvin mallintamiseen, tulee jokaisessa piirtovaiheessa ottaa huomioon, mitä on tehty aikaisemmin ja missä suhteessa seuraava piirto on edelliseen, toisin sanoen seuraava piirto riippuu edeltäjän kontekstista. Tämä mallintamistapa on käytännöllinen kun halutaan simuloida esim. kasvissa tapahtuvaa ravinteiden kulkua [Prusinkiewitz and Lindenmayer, 1990].

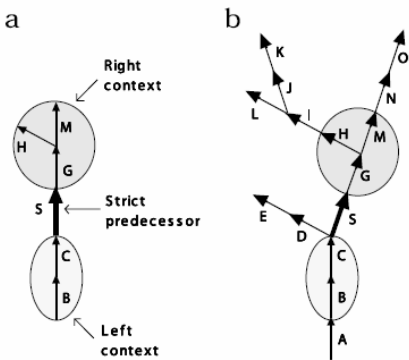
Kontekstisia L-systeemeitä on kahta eri tyyppiä: 2L-systeemit ja 1L-systeemit. 2L-systeemeissä säännöillä on oikea ja vasen konteksti, kun taas 1L-systeemin säännöillä on vain toinen näistä. 2L-systeemien säännöt ovat muotoa $a_l < a > a_r \rightarrow \chi$, jossa kirjain a

(strict predecessor) voi toteuttaa sanan χ jos ja vain jos a_i ja a :ta edeltää a_i ja a :ta seuraa a_r . 1L-systeemin säännöt ovat muotoa $a_i < a \rightarrow \chi$ tai $a > a_r \rightarrow \chi$ [Koli, 2001].

Olkoon 1L-systeemin aksiooma baaaaaaa ja säännöt $b < a \rightarrow b$ ja $b \rightarrow a$. Saadaan johto, jossa aksioomassa oleva b-merkki liikkuu oikealle:

baaaaaaa \rightarrow abaaaaaa \rightarrow aabaaaaa \rightarrow aaabaaaa \rightarrow aaaabaaa ...

Kuvassa 11 on esimerkki kontekstisen puun mallintamisesta, jossa kohdassa (a) oleva edeltäjä (strict predecessor) sopii särmään S kohdassa (b). Käytetään vasemmasta kontekstista nimitystä l , ja oikeasta kontekstista nimitystä r . Kohdassa (a) oleva polku sopii kohtaan (b), jos l päättyy S :n alkusolmuun ja r on alipuu kohdassa (b), jos se alkaa S :n loppusolmusta. Tällöin voidaan korvata särmä S puurakenteella, joka on määritelty ehdon seuraajana [Prusinkiewicz and Lindenmayer, 1990].



Kuva 11.

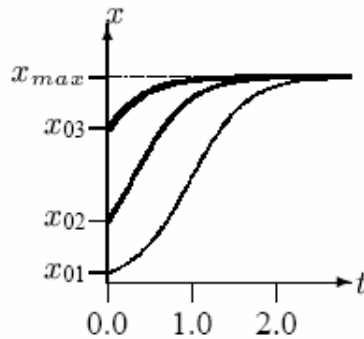
2.6 Kasvun kuvaaminen

Tietokoneella tehdyillä kasvimalleilla on useita etuja. Ne voivat tarjota kvantitatiivisen ymmärtämyksen kehitysmekanismeista, silloin kun kvalitatiiviset selitykset eivät pohjimmiltaan riitä esim. kasvun dynamiikka. Mallit voivat johtaa synteettiseen ymmärtämykseen kasvin kehitykseen vaikuttavien seikkojen vuorovaikutuksesta kuten geneettisistä säännöistä, fyysisistä prosesseista ja ympäristön vaikutuksista kasvin kehityksessä. Mallien avulla voidaan löytää kasvin kehityksestä myös alueita, joista ei ole aikaisemmin tiedetty [Prusinkiewicz, 2004].

Kirjallisuudessa on esitelty monenlaisia kasvuun liittyviä funktioita. Niille ominaista on, että alussa arvoa kasvatetaan hitaasti, kiihdytetään vauhtia ja lopuksi tasoitetaan

vauhti kun päästään lähelle maksimiarvoa tai saavutetaan maksimiarvo (*sigmoidal functions*) [Prusinkiewicz, 1998].

Esimerkkinä voidaan mainita Velhurstin looginen yhtälö $\frac{dx}{dt} = r \left(1 - \frac{x}{x_{max}}\right) x$, jota on havainnollistettu kuvassa 12, kun r saa arvon 3.0 [Prusinkiewicz *et al.*, 1993].



Kuva 12.

Kasvun realistinen mallintaminen on monien yhtälöiden summa, koska siihen vaikuttavat mm. ympäristötekijät, auringonvalo, ravinteet, juuriston kasvu jne.. Syvällisempää tietoa janoaville suosittelem Calgaryn yliopiston tietojenkäsittelytieteiden laitoksen sivustoa, josta löytyy aiheeseen liittyen useita julkaisuja.

3. Yhteenveto

L-systeemit ovat erinomainen väline kasvien mallinnukseen. Mallinnustapa näyttää aluksi suhteellisen yksinkertaiselta lyhyine merkkijonoineen, mutta iteraatiokierroksien lisääntyessä merkkijonot saattavat kasvaa todella monimutkaisiksi. Tässä piileekin L-systeemien etu muihin mallinnustapoihin verrattuna. L-systeemien avulla voi helposti iteraatioita lisäämällä ja parametreja muuttamalla saada aikaan todella näyttäviä ja aidon näköisiä malleja hyvinkin yksinkertaisista aksioomista ja säännöistä. Tietylnainen toistuvuus malleissa haittaa hieman, mutta esim. stokastisilla L-systeemeillä saadaan aikaan tarvittavaa vaihtuvuutta. Eri asia on, kuinka tietokoneella luotu satunnaisuus vastaa luonnossa esiintyvää todellisuutta. Mallien avulla pystytään myös oppimaan uutta mm. kasvien kehityksestä ja ympäristötekijöiden vaikutuksesta.

Tässä tutkielmassa kasvien mallit ovat melko vaatimattomia ja antavatkin vain pienen yleissilmäyksen siitä, mitä L-systeemeillä saadaan mallinnettua. Kasvien realistiseen mallintamiseen täytyy kuitenkin ottaa huomioon monia muitakin elementtejä kuin pelkästään varret ja lehdet. L-systeemien monipuolinen hallinta vaatiikin syvällistä perehtymistä L-systeemien matemaattisiin ominaisuuksiin ja myös muihin laajennuksiin, joita en käsitellyt tässä tutkielmassa. Mallintamisesta kiinnostuneille suosittelenkin luettavaksi Prusinkiewiczin ja Lindenmeyerin teosta *Algorithmic Beauty of Plants* [1990].

Viiteluettelo

- [Koli, 2001] Mikko Koli, *L-systeemeistä*, Tampereen yliopisto, Tietojenkäsittelytieteen laitos, Pro gradu -tutkielma, 2001.
- [Lindenmayer and Prusinkiewicz, 1990] Aristid Lindenmayer and Przemyslaw Prusinkiewicz, *The Algorithmic Beauty of Plants*, Springer-Verlag, New York, 1990.
- [Prusinkiewicz, 2004] Przemyslaw Prusinkiewicz, Modeling plant growth and development. *Current Opinion in Plant Biology* **7**, 1 (2004), 79-83.
- [Prusinkiewicz *et. al.*, 1993] Przemyslaw Prusinkiewicz, Mark S. Hammel and Eric Mjolsness, Animation of plant development. In: *Proceedings of the 20th annual Conference on Computer Graphics and Interactive Techniques*, 351-360.
- [Prusinkiewicz, 1998] Przemyslaw Prusinkiewicz, Modeling of spatial structure and development of plants: a review. *Scientia Horticulturae* **74**, 1-2 (1998), 113-149.

Muita lähteitä:

<http://algorithmicbotany.org> – Calgaryn yliopiston tietojenkäsittelytieteiden laitoksen sivusto, josta löytyy paljon mielenkiintoista materiaalia aiheeseen liittyen.

Käsinkosketeltavien käyttöliittymien luokittelusta

Antti Nyman

Tiivistelmä.

Käsinkosketeltavat käyttöliittymät ovat melko uusi tutkimussuuntaus, johon liittyvät määritelmät ja käsitteet eivät ole vielä vakiintuneet. Kehittyvä ala tarvitsee myös luokitteluja ja teorioita, jotka helpottavat tutkimusalueen määrittelyä sekä sovellusten vertailua ja suunnittelua. Tässä tutkielmassa esittelen tutkimussuuntaukseen liittyviä termejä, määritelmiä, teorioita ja luokitteluja. Jaoin luokittelut kahteen ryhmään sen mukaan, perustuvatko ne objektien merkitysten vai suhteiden analysointiin. Samalla havaitsin, että kumpaankin ryhmään kuuluvissa luokitteluissa on omat puutteensa ja uusille luokitteluille on vielä tarvetta.

Avainsanat ja -sanonnat: Käsinkosketeltava käyttöliittymä, luokittelu.

CR-luokat: H.5.2

1. Johdanto

Visio kaikkialla läsnäolevasta tietotekniikasta, joka katoaa taustalle käyttäjän suorittaessa jokapäiväisiä tehtäviään [Weiser, 1991], on inspiroinut laajasti tutkimusta ihmisen ja tietokoneen vuorovaikutuksessa. *Jokapaikan tietotekniikassa* (ubiquitous computing) tietokoneet häviävät käytöstä sellaisina kuin olemme ne aikaisemmin tunteneet ja sulautuvat osaksi ympäristöämme. Tällöin voimme käyttää niitä työkaluina suorittaessamme jokapäiväisiä tehtäviämme, ilman että ne vaativat erityistä huomiotaamme ja irrottavat keskittymisemme varsinaisesta tehtävästä. Samalla koko nykyinen käsityksemme tietokoneenkäyttötaidosta voidaan unohtaa. Vaikka vision tärkeimpänä viestinä pidetään monesti verkottuneiden tietokoneiden leviämistä kaikkialle, oli siinä heti alusta alkaen tärkeää myös tietokoneiden mahdollistaman laskennan yhdistyminen huomaamattomaksi osaksi jokapäiväistä aineellista ympäristöämme.

Eräs aikaisimmista ja vaikuttavimmista yrityksistä yhdistää jokapäiväisissä toimissa käyttämämme esineet tietotekniikan kanssa on DigitalDesk-ympäristö [Wellner, 1993], joka on hyvä esimerkki *lisätyn todellisuuden* (augmented reality) sovelluksesta. Se mahdollistaa käyttäjälle totutun vuorovaikutuksen paperin kanssa, samalla kuitenkin helpottaen paperilla esitetyn informaation laskennallista käsittelyä. Videokameran ja -projektorin avulla toteutetulla työpöydällä käyttäjä pystyy käsittelemään aineellisia papereita ja samalla hyödyntämään digitaalisten dokumenttien hyviä puolia, kuten esimerkiksi kopiointia ja liittä-

mistä. Jo tässä järjestelmässä käyttäjän ja tietokoneen välinen vuorovaikutus perustui aineellisiin esineisiin, vaikka siitä suuri osa tapahtuikin eleiden avulla.

Aikaisempaa kiinteämmän liitoksen aineellisten esineiden ja digitaalisen informaation välille toteuttivat *tartuttavat käyttöliittymät* (graspable user interface) [Fitzmaurice *et al.*, 1995], jotka mahdollistavat digitaalisen datan muokkaamisen aineellisten objektien avulla. Ne toivat käyttöliittymäkomponentit näytöltä fyysisesti kosketeltaviksi. Esitetyssä sovelluksessa käyttäjä pystyy asettamaan tasolle heijastettujen piirustusohjelman elementtien päälle aineellisia palikoita ja näitä liikuttamalla muuttamaan graafisten elementtien ominaisuuksia, kuten esimerkiksi sijaintia, suuntausta tai kokoa. Aineellisten objektien käyttämisellä syötelaitteina on useita etuja graafisiin käyttöliittymiin (graphical user interface) verrattuna. Koska syötelaitteet voivat olla sovelluskohtaisia, voidaan ne suunnitella tarkasti johonkin tiettyyn tehtävään. Kukin syötelaite voidaan myös niin halutessa sitoa ainoastaan yhteen toimintoon, jolloin eri toimintoja voidaan käsitellä ajallisesti rinnakkain (space-multiplexing) [Fitzmaurice *et al.*, 1995]. Useat syötelaitteet mahdollistavat kummankin käden samanaikaisen käytön ja tukevat monen käyttäjän yhteistyötä. Lisäksi käyttäjät voivat paremmin hyödyntää esineiden käsittelyn ja avaruudellisen päättelyn taitojaan.

Tartuttavat käyttöliittymät ovat vaikuttaneet vahvasti myöhempään tutkimukseen digitaalisen informaation ja aineellisten objektien yhdistämisessä. Inspiraatiota siitä on saanut myös visio *käsinkosketeltavista käyttöliittymistä* (tangible user interface) [Ishii and Ullmer, 1997], joka antoi nimen tutkimussuuntaukselle. Esitettyä visiota laajasti tulkiten käsinkosketeltaviksi käyttöliittymiksi voidaan käsittää kaikki käyttöliittymät, joissa käyttäjä käsittelee aineellisia objekteja muuttaakseen niihin sidottua digitaalista informaatiota. Kuten jatkossa käy ilmi, ei täyttä yksimielisyyttä käsinkosketeltavien käyttöliittymien määritelmästä tai peruselementeistä ole kuitenkaan vielä saavutettu.

Tässä tutkielmassa käsittelen käsinkosketeltaville käyttöliittymille ehdotettuja termejä, määritelmiä, teorioita ja luokitteluja. Seuraavassa luvussa esittelen tutkimussuuntauksen termejä ja määritelmiä. Kolmannessa luvussa kuvaan erilaisia teorioita ja luokitteluja sekä vertailen niitä. Lopuksi esitän yhteenvedon.

2. Mitä käsinkosketeltavat käyttöliittymät ovat?

Käsinkosketeltavat käyttöliittymät ovat monille vielä tuntemattomia. Tässä luvussa yritän selventää niihin liittyvää käsitteistöä sekä niiden tärkeimpiä periaatteita ja yleisimpiä käyttötarkoituksia. Vertailen aluksi käsinkosketeltavista käyttöliittymistä käytettyjä termejä, jonka jälkeen käsittelen niille esitettyjä

määritelmiä. Lopuksi esittelen käsinkosketeltavien käyttöliittymien yleisimpiä sovellusalueita.

2.1. Terminologiaa

Käyttämäni suomennos, "käsinkosketeltava käyttöliittymä", perustuu englanninkieliseen termiin "tangible user interface", jonka olen valinnut perustaksi suomennokselle sen saavuttaman laajan käytön takia. Englanninkielisessä termissä käytettävä sana "tangible" puolestaan on saanut alkunsa latinankielisistä sanoista "tangibilis" ja "tangere", jotka tarkoittavat suomen kielessä kutakuinkin "jota voi koskea" ja "koskea".

Käsinkosketeltavista käyttöliittymistä on englanninkielisissä julkaisuissa käytetty vaihtelevasti termejä, joissa sana "tangible" on korvattu jollakin muulla (esimerkiksi "embodied", "graspable", "manipulative" tai "physical"). Vaikka hienoisia painotuseroja voidaan termeissä havaita, tarkoitetaan niillä pääosin samaa käsitettä.

2.2. Määritelmiä

Tartuttavat käyttöliittymät määriteltiin alun perin sellaisiksi käyttöliittymiksi, joissa osa käyttöliittymän normaalisti virtuaalisista elementeistä saa aineellisen olomuodon [Fitzmaurice *et al.*, 1995]. Tämä määritelmä painottaa eroavuutta graafisiin käyttöliittymiin, sillä sen mukaan tartuttavat esineet tulevat aikaisempien syötelaitteiden korvaajiksi ja mahdollistavat joidenkin graafisen käyttöliittymän elementtien suoran käsittelemisen. Määritelmä ei kuitenkaan ota kantaa siihen, millaisia tulostuskanavia tartuttavien käyttöliittymien tulisi käyttää, ja käyttöliittymä saattaa uusista syötelaitteista huolimatta muistuttaa muilta osin graafisia käyttöliittymiä.

Suuressa osassa uusista tämän tutkimussuuntauksen julkaisuista aihepiirin määrittelyssä viitataan Ishiin ja Ullmerin [1997] Tangible Bits -visioon, joka tähtää virtuaalisen informaation tekemiseen käsinkosketeltavaksi ja jonka taustalla vaikuttavat ihmisten luonnolliset kyvyt käsitellä esineitä ja asioita. Ishii ja Ullmer määrittelivät käsinkosketeltavat käyttöliittymät alun perin siten, että ne antavat käyttäjälle mahdollisuuden käsitellä digitaalista informaatiota tarttumalla informaatioon liitettyihin aineellisiin objekteihin ja käsittelemällä niitä. Myöhemmin Ullmer ja Ishii [2000] ovat lisänneet määritelmään vaatimuksen siitä, että käsinkosketeltavan käyttöliittymän ei tulisi tehdä eroa syöte- ja tulostuslaitteiden välille. Nykyisen määritelmän mukaan vuorovaikutuksen tulisi siis perustua aineellisiin, kosketeltavissa oleviin esineisiin ja asioihin, jotka mahdollistavat digitaalisen informaation esittämisen ja muokkaamisen. Käsinkosketeltavissa käyttöliittymissä käytettävien aineellisten objektien tulisi olla yhtäaikaaisesti sekä esitys, joka osittain vastaa virtuaalista informaatiota,

että ohjain, jonka avulla informaatiota voidaan muokata. Tällöin niissä yhdistyy sekä syöte- että tulostuskanavan ominaisuuksia.

Käsinkosketeltavilla käyttöliittymillä on määritelmän mukaan neljä ominaispiirrettä [Ullmer and Ishii, 2000]:

- aineellinen esitys on sidottu tietokoneen avulla digitaaliseen informaatioon,
- aineellinen esitys mahdollistaa vuorovaikutteisen kontrolloinnin,
- aineellinen esitys on sidottu havaittavasti aineettomaan esitykseen ja
- aineellisten objektien tila ilmentää osaltaan järjestelmän virtuaalista tilaa.

Esitettyjen piirteiden perusteella käyttöliittymään kuuluviin aineellisiin objekteihin tulee olla liitettynä jotakin digitaalista informaatiota tai laskennallisia malleja, joiden tilaa järjestelmään kuuluvat aineelliset objektit osaltaan ilmentävät. Vaikka aineelliset objektit muodostavatkin osan esityksestä, liittyy siihen usein myös muita tulosteita (kuten grafiikkaa tai ääntä), joiden tulee liittyä havaittavasti aineellisiin objekteihin. Aineelliset objektit eivät myöskään ole olemassa pelkkää informaation esittämistä varten, vaan toimivat järjestelmän ensisijaisena ohjaimena.

Vielä Ishiin ja Ullmerin määritelmää laajemman määritelmän käsinkosketeltaville käyttöliittymille esittää Fishkin [2004], jonka määritelmä ei ota yhtä tarkasti kantaa siihen, kuinka aineelliset objektit esittävät digitaalista informaatiota ja mahdollistavat sen kontrolloinnin. Tämä määritelmä kuitenkin kattaa myös monia sellaisia käyttöliittymiä, jotka aiemman määritelmän perusteella eivät olisi olleet luokiteltavissa käsinkosketeltaviksi. Määritelmän mukaan käsinkosketeltavan käyttöliittymän tulee toteuttaa ainoastaan seuraavat asiat [Fishkin, 2004]:

- käyttäjä saa aikaan syötetapahtuman käsittelemällä aineellisia objekteja,
- tietokonejärjestelmä havaitsee syöteen ja muuttaa tilaansa ja
- järjestelmä laukaisee tulostustapahtuman, joka näkyy käyttäjälle jonkin aineellisen objektin ominaisuuden muutoksena (esimerkiksi objektin muoto, ääni, grafiikka näyttöpinnassa tai tuntopalaute).

Edellä mainitut kolme piirrettä vaativat järjestelmältä kaikessa yksinkertaisuudessaan vain sen, että siihen kuuluvia aineellisia objekteja käsittelemällä on mahdollista saada aikaan tietokoneellisesti ohjattua palautetta, joka liittyy aineellisiin objekteihin. Määritelmä on hyvin laaja ja siihen liittyykin myöhemmin esiteltävä luokittelu, jonka avulla järjestelmän käsinkosketeltavuuden taso on mahdollista määrittää.

2.3. Sovellusalueita

Vaikka käsinkosketeltavien käyttöliittymien on todettu soveltuvat monenlaisiin tehtäviin, on olemassa tiettyjä sovellusalueita, joilla ne ovat alkaneet yleistyä

muuta enemmän. Vaikka kattavaa luokittelua sovellusalueista ei välttämättä ole järkevää tehdä, on yleiskuvan saamiseksi hyödyllistä määrittää yleisimpiä käsinkosketeltavien käyttöliittymien käyttötarkoituksia. Seuraavat sovellusluokat on havaittu laajoiksi ja kehittyviksi [Ullmer and Ishii, 2001]:

- informaatiota säilyttävät, hakevat ja muokkaavat,
- informaatiota visualisoivat,
- järjestelmiä mallintavat tai simuloivat,
- järjestelmiä hallinnoivat tai konfiguroivat ja
- opetus-, viihde- tai ohjelmointisovellukset.

Sovellusalueuokkien sisälläkin järjestelmät voivat erota toisistaan merkittävästi ja painottua eri tavoin. Monilla edellä mainituilla sovellusalueilla käsinkosketeltavia käyttöliittymiä on toteutettu erityisesti tietokoneavusteisen ryhmätyön (computer supported collaborative work) tutkimuksen tueksi.

3. Käsinkosketeltavien käyttöliittymien luokitteluja

Hyvät teoriat ja luokittelut ovat tarpeellisia kaikille tutkimussuuntauksille, sillä ne auttavat tutkimusalueen määrittelemisessä, helpottavat erilaisten sovellusten vertailua ja ohjaavat suunnittelua. Ne myös auttavat näkemään sovellusten hyviä puolia sekä alueita, joilla on vielä käyttämättömiä mahdollisuuksia.

Tässä luvussa käsittelen teorioita ja luokitteluja, jotka helpottavat käsinkosketeltavien käyttöliittymien määrittelyä, ryhmittelyä, vertailua ja suunnittelua. Esittelen vaihtoehtoisia ja toisiaan täydentäviä luokitteluja sekä vertailen niitä keskenään. Tarkoitukseni on löytää luokitteluista ja teorioista eroja ja yhtymäkohtia sekä eri ratkaisuille ominaisia hyviä ja huonoja puolia. Olen ryhmitellyt luokittelut sen mukaan, perustuvatko ne aineellisten objektien merkityksiin vai niiden välisiin suhteisiin. Aloitan selvittämällä aihepiiriin liittyviä terminologisia ongelmia, jonka jälkeen esittelen eri luokittelut.

3.1. Terminologiaa

Jotta käyttöliittymään kuuluvat aineelliset objektit voidaan erottaa kaikista muista aineellisista objekteista, tarvitaan nimityksiä, joilla niihin voidaan helposti ja yksiselitteisesti viitata. "Aineellinen objekti" on jatkuvassa käytössä pitkä ja hankala, eikä tee eroa käyttöliittymään kuuluvien ja kuulumattomien objektien välille (sama ongelma pätee myös englanninkieliseen termiin "physical object"). "Aineellinen esine" on merkitykseltään samankaltainen ja siihen liittyvät samat heikkoudet. Lähes vastaava lyhempi termi on "artefakti", johon kuitenkin sisältyy vaatimus, että se on ihmisen tekemä. Tämä taas ei ole hyväksyttävää, sillä käsinkosketeltavissa käyttöliittymissä voidaan käyttää muitakin kuin ihmisten tekemiä objekteja [Ullmer and Ishii, 2001].

Englanninkielestä johdetun termin "phicon" ("physical icon") esittelivät Ishii ja Ullmer [1997], mutta ovat itsekin todenneet sen soveltumattomuuden kaikkia objekteja kuvaavaksi termiksi, sillä objektit eivät välttämättä ole merkitykseltään ikonisia, vaan voivat olla myös symbolisia [Ullmer and Ishii, 2000]. Myös termiä "pyfo" on ehdotettu käytettäväksi samaan tarkoitukseen [Calvillo-Gómez *et al.*, 2003; Shaer *et al.*, 2004], mutta ainakaan vielä se ei ole levinnyt laajempaan käyttöön. Ishii ja Ullmer toteavat käyttökelpoiseksi muuallakin käytetyn termin "tangible", mutta päätyvät ehdottamaan termin "token" käyttöä kuvaamaan käsinkosketeltaviin käyttöliittymiin kuuluvia aineellisia objekteja. Lisäksi he ehdottavat termiä "reference frame" (suomenkielessä "koordinaatisto") kuvaamaan aineellista tilaa, jossa vuorovaikutus tapahtuu.

Käsitteiden eroavaisuudet aiheuttavat myös jonkin verran hankaluuksia jäljempänä esiteltävien teorioiden tulkinnessa ja vertailussa. Englanninkielinen termi "token" tarkoittaa muiden merkitystensä ohella pientä osaa, joka edustaa kokonaista. Tätä painottaen Holmquist ja muut [1999] käyttävät sitä kuvaamaan aineellisia objekteja, jotka ilmentävät olemuksellaan ainakin jotakin virtuaalisen vastineensa piirrettä. He harkitsivat termin sijasta myös "phicon" termin käyttöä, mutta hylkäsivät sen, sillä sitä on aikaisemmissa julkaisuissa käytetty heidän käytöstään poikkeavalla tavalla. Ullmer ja Ishii [2000] puolestaan toteavat, että Holmquistin ja muiden "token" käsitettä vastaa heidän käsitteistössään juuri "phicon" ja heillä "token" on yläkäsite kaikille käyttöliittymään kuuluville aineellisille objekteille.

Ellei toisin mainita, tässä tutkielmassa termillä "aineellinen objekti" tarkoitetaan käyttöliittymään kuuluvia objekteja. Syynä tähän on edellä mainittu termien päällekkäisyys, joka saattaa aiheuttaa sekaannuksia. Kunkin luokittelun tai teorian yhteydessä käytetään suomennoksia siinä käytettävistä termeistä ja ne selitetään erikseen kunkin esittelyn kohdalla.

3.2. Objektien merkityksiin perustuvat luokittelut

Monissa tapauksissa käsinkosketeltavien käyttöliittymien analysointi on helpointa aloittaa siitä, miten niihin kuuluvia aineellisia objekteja käytetään digitaalisen informaation esittämiseen ja käsittelemiseen. Useat nykyisistä teorianluomisyrityksistä onkin tehty tällä tavalla.

Aikaisessa yrityksessään teorian luomiseksi Holmquist ja muut [1999] ehdottavat mallia, jossa käyttöliittymään kuuluvat aineelliset objektit voidaan jakaa kolmeen luokkaan: säiliöt (container), merkit (token) ja työkalut (tool). *Säiliöt* ovat tässä luokittelussa yleisluontoisia aineellisia objekteja, jotka eivät ulkoisella olemuksellaan ilmennä niihin liittyvien virtuaalisten objektien luonnetta ja jotka voidaan tilanteesta riippuen liittää moneen erityyppiseen

informaatioon. Säiliöitä käytetään usein informaation siirtämiseen eri kohteiden välillä ja ne ovat pääosin vain tilapäisesti liitettynä virtuaalisiin objekteihin. *Merkit* puolestaan muistuttavat vähintään joltakin aineelliselta ominaisuudeltaan niihin liitettyä virtuaalista objektiä. Merkkien liitokset virtuaalisiin objekteihin ovat monesti luonteeltaan pysyvämpiä kuin säiliöiden vastaavat, eivätkä merkit yleensä voi edustaa kuin tietyn tyyppistä informaatiota. *Työkalut* toimivat edustajina laskennallisille funktioille. Ne voivat esimerkiksi toimia tartuntakahvoina virtuaalisiin objekteihin, tarjota linssin läpi vaihtoehtoisen näkymän käsiteltävään maailmaan tai muuttaa jotakin järjestelmään kuuluvaa virtuaalista ominaisuutta. Jotkut työkalut muistuttavat ominaisuuksiltaan edustamaansa laskennallista funktiota.

Käsiteltävien objektien luokittelun lisäksi Holmquist ja muut esittävät käytettäväksi termiä *hana* (faucet), jolla tarkoitetaan sellaista pistettä, jossa käyttäjän on mahdollista päästä käsiksi merkkeihin liitettyyn informaatioon. Hanoja voivat olla esimerkiksi näytöt, kaiuttimet tai muut vaaditun toiminnallisuuden toteuttavat tulostuslaitteet.

Termi *informaatioon pääsy* (access) puolestaan tarkoittaa merkin esittämistä hanalle, jolloin päästään käsittelemään informaatiota. Pääsyä voidaan kontrolloida merkkien saatavuutta rajoittamalla, vaatimalla usean merkin yhtäaikaista käyttöä tai asettamalla tietyt merkit toimivaan vain tietyissä paikoissa tai sijainneissa. Toinen merkkeihin liittyvä operaatio on *informaation liittäminen* (association), joka tarkoittaa digitaalisen informaation yhdistämistä merkkiin. Järjestelmästä riippuen liitos informaatioon voi olla muutettavissa tai kiinnitetty lopullisesti jo järjestelmän toteutusvaiheessa. Näistä vaihtoehdoista on käytetty myös nimityksiä muuttuva (dynamic) ja muuttumaton (static) liitos [Ullmer and Ishii, 2000]. Joitakin aineellisia objekteja voidaan myös ylikuormittaa (overloading), jolloin yksi objekti liittyy useaan eri informaatioon. Tällöin objektin kautta päästään käsiksi tiettyyn, kontekstista riippuvaan informaatioon tai vaihtoehtoisesti kaikkeen siihen liittyvään informaatioon kerrallaan.

Puhdas objekti	Attribuutti	Substantiivi	Verbi	Uudelleenkonfiguroitava työkalu
----------------	-------------	--------------	-------	---------------------------------

Taulukko 1. Objektien luokittelu Underkofflerin ja Ishiin [1999] kontekstissa.

Kaikki luokittelut eivät välttämättä tähtää yleiskäyttöisten mallien luomiseen. Tämä pätee esimerkiksi seuraavan luokittelun kohdalla, joka edellisen luokittelun lailla koskee yksittäisiä objekteja, mutta vain erääseen tiettyyn sovellustyyppiin liittyen [Underkoffler and Ishii, 1999]. Objektien luokat on sijoitettu akselille sen perusteella, kuinka tarkasti objektit järjestelmässä

vastaavat sitä objektia, jota ne tosielämässä esittävät (taulukko 1). *Puhtaat objektit* (object as pure object) ovat järjestelmän kannalta merkityksellisiä vain objekteina. Ne voivat olla joko yksilöityjä tai yksilöimättömiä, mutta niiden ominaisuuksilla ei ole merkitystä. *Attribuuttiobjekteissa* (object as attribute) järjestelmä välittää vain yhden niiden ominaisuuden arvosta, kuten esimerkiksi muodosta tai väristä. *Substantiiviobjektit* (object as noun) edustavat järjestelmässä koko merkitystään ja toimivat pääosin samalla tavalla kuin vastaavat objektit aineellisessa maailmassa. *Verbiobjektit* (object as verb) taas ovat substantiiviobjekteja abstraktimpia, eivätkä esiinny järjestelmässä itsenään, vaan vaikuttavat substantiiviobjekteihin tai koko ympäristöön. *Uudelleenkonfiguroitavat työkalut* (object as reconfigurable tool) toimivat eri tavoin riippuen kontekstista ja tilasta ja ne voidaan liittää tilanteesta riippuen eri objekteihin.

Yleiskäyttöinen työkalu	Erikoistyökalu	Tunniste	Edustaja	Heijastuma	Sama objekti
-------------------------	----------------	----------	----------	------------	--------------

Taulukko 2. Objektien luokittelu Kolevan ja muiden [2003] mukaan.

Koleva ja muut [2003] ehdottavat käsinkosketeltavien käyttöliittymien luokittelua sen perusteella, kuinka yhtenäisesti aineelliset ja virtuaaliset objektit on liitetty toisiinsa (degree of coherence) (taulukko 2). Samalla he esittelevät luokittelun ympärille rakentuvan teorian ja siihen liittyvät käsitteet.

Luokittelussa heikoin liitos on *yleiskäyttöisillä työkaluilla* (general purpose tool), jotka käyttäjä voi liittää mihin tahansa virtuaaliseen objektiin. Vahvimmillaan liitos taas on silloin, kun toisiinsa liitetyt aineellinen ja virtuaalinen objekti hahmotetaan *samaksi objektiksi*. Näin voi tapahtua esimerkiksi sovelluksissa, joissa objekti voi siirtyä aineellisesta todellisuudesta virtuaaliseen, ollessaan kerrallaan läsnä vain toisessa. Näitä kahta edellä mainittua luokkaa Koleva ja muut eivät laske käsinkosketeltaviksi käyttöliittymiksi, toisin kuin kolme niiden välissä olevaa luokkaa.

Ensimmäinen varsinainen käsinkosketeltavien käyttöliittymien luokka on *erikoistyökalut* (specialized tool), jolla tarkoitetaan tehtävältään yleiskäyttöisiä työkaluja tarkemmin rajoittuneita, mutta silti moniin eri objekteihin liittyviä työkaluja. Hieman vahvempi liitos on *tunnisteilla* (identifier), jotka vastaavat jotakin virtuaalista objektia ja tarjoavat pääsyn siihen. *Edustajat* (proxy) puolestaan ovat vielä vahvemmin ja pysyvämmiin liitettyjä virtuaalisiin objekteihin ja mahdollistavat niiden monipuolisemman käsittelyn. Viimeinen käsinkosketeltaviin käyttöliittymiin kuuluva luokka on *heijastumat* (projection), joihin liittyvät virtuaaliset objektit ovat suoria vastineita aineellisille objekteille ja riippuvaisia näiden olemassaolosta.

Koleva ja muut [2003] ovat myös jakaneet liitosten luonteen ominaisuuksiksi, joiden avulla yhtenäisyyttä voidaan arvioida. Ensimmäinen ominaisuus on *muunnos* (transformation), joka voi saada arvokseen joko kirjaimellinen (literal) tai muunnettu (transformed), riippuen siitä, saako aineelliseen objektiin kohdistuva toiminta saman muutoksen aikaan vastaavassa virtuaalisessa objektissa vai onko näiden kahden välillä eroa. Toinen ominaisuus on *vuorovaikutuksen havaitseminen* (sensing of interaction), jonka avulla määritetään, mitkä aineelliseen objektiin kohdistuvat toiminnot ovat järjestelmän kannalta merkityksellisiä. Näitä voivat olla esimerkiksi pyörittäminen, liikuttaminen tai asettaminen. *Muunnoksen konfiguroitavuus* (configurability of transformation) puolestaan tarkoittaa sitä, voidaanko objektien välissä olevaa muunnosta muuttaa (configurable) vai onko se kiinteä (fixed). *Liitoksen elinikä* (lifetime of link) määrittää onko objektien välinen liitos väliaikainen (temporary) vai pysyvä (permanent). *Itsenäisyys* (autonomy) kuvaa missä määrin virtuaalinen ja aineellinen objekti ovat riippuvaisia toisistaan. Itsenäinen (autonomous) objekti on olemassa toisesta objektista riippumatta, mutta riippuvaiset (dependent) objektit luodaan vasta toisten objektien myötä ja tuhoataan niiden hävittyä. *Liitosten lukumäärä* (cardinality of link) kuvaa moneenko virtuaaliseen objektiin aineellinen objekti on yhtäaikaaisesti liitetty. *Liitoksen lähde* (link source) taas tarkoittaa onko käyttäjälle palautetta antava lähde aineellinen vai virtuaalinen.

	Pysyvä		Joustava	
	Symbolinen	Ikoninen	Symbolinen	Ikoninen
Yleinen				
Henkilökohtainen				

Taulukko 3. Lisäys objektien luokitteluun van den Hovenin ja Eggenin [2004] mukaan.

Eri luokittelut voivat myös laajentaa toisiaan. van den Hoven ja Eggen [2004] ehdottavat laajennusta Ullmerin ja Ishiin [2000] luokitteluun, jotta se tukisi paremmin henkilökohtaisten esineiden käyttöä käyttöliittymän objekteina (taulukko 3). Ensimmäisen luokitteluperusteen mukaan objekti on joko *henkilökohtainen* (personal), jolloin objektin käsittelijä on yleensä myös omistaja, tai *yleinen*, jolloin objektilla ei ole henkilökohtaista merkitystä käyttäjälle. Toisaalta jaotteluun vaikuttaa myös se, onko objektin ja informaation välillä oleva liitos *pysyvä* (fixed) vai *joustava* (flexible). Nämä vastaavat Ullmerin ja Ishiin [2000] muuttuvaa ja muuttumatonta liitosta. Kolmantena jaotteluna

merkit voidaan vielä jakaa *symbolisiin* ja *ikonisiin* niiden semantiikan perusteella.

Vertauskuvallisuus	Olematon	Substantiivi	Verbi	Substantiivi ja verbi	Täysi
Ruumiillistuvuus					
Täydellinen					
Läheinen					
Ympäristöllinen					
Kaukainen					

Taulukko 4. Järjestelmien luokittelu Underkofflerin ja Ishiin [1999] mukaan.

Koska yhtä ainoaa käsinkosketeltavuuden astetta määrittävää ominaisuutta ei ole löytynyt, voidaan luokitteluongelmaa yrittää ratkaista moniulotteisen asteikon avulla. Mitä useampi ulottuvuus asteikolla on, sitä tarkempi kuvaavuus saavutetaan, mutta sitä monimutkaisemmaksi luokittelu käy. Siksi kaksiulotteinen jaottelu saattaa hyvinkin olla sopiva kompromissi. Eräessä kaksiulotteisessa luokittelussa (taulukko 4) asteikkoina toimivat ruumiillistuvuus (embodiment) ja vertauskuvallisuus (metaphor) [Fishkin, 2004]. Tässä lähestymistavassa käyttöliittymän käsinkosketeltavuuden taso voidaan määrittää sen perusteella, kuinka tiukasti käyttöliittymä noudattaa jotakin metaforaa ja kuinka tiukasti käyttöliittymän antama palaute on sidottu kulloinkin käsiteltävänä olevaan objektiin. Mitä enemmän nämä kaksi ominaisuutta käyttöliittymässä toteutuvat, sitä käsinkosketeltavampana käyttöliittymää voidaan pitää.

Ruumiillistuvuus tarkoittaa sitä, kuinka tiukasti syöte ja tuloste ovat sidottu toisiinsa paikan suhteen. Tämä voidaan ymmärtää niin, että mikäli syöte ja tuloste ovat täysin toisiinsa sidottuja, käsittää käyttäjä käsittelemänsä objektin kokonaisuudeksi, joka sisältää kaiken tarvitsemansa toiminnallisuuden. Tässä luokittelussa ruumiillistuvuus voi olla *täydellistä* (full), jolloin syöteobjekti on sama kuin tulostusobjekti, *läheistä* (nearby), jolloin tuloste ilmenee syöte objektin lähellä, *ympäristöllistä* (environmental), jolloin tuloste ilmenee käyttäjän ympäristössä tai *kaukaista* (distant), jolloin tuloste on käyttäjästä etäällä, esimerkiksi jollakin kaukaisella näytöllä.

Vertauskuvallisuus tarkoittaa käyttöliittymäobjektin toiminnan samankaltaisuutta reaali maailman kanssa. Vertauskuvallisuus voi olla olematonta (none), mutta jos sitä on, se voi vastata substantiivia (noun), verbiä (verb), substantiivia ja verbiä (noun and verb) tai olla *täyttä* (full), jolloin vastaavuus on niin ilmeistä, että mitään vertauksia ei tarvitse tehdä. *Substantiivi* vertauskuvana

tarkoittaa sitä, että objekti näyttää esikuvaltaan, mutta ei toiminnaltaan juuri-kaan muistuta sitä ja *verbi* taas sitä, että objektin toiminta muistuttaa jotakin reaali maailman toimintaa. Kuten mainittu, nämä kaksi voivat toteutua samanaikaisesti, jolloin niille on oma luokkansa.

Objektien merkityksiin perustuvissa luokitteluissa on yksi perustavanlaatuinen ongelma. Jos niillä yritetään luokitella kokonaisia järjestelmiä, niin ongelmia tulee vastaan aina kun järjestelmässä on useanlaisia objekteja. Tällöin voidaan sijoittaa järjestelmä siihen luokkaan, johon sen tärkeimmät objektit kuuluvat tai sitten järjestelmän voidaan käsittää kuuluvan useaan luokkaan. Ensimmäinen vaihtoehto on huono, koska se jättää huomioimatta järjestelmän kokonaisuutena, ja jälkimmäisessä taas menetetään luokkien erottelukyky. Esiteltyjä luokitteluja voidaan kuitenkin hyvin käyttää järjestelmien analysointiin ilman luokittelupäämäärää. Tällöin voidaan käsitellä kaikki objektit erillisinä ja saadaan tarkempaa tietoa järjestelmästä.

Osittain erot luokittelujen välillä ovat terminologisia, mutta eri ratkaisuisissa on myös konkreettisia eroja. Holmquistin ja muiden [1999] luokittelu on hyvin kuvaava ja helppo ymmärtää, mutta kuten tekijät itsekin huomauttavat, on yksiselitteinen luokkiin jakaminen siinä ongelmallista. Erityisesti objektien luokittelu joko merkiksi tai työkaluksi saattaa aiheuttaa vaikeuksia. Ongelmia seuraa ainakin jos virtuaalisen objektin ominaisuuksia voidaan muokata sitä itseään esittävällä aineellisella objektilla. Myös objektien tarkoitukseen otettava jyrkkä kanta rajaa luokittelun käyttökelpoisuutta eri konteksteissa.

Underkofflerin ja Ishiin [1999] ja Kolevan ja muiden [2003] luokittelut ovat hyvin samankaltaisia, vaikkakin ensimmäisessä ovat lähtökohtana virtuaaliset ja aineelliset objektit ja toisessa niiden välinen linkki. Kummassakin voi kuitenkin olla vaikeuksia sijoittaa joitakin objekteja yhteen luokkaan, vaikka luokkia on useita. Henkilökohtaisiin objekteihin keskittyvä luokittelu [van den Hoven and Eggen, 2004] osoittaa koko hyötynsä vasta sellaisia sovelluksia luokiteltaessa, joissa käytetään henkilökohtaisia esineitä, mutta tuo myös uusia näkökulmia kaikkien sovellusten analysointiin. Myös viimeiseksi esiteltyä Fishkinin [2004] luokittelua heikentää se, että objekteja analysoidaan yksittäisinä, jolloin eri objektit voivat kuulua eri luokkiin. Siinä myös eri sovellukset voivat kattaa monta aluetta, minkä vuoksi luokittelu menettää tärkeimmän merkityksensä eli sovellusten yksiselitteisen jakamisen eri luokkiin. Lisäksi sovellusten sijoittaminen kummallakin eri asteikolla oikeaan kohtaan saattaa olla vaikeaa. Luokittelussa on silti myös monia hyviä puolia. Se kattaa monia muita luokitteluja ja lisäksi onnistuu luokittelemaan sovelluksia, jotka eivät sisälly mihinkään muuhun luokitteluun. Se myös kuvaa havainnollisesti sitä aluetta, jolle erilaiset käsinkosketeltavat käyttöliittymät voivat kuulua ja välittää informaatiota siitä,

mitä mahdollisuuksia on vielä käyttämättä ja millä alueella on ruuhkaa. Fishkinin luokittelu antaa myös kuvan siitä, kuinka käsinkosketeltavia sovellukset ovat. Tässä yhteydessä täytyy kuitenkin muistaa, että vaikka jonkin sovelluksen käsinkosketeltavuuden aste olisi korkea, siinä ei silti ole välttämättä parempi käyttöliittymä kuin missään muussakaan.

3.3. Objektien suhteisiin perustuvat luokittelut

Vaikka objektien merkitykset ovatkin usein helpoiten havaittavissa, on useissa käsinkosketeltavissa käyttöliittymissä järjestelmän kannalta vielä enemmän merkitystä objektien suhteilla. Suhteisiin perustuvia luokitteluja on vähemmän kuin merkityksiin perustuvia, mutta ne ovat pääosin tuoreempia.

	Esitys		Toiminta		
	Merkki	Rajoitteet	Muuttuja	Tapahtuma	Palaute
TAC					

Taulukko 5. Objektien analysointi Shaerin ja muiden [2004] mukaan.

TAC-paradigma [Calvillo-Gómez *et al.*, 2003; Shaer *et al.*, 2004] tarjoaa hieinan erilaisen työkalun käsinkosketeltavien käyttöliittymien analysointiin, sillä sen avulla ei ole tarkoituskaan luokitella käyttöliittymiä, vaan kuvailla ja arvioida niitä (taulukko 5). Tässä teoriassa kaikki käyttöliittymään kuuluvat aineelliset objektit (pyfo) voivat olla joko merkkejä (token), rajoitteita (constraint) tai molempia.

Analysoinnin lähtökohtana ovat *merkit*, joilla tarkoitetaan niitä aineellisia objekteja, joihin voidaan tarttua ja jotka edustavat digitaalista informaatiota. Merkin käyttäytymistä ja käsittelemistä rajoittavat aineelliset objektit taas ovat sen *rajoitteita*. Yksi TAC (Token And its Constraints) koostuu siis merkistä ja sen rajoitteista. Jotta objekti voi olla merkki, täytyy siihen liittyä vähintään yksi rajoite sekä digitaalista informaatiota edustava *muuttuja* (variable). Merkki voi rajoittaa muiden merkkien käyttäytymistä, jolloin se on rajoitteena muissa TAC:eissa. Yksi rajoite voi myös samanaikaisesti rajoittaa useita merkkejä. Kaikkien merkkien on oltava tietoisia omista rajoitteistaan, mutta rajoitteiden ei tarvitse tietää niihin liittyviä merkkejä. Digitaaliseen informaatioon merkit on sidottu muuttujien kautta, ja jokaisen TAC:in on liityttävä johonkin muuttujaan ja muutettava sitä joko *erillisen* (discrete) tai *jatkuvan* (continuous) toiminnan kautta. Erilliseksi muuttujaksi voidaan lukea esimerkiksi napin painalluksesta vaihtuva muuttuja, kun taas jonkin objektin suuntaa ilmaiseva muuttuja on jatkuva.

Rajoite voi määrittää merkkien toimintaa kolmella erilaisella tavalla. Ensinnäkin sen havaittavissa olevat ominaisuudet voivat antaa käyttäjälle ohjeita

siitä, kuinka rajoitteeseen liittyvää merkkiä tulisi käsitellä. Toisaalta rajoite saattaa myös fyysisesti määrittää sen tilan, jossa merkkiä voidaan liikuttaa. Kolmanneksi rajoite voi toimia koordinaatistona, jonka suhteen merkkien toimintaa tulkitaan ja muunnetaan laskennallisiksi funktioiksi.

Kun edellisessä luokittelussa käsinkosketeltavia käyttöliittymiä analysoitiin jokaisen niihin kuuluvan aineellisen objektin kohdalla erikseen, Ullmer ja Ishii [2000] esittävät neljä kategoriala, joiden avulla voidaan tulkita useiden objektien muodostamaa järjestelmää kokonaisuutena (luokittelun ei kuitenkaan ole tarkoitus olla kattava). Heidän mukaansa luokittelun avulla voidaan analysoida objektien suhdetta digitaaliseen informaatioon. Alun perin luokkia oli neljä: miellelyhtymiin perustuva (associative), tilaan perustuva (spatial), suhteisiin perustuva (relational) ja rakenteisiin perustuva (constructive). Myöhemmin he kuitenkin jättivät miellelyhtymiin perustuvan luokan pois, koska eivät olleet sen hyödyllisyydestä yhtä varmoja kuin kolmen muun luokan [Ullmer and Ishii, 2000]. Tässä tutkielmassa kaikki neljä ovat kuitenkin mukana, sillä myös pois jätetyllä luokalla on osoitettu olevan arvoa [van den Hoven and Eggen, 2004].

Miellelyhtymiin perustuvoissa lähestymistavoissa objektit liittyvät kukin itseenäisesti digitaaliseen informaatioon, eivätkä saa lisämerkityksiä suhteistaan muihin objekteihin. *Tilaan perustuvoissa* järjestelmissä merkkien avaruudellinen sijainti ja suunta voidaan tulkita suoraan syötteenä, kun taas *suhteisiin perustuvoissa* järjestelmissä tulkinta tehdään merkkien peräkkäisyyksien, vierekkäisyyksien tai muiden vastaavanlaisten suhteiden perusteella. *Rakenteisiin perustuvoissa* järjestelmissä merkit ovat moduuleita, jotka voidaan kiinnittää toisiinsa rakennuspalikoiden tavoin mekaanisesti.

Kokonaisia käsinkosketeltavia käyttöliittymiä on yritetty luokitella myös niiden rakenteen perusteella. Eräässä luokittelussa käsinkosketeltavat käyttöliittymät jaetaan kolmeen lajityyppiin: vuorovaikutteisiin pintoihin (interactive surface), rakennettaviin järjestelmiin (constructive assembly) ja merkki+rajoitteet -järjestelmiin (tokens and constraints) [Ullmer et al., 2005]. *Vuorovaikutteisiksi pinnoiksi* voidaan käsittää graafisesti rikastetut (augmented) tasot ja seinät, joiden pinnalla sijaitsevia aineellisia objekteja käyttäjät käsittelevät. *Rakennettaviksi järjestelmiksi* puolestaan luetaan sellaiset käyttöliittymät, joissa vuorovaikutus perustuu rakennuspalikoiden tavoin toimivien objektien yhteenliittämiseen tai kokoamiseen. *Merkki+rajoitteet* -järjestelmissä merkit sisältävät viittauksia digitaaliseen informaatioon, ja tietokone tulkitsee niiden käsittelemistä, joka tapahtuu niiden omien rajoitteiden asettamisissa puitteissa. *Merkki+rajoitteet* -järjestelmien vuorovaikutus voidaan jakaa kahteen vaiheeseen: yhdistämiseen (associate) ja käsittelemiseen (manipulate). *Yhdistämisessä* merkki asetetaan johonkin rajoitteiden määrittämään ja järjestelmän kannalta

merkitykselliseen paikkaan tai sijaintiin. Yhdistämisen jälkeen objektin *käsittelen* on mahdollista rajoitteiden sallimissa puitteissa, esimerkiksi objektia pyörittämällä tai liikuttamalla. Osa merkki+rajoitteet -järjestelmistä tukee vain yhdistämisvaihetta.

Vaikka objektien suhteisiin perustuvilla luokitteluilla olisi mielestäni paljon potentiaalia käsinkosketeltavien käyttöliittymien luokittelussa, liittyy niihin samankaltaisia ongelmia kuin objektien merkityksiin perustuviin luokitteluihin. Vaikka suhteisiin perustuvien luokittelujen avulla voidaan periaatteessa luokitella kokonaisia järjestelmiä, saattavat eri objektien suhteet vaihdella järjestelmän sisällä, jolloin yksiselitteistä luokittelua ei välttämättä voida saada aikaan.

Ensimmäisenä esitelty TAC-paradigma [Calvillo-Gómez *et al.*, 2003; Shaer *et al.*, 2004] huomioi objektien erilaisuuden käsittelemällä jokaisen objektin erikseen, mutta menettää samalla mahdollisuuden luokitteluun. Se on hyvä työkalu joidenkin käsinkosketeltavien käyttöliittymien ja niiden vuorovaikutuksen analysoinnissa, mutta ei sovi yhtä hyvin kaikkien sovellustyyppien analysointiin. Toisena esitelty Ullmerin ja Ishiin [2000] jaottelun kolme ensimmäistä luokkaa sopivat hyvin yhteen, mutta vaikeuksia aiheuttaa neljäs, rakennettavia järjestelmiä kuvaava luokka. Mielestäni luokka on enemmänkin ominaisuus, joka voi yhtä hyvin liittyä mihin tahansa muista luokista, vaikka on esitetty omana luokkana. Sen joukkoon kuulumattomuutta puolustaa myös se, että sama luokka esiintyy samanlaisena toisessa myöhemmin esiteltävässä Underkofflerin ja Ishiin [1999] luokittelussa, johon se sopii paremmin. Tämän jälkimmäisen luokittelun ongelmana on kuitenkin se, että on olemassa käyttöliittymiä, jotka eivät kuulu mihinkään esitettyyn luokkaan. Luokittelua voidaan pitää enemmänkin hyvin kuvailevana, kuin tarkasti eri ryhmiin jakavana. Luokat ovat kuitenkin erittäin konkreettisia ja antavat nopeasti yleiskuvan järjestelmän luonteesta. Mielestäni tämä luokittelu onkin lähimpänä onnistua kuvailemaan käsinkosketeltavia käyttöliittymiä yhdellä akselilla.

4. Lopuksi

Tässä tutkielmassa esiteltiin ja vertailtiin erilaisia käsinkosketeltavien käyttöliittymien luokitteluja ja teorioita.

Objektien merkityksiin perustuvia jaotteluita on ollut olemassa jo melko pitkään, mutta täydellistä jaotteluperustetta ei ole vielä löytynyt. Lähimmäksi tässä pääsee Fishkin [2004], jonka luokittelun avulla voidaan saavuttaa hyötyjä sovellusten suunnittelussa, analysoinnissa ja vertailussa. Objektien merkityksiin perustuvissa luokitteluissa on ongelmana se, että yhteen käyttöliittymään

sisältyy useasti monen tyyppisiä objekteja, jolloin päätelmiä koko järjestelmästä on vaikea yleistää luokittelua varten.

Objektien suhteisiin perustuvia luokitteluja ei ole ollut olemassa yhtä kauan kuin merkityksiin perustuvia. Niiden etuna ovat kuitenkin paremmat edellytykset kokonaisten järjestelmien luokitteluun. Vielä ne eivät analysoinnin tarkkuudessa kuitenkaan pärjää parhaille merkitysten analysointiin perustuvilla luokitteluilla. Underkofflerin ja Ishiin [1999] luokittelu ei ole täydellinen, mutta kuvaa yksinkertaisesti muutamalla luokalla koko suunnittelutilaa.

Edellä esiteltyjen lisäksi on olemassa myös muita luokitteluja ja teorioita, joita voidaan soveltaa käsinkosketeltaviin käyttöliittymiin, mutta jotka kuitenkin sopivat myös muihin vuorovaikutustapoihin (esimerkiksi [Wensveen *et al.*, 2004]) tai ovat suunniteltu erityisesti jonkin käsinkosketeltavien käyttöliittymien alaryhmän arviointiin (esimerkiksi [Hornecker, 2004]). Tässä tutkielmassa olen kuitenkin keskittynyt vertailemaan vain niitä teorioita, jotka kuvaavat käsinkosketeltavia käyttöliittymiä kokonaisuutena määrittelyjen rajoja ylittämättä.

Mielestäni yksikään edellä esitellyistä käsinkosketeltavien käyttöliittymien luokitteluista ei ole niin hyvä, että tutkimus aihepiiriin liittyen kannattaisi lopettaa. Ei ole kuitenkaan selvää, kannattaisiko painopiste suunnata yksinkertaisen muutamaa luokkaan sovellukset jakavan luokittelun aikaansaamiseen vai useampiulotteisen luokittelun luomiseen aivan uusien akseleiden ympärille. Tutkimussuuntauksen tulevaisuudessa löytyy varmaankin molemmille tarvetta.

Viiteluettelo

- [Calvillo-Gómez *et al.*, 2003] Eduardo H. Calvillo-Gómez, Nancy Leland, Orit Shaer and Robert J. K. Jacob, The TAC paradigm: unified conceptual framework to represent tangible user interfaces. In: *Proceedings of CLIHC 2003, Latin American Conference on Human-Computer Interaction (2003)*, ACM Press, 9-15.
- [Fishkin, 2004] Kenneth P. Fishkin, A taxonomy for and analysis of tangible interfaces. *Personal and Ubiquitous Computing* 8, 5 (2004), 347-358.
- [Fitzmaurice *et al.*, 1995] George W. Fitzmaurice, Hiroshi Ishii and William Buxton, Bricks: laying the foundations for graspable user interfaces. In: *Proceedings of CHI 1995, Conference on Human Factors in Computing Systems (1995)*, ACM Press, 442-449.
- [Holmquist *et al.*, 1999] Lars Erik Holmquist, Johan Redström and Peter Ljungstrand, Token-based access to digital information. In: *Proceedings of*

- HUC 1999, Handheld and Ubiquitous Computing: First International Symposium* (1999), Springer, 234-245.
- [Hornecker, 2004] Eva Hornecker, A framework for the design of tangible interaction for collaborative use. In: *Proceedings of Danish HCI Research Symposium* (2004), 57-61.
- [van den Hoven and Eggen, 2004] Elise van den Hoven and Berry Eggen, Tangible computing in everyday life: extending current frameworks for tangible user interfaces with personal objects. In: *Proceedings of Ambient Intelligence: Second European Symposium* (2004), Springer, 230-242.
- [Ishii and Ullmer, 1997] Hiroshi Ishii and Brygg Ullmer, Tangible Bits: towards seamless interfaces between people, bits and atoms. In: *Proceedings of CHI 1997, Conference on Human Factors in Computing Systems* (1997), ACM Press, 234-241.
- [Koleva *et al.*, 2003] Borianna Koleva, Steve Benford, Kher Hui Ng and Tom Rodden, A framework for tangible user interfaces. In: *Proceedings of 5th International Symposium, Mobile HCI Conference* (2003), Springer, 46-50.
- [Shaer *et al.*, 2004] Orit Shaer, Nancy Leland, Eduardo H. Calvillo-Gamez and Robert J. K. Jacob, The TAC paradigm: specifying tangible user interfaces. *Personal and Ubiquitous Computing* **8**, 5 (2004), 359-369.
- [Ullmer and Ishii, 2000] Brygg Ullmer and Hiroshi Ishii, Emerging frameworks for tangible user interfaces. *IBM Systems Journal* **39**, 1 (2000), 915-931.
- [Ullmer and Ishii, 2001] Brygg Ullmer and Hiroshi Ishii, Emerging frameworks for tangible user interfaces. In: John M. Carroll (ed.), *Human-Computer Interaction in the New Millennium*. Addison-Wesley, 2001, 579-601.
- [Ullmer *et al.*, 2005] Brygg Ullmer, Hiroshi Ishii and Robert J. K. Jacob, Token+constraint systems for tangible interaction with digital information. *ACM Transactions on Computer-Human Interaction* **12**, 1 (2005), 81-118.
- [Underkoffler and Ishii, 1999] John Underkoffler and Hiroshi Ishii, Urp: a luminous-tangible workbench for urban planning and design. In: *Proceedings of CHI 1999, Conference on Human Factors in Computing Systems* (1999), ACM Press, 386-393.
- [Weiser, 1991] Mark Weiser, The computer for the 21st century. *Scientific American* **265**, 3 (1991), 94-104.
- [Wellner, 1993] Pierre Wellner, Interacting with paper on the DigitalDesk. *Communications of the ACM* **36**, 7 (1993), 87-97.
- [Wensveen *et al.*, 2004] Stephan Wensveen, J. P. Djajadiningrat and Kees C. J. Overbeeke, Interaction frogger: a design framework to couple action and function through feedback and feedforward. In: *Proceedings of DIS 2004, Conference on Designing Interactive Systems* (2004), ACM Press, 177-184.

Sukupuoliroolit tietotekniikan alalla Suomessa

Anne Pyykönen

Tiivistelmä.

Tässä tutkimuksessa tarkastelen naisten asemaa tietotekniikan alalla Suomessa. Tietotekniikan alaa on pidetty miehisenä kenttänä, niin historiallisesti tarkasteltuna kuin nykypäivän tilannetta analysoitaessa. Tämän tutkielman tavoitteena on valottaa niitä tekijöitä, jotka ovat vaikuttaneet alan sukupuolisten käytäntöjen muodostumiseen. Samassa yhteydessä tarkastelen myös, kuinka sukupuoliset ennakoasenteet ovat ilmenneet ja mitkä ovat niiden vaikutukset nykypäivän tietotekniikan asiantuntijuuteen.

Avainsanat ja -sanonnat: Tietotekniikka, sukupuolittuneet käytännöt, asiantuntijuus.

CR-luokat: K.2, K.7.1

1. Johdanto

Tutkimusmielenkiintoni kohdistuu naisten asemaan tietotekniikan alalla erityisesti historiallisesta näkökulmasta, mutta luon katsauksen myös nykypäivän tilanteeseen. Tietotekniikka ja sukupuoli ovat erottamaton pari, ja jo stereotyyppiatkin kertovat miehistä tietokoneiden äärellä – tietotekniikan alan sanotaankin olevan erityisen miehinen [Vehviläinen 1996, 1997a ja 1997b; Tiainen 2002].

Marja Vehviläinen [1997a] on todennut, että tietotekniikka on maailma ilman naisia; maailma, joka on rakennettu miehisestä näkökulmasta; maailma, jossa naisilla on hyvin vähän sananvaltaa. Tietotekniikan maailma kokonaisuudessaan muodostuukin historiallisista ja yhteisöllisistä lähtökohdistaan, jotka vaihtelevat maittain ja organisaatioittain [Vehviläinen 1997a]. Vehviläinen [1997b] tiivistää asian näin: "[T]ietotekniikan sukupuolijakojen 'suuri linja' on kulkenut 1960-luvun tietotekniikan alkuaian 'miesten maailmasta' 1970- ja 1980-lukujen lisääntyvään naisten tilan kautta 1990-luvun miehistymiseen. Linjan sisällä sukupuolijaot muotoutuvat vaihdelleen yhteiskunnasta, organisaatiosta ja jopa tilanteesta toiseen." Tämän tutkielman tavoitteena on avartaa joitakin niitä Suomelle tyypillisiä lähtökohtia, jotka ovat vaikuttaneet alan sukupuolisiin käytäntöihin.

Toisessa luvussa luon tälle tutkielmalle tarpeellisen yleisen katsauksen työelämän sukupuolittuneisiin käytäntöihin. Kolmannessa luvussa tarkastelen alalla historiallisesti vallitsevia ammatillisia sukupuolirooleja. Neljännessä luvussa valotan joitakin sukupuolittumiseen vaikuttaneita historiallisia taustatekijöitä. Viidennessä luvussa kuvaan alan sukupuolittumisen ilmenemis-

muotoja. Kuudennessa luvussa kerron, kuinka naiset vihdoin valtasivat edes jonkinlaista jalansijaa alalla. Seitsemännessä luvussa esittelen alan tyypillisiä henkilökuvia. Kahdeksannessa luvussa kuvaan, kuinka tietotekniikan asiantuntijuutta voidaan edelleenkin pitää maskuliinisena alueena ja miten se vaikuttaa tietotekniikan alaan kokonaisuudessaan. Yhdeksännessä luvussa esittelen mahdollisuuksia ja motiiveja muuttaa nykyistä tilannetta. Kymmenennessä luvussa teen päätelmiä tutkielman teemoista.

2. Yleisesti työelämän sukupuolittuneista käytännöistä

Sukupuolen perusteella tehtävä työnjako on yhtä vanha kuin on ihmisten yhteistoimintakin. Yhtä luontevasti myös tietyt taidot yhdistetään joko miehille tai naisille ominaisiksi: näppäryys, empaattisuus, huolellisuus ja pikkutarkkuus ovat naisten ominaisuuksia ja vastuunkanto, kokonaisuuksien hallinta, riskinotto, ongelmanratkaisu- ja tekniset taidot ovat miesten vahvuuksia. Taidot eivät itsessään ole toisia ylempiä tai alempia, mutta ne pitävät yllä ja tuottavat käsitystä siitä, millaiseksi naiset ja miehet ja heidän työnsä kulttuurissamme ymmärretään. [Kinnunen ja Korvajärvi 1996.]

”Suomalaisessa työelämässä keskeinen sukupuolistava käytäntö on paradoksaalisesti naissukupuolen samanaikainen läsnäolo ja poissaolo. Samanaikaisen läsnäolon ja poissaolon käytäntö antaa tilaa naisille, mutta naisten tilat ovat heidän omia erityisiä tilojaan [Korvajärvi ja Kinnunen 1996]. Toinen paradoksi on yhtäaikainen tasa-arvo ja oikeudenmukaisuuden puute: naiset ovat kyllä työelämässä vahvasti mukana, mutta heidän etenemismahdollisuutensa ovat perinteisesti miehiä huonommat [Korvajärvi 1998].

Suomalaiselle työelämälle on tyypillistä sekä horisontaalinen että vertikaalinen segregatio, eli työnjako. Horisontaalinen segregatio merkitsee sitä, että miehet ja naiset sijoittuvat eri tehtäviin ja ammatteihin erilaisissa organisaatioissa, eri toimialoilla ja työmarkkinoiden sektoreilla. Vertikaalinen segregatio merkitsee miesten sijoittumista hierarkkisesti naisia ylempiin asemiin sekä ammattihierarkkisesti että samassa työtehtävissä toimiessaan. [Kolehmainen-Lindén 1996.]

Perinteisten kulttuurisen järjestyksen tyypillinen ominaisuus on, että se tuntuu toistavan itseään aina uusissa ympäristöissä ja käytännöissä, ja se kiertyy osaksi maailmankuvamme, tajuntaamme ja tapamme olla sukupuolemme edustajia. Sen muuttaminen tai edes lieventäminen on vaikeaa. [Lavikka 1996.]

3. Historialliset sukupuoliroolit tietotekniikan alalla

Naisten ja miesten työnjaolla tietotekniikan maailmassa on pitkät perinteet. Kuten Marja Vehviläinen [1996] toteaa, tietotekniikka on kansainvälisesti lei-

mautunut miehiseksi alueeksi, mutta Suomessa naisille on ollut tilaa tietotekniikan ammateissa, vaikka toisaalta havaittavissa onkin ollut naisia poissulkeva sävy. Suominen [2003] kuvaa Suomen tietoteknistymistä, ja hänen mukaansa jo ensimmäisten reikäkorttikoneiden myötä miehet olivat esimiesasemassa, ja naiset hoitivat suorittavan työn. Suominen [2003] mukaan jo 1920- ja 1930-luvulla ”miehet toimivat koneiden valvojina, naiset yhä enemmän datan syöttäjinä ja käsittelijöinä”.

Suominen [2003] listaa ammattinimikkeitä, joita tietotekniikan alalla tunnettiin esimerkiksi 1950-luvulla: reikäkorttien lävistäjät, operatöörit, jotka vastasivat koneiden käyttö- ja kytkentätöistä, sekä päälliköt. Vehviläinen [1996] esittelee vuoden 1961 käsitystä alan ammattinimikkeistä: osastopäällikkö, ohjelmoitsija, konsolioperatööri, operatööri, tarkistuslävistäjä, lävistäjä, tekninen henkilökunta, matemaatikko ja konttorisuunnittelija. Tuolloin tehtävät olivat selkeästi sukupuolittuneet, sillä Vehviläisen [1996] mukaan esimerkiksi lävistäjät olivat kaikki naisia. Tilanne oli siis sellainen, jossa ”miehet johtivat ja hoitivat asiantuntijatehtävät sekä asiantuntijuuden kehittämisen, ja naiset olivat rutiininomaisissa töissä lävistäjinä ja operatööreinä” [Vehviläinen 1996].

Joissakin tapauksissa naiset eivät päässeet edes operaattoreiksi, kuten Suominen [2003] kuvaa erään naisen tarinaa työnhaustaan Tietokonepalvelu Oy:öön:

”Hain lehti-ilmoituksen perusteella operaattoriksi. Sain vastauksen jossa ilmoitettiin, ettei taloon oteta naispuolisia operaattoreja, mutta lävistäjän paikka oli tarjolla ja niin tulin lävistäjäksi TKP:hen. Tulin taloon 15.5.62. Aluksi harjoittelin puhelinluettelosta lävistämistä. Ensimmäinen olin lävistyksessä ja sen jälkeen korjauslävistyksessä.”

Sirpa Kolehmainen [1999] on tutkinut työmarkkinoiden segregatiota vuosien 1970–1990 ajalta. Kolehmainen [1999] mainitsee tarkastelujakson aikana naisistuneista ammateista mm. tietokoneoperaattorin, kun taas ”ATK-päälliköiden, -suunnittelijoiden ja ohjelmoijien ammattiryhmä kuului tarkasteluaikana voimakkaimmin laajentuneisiin miesammattiteihin. Naistyövoiman kasvusta huolimatta tämä ammattiryhmä säilyi miesvaltaisena”.

4. Sukupuolittumisen taustoja

Suominen [2003] toteaa, kuinka yleisestikin tekniikka, myös tietotekniikka, käsitetään usein miehiseksi kenttänä, vaikka kokonaisuutta tarkasteltaessa alalla on aina ollut myös naisia – joskin piilotetummassa, luonteeltaan rutiininomaisissa tehtävissä verrattuna näkyviin miesten hallitsemiin johtamis- ja suunnittelutöihin. Suominen [2003] korostaa kuitenkin, ettei alan miehisyyttä välttämättä ole ollut hegemonista, vaan joidenkin tietokoneiden feminiinisyyttä

puhuttaessa käyttäytymisestä, ajattelutavoista ja ruumiinvoimien merkityksestä on ollut vaikuttamassa tietotekniikan sukupuolisuuden muotoisuuteen.

Vehviläinen [1994a] toteaa, että miesten ylivoimainen asema tietotekniikan alalla on monen tekijän yhteisvaikutusta. Esimerkiksi naisten perinteinen asema työmarkkinoilla, hyvinvointivaltion tuki ja palvelut perheellisille, sukupuolikäytännöt tietyissä instituutioissa, sekä alan koulutus ovat vaikuttaneet kukin omalla tavallaan [Vehviläinen 1994a].

Suominen [2003] esittää oivallisesti, kuinka sodat vaikuttivat tietotekniikan sukupuolittumiseen, ja kysyy, olivatko naiset petun tapaista korviketta, kun miehet olivat rintamalla. Sodan vaikutus tulee ilmi muullakin tavoin, sillä ei voida väheksyä sotakokemusten merkitystä sen kokeneiden miesten elämään. Siten sota on vaikuttanut myös heidän asenteisiinsa ja tapoihinsa tehdä työtä. Marja Vehviläinen [1996] kuvaakin, kuinka sotilastermit ja armeijamainen työtapa olivat ominaista alan pioneereille Suomessa.

Vehviläinen [1996] on kartoittanut sukupuolen suomalaista muotoutumista erityisesti miespioneereja tutkimalla. Esimerkkinä voidaan pitää Otto Karttusta, jonka omaelämäkerrallisia tarinoita Vehviläinen [1996] on tutkinut kartoittaakseen Suomen tietokoneistumisen historiaa. Tässä yhteydessä voidaankin todeta, miten muutamat tienraivaajat voivat vaikuttaa kokonaisen maan kehitykseen jollakin osa-alueella. On huomioitavaa, että ala muodostuu väistämättäkin esikuviansa näköiseksi. Se, että Otto Karttusen tapainen mies oli eturintamassa myös Suomen tietokoneistamisessa, on tehnyt Suomen tietokonekulttuurista Otto Karttusen näköisen. Mikäli voimapersoonana olisi ollut joku muu, tulos olisi eittämättä ollut toinen. Nykytilanne on tulosta siitä, että armeijatutut istuivat suomalaisen yhteiskunnan kärkipaikoilla, ja he muodostivat toimintatavat kullekin alalle [Vehviläinen 1996].

Vehviläinen [1996] puhuu reikäkorttimiehistä tarkoittaessaan vuonna 1953 perustetun Reikäkorttiyhdistyksen jäseniä. Reikäkorttikoneiden korvautuessa tietokoneilla tietokoneasiantuntijat olivat entisiä reikäkorttimiehiä. Reikäkorttiyhdistyksen jäsenistä vain muutama prosentti oli naisia. Yhdistyksen nimi muutettiin Tietokoneyhdistykseksi vuonna 1960. [Vehviläinen 1996.]

Reikäkorttimiesten hyvä veli -järjestelmä kehittyi ATK-kerhoksi vuonna 1961, ja se toimi varsinaisen emoyhdistyksen rinnalla. ATK-kerhon vaikutukset ovat laajat sekä suomalaisessa tietokoneistumisessa, alan koulutuksessa, ja myös sukupuolittumisessa: miessukupuolesta tuli ainoa sukupuoli pitkäksi aikaa. [Vehviläinen 1996.]

Reikäkortti/Tietokoneyhdistyksen vaikutus alan sukupuolittumiseen on siis kiistaton. Alan käytäntöjä luovat pioneerit olivat kauttaaltaan miehiä. Jopa

yhdistyksen säännöistä kävi ilmi naisten asema alalla. Vasta kun yhdistyksen jäseniksi hyväksyttiin muitakin kuin alan johtohenkilöitä, myös naiset pääsivät yhdistyksen toimintaan mukaan. Naiset olivat pitkään sivussa sekä yhdistyksen toiminnasta että sen julkaisemassa lehdestä, ja ala olikin pitkään ”maailma ilman naisia”. [Vehviläinen 1996.]

Tutkiessaan tietotekniikan alan sukupuolittumisen historiaa, Vehviläinen [1996] nostaa esille vielä yhden seikan, jota ei voi sivuuttaa: IBM:n vaikutusta. Alan koulutuksesta vastasivat pitkään vain koneiden maahantuojat, Suomessa erityisesti IBM, sillä se nousi pian valta-asemaan markkinoilla - IBM oli tunnetusti miesvaltainen yhtiö [Suominen 2003]. Vehviläinen [1996] toteaaakin, että “[T]ietokonealan vahva yhteys IBM:n kanssa on tuskin ainakaan horjuttanut tietokoneasiantuntemuksen leimautumista miehiseksi.”

5. Vitsejä, työttelyä ja objektivointia

Vaikka naiset olivatkin ulkona Reikäkortti/Tietokoneyhdistyksen lehden teosta, he olivat usein sen aiheena, mutta vain vitseissä ja kaskuissa. Jaakko Suominen [2003] kuvaa, kuinka Reikäkorttilehti julkaisi alaan kuulumattomia vitsejä maskuliinisesta näkökulmasta, joka yhtäläillä korostaa yhdistyksen miesvaltaisuutta – vitsien sankareina, ja joskus antisankareinakin olivat miehet ja pojat, ja naisille jäi nalkuttavan vaimon tai äidin rooli.

Myös lehden kuvituksessa ja mainoskuvastossa on Suomisen [2003] mukaan havaittavissa sukupuolisesti värittyneitä vivahteita. Esimerkkinä Suominen [2003] mainitsee alaotsikon ”Kuuluuko tyttökin sopimukseen?” esiteltäessä IBM 650-konetta. Työttely oli yleistä alalla aikana, jolloin naiset eivät olleet yksilöitä vaan osa kollektiivista koneistoa, kuten Suominen [2003] toteaa lainatessaan Jennifer S. Lightin huomiota siitä, kuinka naisia kutsuttiin johtajiensa tai käyttämiensä koneiden mukaan esimerkiksi ENIAC-tyttöiksi.

Suominen [2003] toteaa, että ”ammattilehtien kuvissa, erityisesti mainoksissa, sukupuolten roolit pysyivät pitkälle muuttumattomina 1960-luvun aikana”. Naiset tekivät tarkkuutta vaativia naisten töitä ja olivat sieviä. Miehet olivat asiantuntevia edelläkävijöitä. 1960-luvun lopulla konepainotusten rinnalle mainoksiin tuli naisten koristava kohteellistaminen eli objektivointi – nainen oli sekä työkone, että halujen kohde. Tämä ilmeni erityisesti flirttaillevien tietokonenaisten kuvaamisena, jossa nuoret ja kauniit naiset esitettiin kameraan katsovina, mutta kuitenkin työhönsä sidottuina, objekteina. Yleinen teema oli myös naisen ja koneen, tai naistyöntekijän ja miesjohtajan rakkaussuhde. Nainen esitettiin koneeseen ihastuneena, tunteellisena oliona. [Suominen 2003.]

Suominen [2003] esittääkin, ettei nainen olisi voinut olla tietotekniikan alan asiantuntija, innovaattori tai visionääri 1960-luvun Suomessa, sillä nainen ulkoistettiin tietotekniikan asiantuntijuudesta täysin. Suominen [2003] perustelee, että asiantuntijaksi ei päässyt tietojensa tai taitojensa perusteella, vaan status voitiin saavuttaa vain sellaisten käytänteiden kautta, joihin naisilla ei silloin ollut asiaa.

Yllä kuvatuunlainen tilanne on ollut myös muualla maailmassa, sillä myös Selby ja muut [1998] kuvaavat uusiseelantilaista tietotekniikan henkilögalleriaa oman maansa julkisuudessa samantyyppisin sanakäntein.

6. Naisten tie alalle kävi koulutuksen kautta

Vehviläinen [1996] korostaa, että vaikka Reikäkortti/Tietokoneyhdistyksen ja ATK-kerhon vaikutukset ovat olleet sukupuolijakoja ja -hierarkioita luovia, ne ovat olleet myös edesauttamassa tietoteknisen ammatti- ja korkeakoulutuksen syntymistä Suomeen 1960-luvulla, ja sitä kautta naisten pääsyä alalle. Vehviläisen [1996] mukaan naisten asiantuntemus oli ennen korkeakoulutusta ”alempaa” asiantuntemusta, ja naiset myös pitäytyivät ns. alemmissa ammattiryhmissä. Suomessa koulutus on aina ollut ainakin periaatteessa mahdollista molemmille sukupuolille, ja tämä muodostikin murroksen naisten siirtymisessä asiantuntijatehtäviin tietotekniikan alalla [Vehviläinen 1996].

Suomessa naisilla oli ollut vahva asema luonnontieteellisten ja kaupallisten tiedekuntien piirissä, joten siirtyminen tietojenkäsittelyoppiin oli mahdollista tiedekuntien sisällä. Suunnittelijan ammatti avautui myös naisille. [Vehviläinen 1996.]

Kolehmainen [1999] toteaa, että koulutus sekä murtaa että vahvistaa sukupuolten välistä työnjakoa, eli segregatiota. Kolehmaisesta [1999] mukaan miesammateissa, joihin myös tietotekniikan ammatit voidaan lukea, työskentelevillä oli muita yleisemmin korkea-asteen koulutustausta, ja tutkimuksen tarkasteluajankohdan, eli vuosien 1970–1990 aikana, naisistuneet miesammatit olivat olleet nimenomaan korkeakoulutusta vaativia asiantuntija-ammatteja. Kolehmainen [1999] kuitenkin toteaa, että ”päästäkseen miesten työmarkkinoille naisten täytyy olla selvästi pätevämpiä kuin samoissa töissä ja ammateissa toimivat miehet”. Kouluttautuminen on osaltaan edistänyt naisten mahdollisuuksia miehisissä ammateissa, ja naisten korkeakoulutus on yleistynyt nimenomaan miesammateissa [Kolehmainen 1999].

Suominen [2003] vahvistaa käsityksen, että naisten esiintulo alaa koskevassa keskustelussa vei aikaa, ja tämä käy ilmi esimerkiksi Suomisen [2003] tarkastelemassa aikakauslehtiartikkeli-indeksistä. Aikakauslehti-indeksistä käy ilmi, ketkä atk-asiantuntijat kirjoittivat eniten artikkeleita ammattilehtiin.

Ahkerimmat kirjoittajat olivat kaikki miehiä, ja vain 2,6 % kaikista artikkeleista oli selkeästi naisten kirjoittamia, ja ne oli kirjoitettu 1970-luvun puolella. [Suominen 2003.]

7. Tietokonetyöläisten historialliset stereotyypit

Tietotekniikan alan asiantuntijaa ovat historiallisesti määrittäneet samat piirteet kuin minkä tahansa muun alan asiantuntijaa: hän on mies, omaa korkean koulutuksen ja on korkeassa asemassa organisaatiossaan. Erityisesti tietotekniikan ammattilaisen piti Suominen [2003] mukaan hallita oman alansa lisäksi myös laaja yleissivistys. Suomalaisen tietotekniikan alan asiantuntijakuvaa muokkasivat 1960-luvulta lähtien mm. Valittujen Palojen henkilökuvat, jotka korostivat "pioneerien peräänantamattomuutta ja lahjakkuutta -- [P]ioneeri uhraisi elämänsä tieteen ja teknologian alttarille työskenneltyään peräänantamattomasti kaikkien hyväksi loppuun asti" [Suominen 2003].

Tietotekniikan alalla 1960-luvulla tyypillisimmäksi naisten työksi muotoutuneesta reikäkorttien lävistämisestä ei Suominen [2003] mukaan juuri julkisuudessa puhuttu, korkeintaan se kuvattiin näppäryyttä vaativana aputyönä, jonka suoritti hehkeä nuori nainen.

Stereotyyppioita vahvistivat erityisesti myös sarjakuvat ja pilapiirroksot, joissa esillä olivat "sedät valkoisissa takeissaan". "Setä" viittasi vanhempaan, arvokkaasti pukeutuvaan, silmälasipäiseen ja solmiolla varustautuneeseen mieheen. Solmio tai rusetti kertoi henkilön kuulumisesta ajattelu- ja hallintotyön piiriin. "Silmälasit taas viittaavat älykkyyteen ja näppäryyteen sekä pehmeuteen, pelkuruuteen, haavoittuvuuteen, nörttiyteen ja feminiinisyyteen suhteessa fyysiseen maskuliinisyyteen, ruumiilliseen voimaan ja kehon täydellisyyteen. Vaikka esimerkiksi yhdysvaltalaisessa kulttuurissa 1960-luku merkitsi silmälasien feminiinisyyden häviämistä ja lasimuodin räjähdysmäistä lisääntymistä, pilapiirrosten tiedemieslasit edustivat 'perinteistä' silmälasikoodistoa." [Suominen 2003.]

Suominen [2003] korostaa, että yleisempi tietokonetyön karrikointi perustui populäärikulttuurin käsitykseen tiedemiehistä ja tietokoneista, mutta se sai vaikutteita myös uutiskuvien tavoista esittää naisten ja miesten töiden eroja sekä alan tarkkaavaisuutta vaativaa luonnetta. Eräs tapa tuottaa asiantuntijuuden statusta oli huolellinen pukeutuminen, joka oli erittäin leimallista tietotekniikan alalla. Myös alan juhlallisuus ja erityisyys tulevat esille. [Suominen 2003.]

Kuten yllä mainittiin, Valitut Palat oli merkittävä vaikuttaja, mitä tuli tietokonetyöläisen perustyyppittelyihin. Suuntaus oli rutiininomaisesta työstä kohti asiantuntija- ja suunnittelutehtäviä. Valitut Palat maalasi myös kuvan, jossa

yksi ihminen – mies – pystyisi huolehtimaan kokonaisista tuotantolaitoksista. Myös ohjelmoijan, ”ohjelmoitsijan”, ammattikunta tuli esille Valittujen Palojen esittelyissä. ”Tämä työ oli harvoja, joista Valitut Palat kertoi sen sopivan sekä miehille että naisille. Lehden mukaan ohjelmointi oli arvostettua työtä, jonka piirissä naiset olivat yhtä kyvykkäitä kuin miehet. -- Työ kuvattiin samanlaisesti sekä intohimoa ja fanaattisuutta mutta myös tarkkuutta ja johdonmukaisuutta vaativaksi. -- Valittujen Palojen ohjelmointityöstä kertova juttu niin ikään viittasi siihen, kuinka naiset voisivat lapsen synnytettyään palata ohjelmointityöhön ainakin osa-aikaisesti tai tehdä töitä kotoa käsin. Yksi sankarillinen ohjelmoija oli jopa viimeistellyt ohjelmakorjauksia synnytyslaitoksella. Naisia tarkasteltiin siis suhteessa perheeseen ja perinteisiin roolikäsityksiin ohjelmoinnin asiantuntijoinakin.” [Suominen 2003.]

Ohjelmointi- ja operointityöt olivatkin ns. atk-ammattien keskiluokkaa, joissa Suomisen [2003] mukaan oli eniten neuvottelun varaa myös asiantuntijuuden ja sukupuolen merkityksestä. ”Ohjelmoijan sukupuoli oli – kiistanalaisella rajavyöhykkeellä suhteessa asiantuntijuuden asteeseen, hegemoniseen maskuliinisuuteen tai vallitsevaan feminiinisyyden kuvaan. Ei ole sattumaa, että juuri operointi- ja ohjelmointityöt ovat keskeisiä, kun sittemmin erityisesti tietokonehackerismia ja nörttiyttä on määritetty. Vaikka nörtti määrittyikin tavallisimmin miespuoliseksi, hänen toimintansa perusteet lepäävät feminiinissä työkuulttuurissa. Siinä on vaadittu rutiininomaista omistautumista, intohimoa ja keskittymistä koneläheiseen työskentelyyn, ei niinkään fyysisistä voimaa tai avointa suhtautumista koneiden ulkopuoliseen maailmaan. – Tietotekniikan hegemoninen maskuliinisuus jakaantuukin useisiin sukupolviin tai vaiheisiin. 1950-luvun reikäkorttiosaston käsityöläisen, kikkataiteilijan ideaali oli niistä yksi, 1960-luvun systeemiasiantuntijan ideaali toinen, 1970-luvun voimistunut hakkeri-ideaali kolmas ja 1980-luvun voimistunut tietokoneharrastajan ihanne neljäs.” [Suominen 2003.]

8. Atk-asiantuntijuuden sukupuoli nykypäivänä

Marja Vehviläinen [1997b] toteaa, miten on tärkeää etsiä sosiaalisia prosesseja, joissa sukupuoli kulttuurisesti muotoutuu, tässä tapauksessa tietotekniikan alalla. Vehviläinen [1997b] kuvailee, kuinka sukupuoli nivoutuu tietotekniikan kaikkiin kolmeen osa-alueeseen: 1) artefakteihin (ohjelmiin, laitteisiin, tietoverkkoihin ja viesteihin), 2) tietämykseen ja 3) sosiaalisesti järjestäytyneeseen toimintaan, eli käytäntöihin. ”Artefakteja, joita voidaan ajatella myös tekstinä, luodaan ja käytetään sosiaalisissa käytännöissä, ja sekä tietotekniikan luomisen että käyttämisen käytännöt edellyttävät tietämystä. Sukupuoli nivoutuu näihin kaikkiin sosiaalisten käytäntöjen kautta.” [Vehviläinen 1997b.]

Tarja Tiainen [2002] tarkastelee alan sukupuolittuneita rakenteita kolmelta näkökulmalta, tietotekniikan ammattilaisten työnjaon, tietotekniikan sukupuolisten symbolien sekä tietotekniikan ammattilaisten sukupuolisen identiteetin ja käytöksen kannalta. Alalla vallitsevaa työnjakoa kuvaavat mm. Tietotekniikan liiton jäsenmäärät, jotka kertovat selvästi miesten dominanssista alalla. Lisäksi on tosiasia, että naiset etenevät alalla miehiä heikommin. Myös sukupuoliset symbolit esittävät (tieto)teknologian maskuliinisena. Sukupuolinen identiteetti ja käytös tulevat ilmi monella tavoin, esimerkiksi historian kirjoituksena, joka päättäväisesti sulkee pois alan naispuoliset merkkihenkilöt, sekä alan lehtien kirjoittajissa, jotka ovat käytännössä kaikki miehiä. Ilmiö tulee hyvin esiin myös Tiainen [2002] tutkimuksessa, jossa hän haastatteli 24 atk-ammattilaista ja tarkasteli heidän käsityksiään sukupuolirooleista. Atk-ammattilaisten käsitysten mukaisesti "äitiys, ja yleensäkin naiseus, on sopimaton ominaisuus atk-asiantuntijalle. Naiset sopivat atk-asiantuntijoiksi vain, mikäli he käyttäytyvät kuin miehet." [Tiainen 2002.]

Saman on havainnut myös Vehviläinen [1994a] tutkiessaan tietotekniikan alan eettistä koodistoa. Hän toteaa, että nainen hyväksytään vain jos hän ei tuo naiseuttaan esiin ja suostuu olemaan "yksi jätkistä". Vehviläinen [1994b] tarkentaa, että nainen, vaikka tietotekniikan alan ammattilaisena, elää naisen ruumiissa, naisena sukupuolittuneessa yhteisössä, omaksuu miesten kehittämät mallit ja ajattelutavat. Tiainen [2002] kuvaa ilmiötä yksinäisten, objektiivisten totuuksien maailmana, jossa kaikki erilaisuus suljetaan pois.

Tiainen [2002] mukaan tietotekniikan miehisestä luonteesta koituu monenlaista haittaa, esimerkiksi tietotekniikkaa käyttävät naiset kärsivät alan miehisydestä, sillä järjestelmien suunnittelijat ovat miehiä, eivätkä ymmärrä järjestelmien käyttäjien (naisten) maailmaa. Alan miehisyys vaikuttaa myös siten, että naiset eivät hakeudu alalle, vaikka he olisivatkin sille sopivia, [Tiainen 2002]. Näin on todellisuudessa käynytkin, sillä 1990-luvulla naisten osuus alan asiantuntijoista kääntyi laskuun [Vehviläinen 1997b].

9. Mahdollisuuksia ja motiiveja muutokseen

Tietotekniikan maailma on miesten maailma, jossa on yksi ainoa totuus. Luopumalla tästä käytännöstä ja sallimalla moniääniset keskustelut, erilaiset käsitykset ja näkökulmat annettaisiin tilaa myös naisten tietämykselle, käytännöille ja asiantuntijuudelle. Samoin myös iän tai erilaisen kokemustaan tuomat särvät pitäisi hyväksyä, ja siten voitaisiin saavuttaa syvempää ymmärrystä. [Tiainen 2002.]

Vehviläinen [1996] puolestaan korostaa, että toisin kuin vielä 1960- ja 1970-luvuilla, tilaisuuden tarjoaminen ei enää nykypäivänä riitä naisten saamiseksi

alalle. Päinvastoin, 1990-luvulla naiset ovat tietoisesti pysytelleet poissa teknisiltä aloilta. Muutoksen onkin lähdeittävä sukupuolittumisen prosesseista. Maailma ilman naisia – perinne olisi murrettava ja luotava tilaa naisten osallistumiselle. Myös miehisyyden ja asiantuntemuksen sidos olisi purettava, ja mahdollistaa naisen identiteetin ja tietotekniikan yhteydet. Identiteetti on monen asian summa, ja juuri moninaisille identiteeteille olisi tehtävä tilaa: ”Naisen tietotekniikan asiantuntijuuden ei pitäisi merkitä naiseuden kieltämistä vaan aitoa vuorovaikutusta teknisen maailman kanssa.” [Vehviläinen 1996.]

Muutoksen tulisi olla kaikenkattava, eli sisältää sekä identiteetin, symbolisen ja kulttuurisen alan että organisaatioiden, instituutioiden ja kotien tilanteet ja käytännöt. Tämä vaatii suuria ponnistuksia niin institutionaalisella, organisatorisella, alueellisella, kansallisella kuin kansainväliselläkin tasolla. [Vehviläinen 1997b.]

Tunnetusti tilanne on samantyyppinen monissa muissa maissa, ja uusi-seelantilaistutkijat Selby ja muut [1998] ehdottavatkin aivan konkreettisia toimia tietotekniikan alan profiilin nostattamiseksi naisnäkökulmasta katsoen. He painottavat, että naisten on näytävä alalla enemmän, tietokoneiden myyjinä, systeemisuunnittelijoina, alan johtotehtävissä ja alan julkaisujen sankarittarina. Tyttöjen tulisi saada onnistumisen kokemuksia tietotekniikan alalta jo koulussa, ja heillä tulisi olla samanlaiset mahdollisuudet käyttää ja harrastaa tietotekniikkaa kuin pojillakin [Selby et al. 1998].

Naiset toisivat alalle myös naisellisia kykyjä ja sitä kautta hyötyjä. Esimerkiksi von Hellens ja muut [2000] korostavat, kuinka naisten kykyä liikkua kykykenttensä sosiaalisesta ääripäästä tekniseen ääripäähän tulisi hyödyntää esimerkiksi suunniteltaessa tietojärjestelmiä yhdessä asiakkaiden kanssa.

10. Päätelmät

Yllä kuvattua taustaa vasten on ymmärrettävää, kuinka tietotekniikan alan sukupuolittuneet käytännöt ovat nykypäivänä sitä mitä ne ovat. Lopputulos on siihen vaikuttaneiden asioidensa summa, mikä Suomessa tietotekniikan alalla on merkinnyt naisia sivuuttavia käytäntöjä.

Näin ei kuitenkaan tarvitsisi, eikä saisi olla. Naisnäkökulma tulisi nähdä rikkautena, joka voisi luoda aivan uusia menestystekijöitä tietotekniikan alalle.

Viiteluettelo

[Kinnunen ja Korvajärvi, 1996] Merja Kinnunen ja Päivi Korvajärvi, Johdanto: Naiset ja miehet työelämässä. Teoksessa Merja Kinnunen ja Päivi Korva-

- järvi (toim.), *Työelämän sukupuolistavat käytännöt*, Vastapaino, Tampere, 1996, 9-19.
- [Kolehmainen, 1999] Sirpa Kolehmainen, *Naisten ja miesten työt, Työmarkkinoiden segregoituminen Suomessa 1970-1990*. Tutkimuksia 227, Tilastokeskus, Helsinki, 1999.
- [Kolehmainen-Lindén, 1998] Sirpa Kolehmainen-Lindén, Sukupuolen mukainen työnjako ja ammatit. *Työelämän tutkimus* 3, 1998, 14-15.
- [Korvajärvi, 1998] Päivi Korvajärvi, Sukupuolistamisen paradoksit työpaikoilla. *Naistutkimus - Kvinnoforskning* 11, 3, 1998, 54-58.
- [Korvajärvi ja Kinnunen, 1996] Päivi Korvajärvi ja Merja Kinnunen, Lopuksi: Työelämän sukupuolistaminen. Teoksessa Merja Kinnunen ja Päivi Korvajärvi (toim.), *Työelämän sukupuolistavat käytännöt*, Vastapaino, Tampere, 1996, 233-240.
- [Lavikka, 1997] Riitta Lavikka, Kulttuuri ja sukupuoli työelämän murroksessa. *Naistutkimus - Kvinnoforskning* 10, 3, 1997, 52-57.
- [Selby et al., 1998] Linda Selby, Ken Ryba and Allison Young, Women in computing: what does the data show? *ACM SIGCSE Bulletin* 30, 4 (December 1998), 62a-67a.
- [Suominen, 2003] Jaakko Suominen, *Koneen kokemus*. Vastapaino, Tampere, 2003.
- [Tiainen, 2002] Tarja Tiainen, Piilotettu maskuliinisuus atk-asiantuntijuudessa. Teoksessa Ilkka Pirttilä ja Susan Eriksson (toim.), *Asiantuntijoiden areenat*, Sphi, Jyväskylä, 2002, 119-136.
- [von Hellens et al., 2000] L. von Hellens, R. Pringle, S.H. Nielsen and A. Greenhill, People, business and IT skills: the perspective of women in the IT industry. In: *Proc. of the 2000 ACM SIGCPR Conference on Computer Personnel Research*, 152-157.
- [Vehviläinen 1994a] Marja Vehviläinen, Reading computing professionals' codes of ethics – a standpoint of finnish office workers. In: Ewa Gunnarsson and Lena Trojer (eds.), *Feminist Voices on Gender, Technology and Ethics*, Centre for Women's Studies, Lulea University of Technology, Lulea 1994, 145-161.
- [Vehviläinen 1994b] Marja Vehviläinen, Living throug the boundariest of information systems expertise – a Work History of Finnish Woman System Developer. In: A. Adam, J. Emms, E. Green and J Owen (eds.), *Proceedings of the IFIP TC9 WG9.1 Fifth International Conference on Women, Work and Computerization: Breaking New Boundaries – Building New Forms*, Manchester 2-5 July 1994, Elsevier Amsterdam, 107-120.

- [Vehviläinen 1996] Marja Vehviläinen, „Maailmoista ilman naisia“ tietotekniikan sukupuolieroihin. Teoksessa Merja Kinnunen ja Päivi Korvajärvi (toim.), *Työelämän sukupuolistavat käytännöt*, Vastapaino, Tampere, 1996, 143-170.
- [Vehviläinen 1997a] Marja Vehviläinen, Gender, expertise and information technology. Ph. D. Thesis, University of Tampere, Dept. of Computer Science **A-1997-1**, February 1997.
- [Vehviläinen 1997b] Marja Vehviläinen, Tietotekniikan sukupuolesta. Teoksessa Kari Stachon (toim.), *Näkökulmia tietoyhteiskuntaan*, Gaudeamus, Helsinki, 1997, 153-170.

Tietohallintostrategian kehitys 2000-luvulle

Antti Rikkinen

Tiivistelmä.

Tutkielmassa tarkastellaan tietohallintostrategian kehitystä automaattisen tietojenkäsittelyn varhaisimmista vaiheista sen nykyisiin suuntauksiin. Tutkielman olennaisin sisältö on tietohallintostrategian tutkimuksen luokittelun laajentaminen ja suhteutus tietojenkäsittelyn hyödyntämisen kehitykseen.

Avainsanat ja -sanonnat: Tietohallinto, tietohallintostrategia.

CR-luokat: K.2, K.6

1. Johdanto

Jo 1970-luvulta lähtien on oltu erittäin kiinnostuneita tietotekniikan ja organisaatioiden toiminnan yhteisestä kehityksestä ja tietotekniikan tuomista mahdollisuuksista. Tietotekniikan vuosikymmenten kehityksen mukana painopiste on siirtynyt tekniikasta sisältöön kuitenkin teknologiaintensiivisyyden säilyttäen. Tietotekniikan ala ei voi jatkuvien uusien innovaatioiden vuoksi saavuttaa täyttä kypsyyttä, joten myös organisaatioiden on pystyttävä jatkuvasti mukautumaan.

Organisaatioihin on kasvanut ja integroitunut funktio, jonka rooli ja osamisala on nopeiden muutosten alainen: tietohallinto. Tietohallinto ohjaa organisaation tietoresursseja ja tekee sen menestyksen kannalta kriittisiä päätöksiä. Siksi on tärkeää perustaa päätökset mahdollisimman vankalle pohjalle. Organisaatioiden päätöstenteon kulmakivi on toimintasuunnitelma eli strategia. Strategian tarkoitus on tukea organisaation toimintaa tehostamalla kokonaisvaltaisesti tietoresurssien hyödyntämistä ja pyrkiä kohti organisaation tavoittelemaa päämäärää tietoresurssien käytön ja saatavuuden suhteen. Koska tietohallinto on aina vain enemmän nivoutunut yhteen organisaatioiden kokonaistoiminnan kanssa, myös tietohallintostrategia kuuluu kiinteäksi osaksi organisaation kokonaisstrategiaa. Tutkimuksessa käsitellään tietohallintostrategiaa liittämättä sitä liikaa mihinkään yksittäiseen sovellusalaan. Historiallisen näkökulman tavoitteena on luoda pohjaa tulevaisuuden kehityksen arvioinnille. Tutkimuksessa yhdistetään tietojenkäsittelyn aikakaudet Sabherwalin, Hirschheimin ja Golesin tietohallintostrategian teoreettisen tutkimuksen luokitteluun [2003] ja laajennetaan luokittelun tulkintaa.

Luvussa 2 määritellään tietohallintostrategia esittämällä tietohallinnon ja strategian perusmääritelmiä, ja esitetään tietohallintostrategian perusajatukset. Luvuissa 3 ja 4 tarkastellaan tietojenkäsittelyn hyödyntämisen kehitystä ja tietohallintostrategian muutosta 1900-luvulta 2000-luvulle. Luvussa 5 luokitellaan tietohallintostrategian kehitysvaiheita ja suuntauksia, ja arvioidaan tärkeimpiä siirtymiä.

2. Mikä on tietohallintostrategia?

2.1. Tietohallinto

Tietohallinnolla tarkoitetaan organisaation tietojärjestelmien suunnittelusta ja ylläpidosta vastaavaa yksikköä. Tälle organisaatioyksikölle on myös useita muita käytössä olevia nimiä, kuten tietotekniikka- tai atk-osasto, joilla viitataan samaan käsitteeseen. Ruohonen ja Salmela [1999, s. 123] määrittelevät tietohallinnon lyhyesti *johton tehtäväkokonaisuudeksi*. Tämä johtuu heidän mukaansa tietojärjestelmiin liittyvien päätösten hajautumisesta organisaatioissa. Eri yksiköt tekevät itsenäisiä päätöksiä tietojärjestelmiin liittyen. Eräs vakiintunut määritelmä tietohallinnolle on "Tietohallinto on yrityksen tai muun organisaation tietoresurssien hyväksikäytön suunnittelua, johtamista, toteutusta ja valvontaa [Reponen *et al.* 1995]." Tietoresurssit taas ovat Ruohosen ja Salmelan [1999, s. 123] määritelmän mukaan kaikki tieto, mitä yrityksessä on, mukaan lukien henkilöstön omaksuma tieto, jota ei voida nykyisille medioille tallentaa. Earlin [1989] mukaan tietohallinnon johtamisen neljä perustehtävää ovat suunnittelu, organisointi, valvonta ja kehittymisen seuranta.

2.2. Strategia

Strategialle on monia määritelmiä aina antiikin Kreikasta ja sotatieteistä lähtien. Tässä tutkimuksessa viitataan strategiaan organisaatiotieteiden näkökulmasta. Organisaatiolla tarkoitetaan tutkimuksessa kaiken tyyppisiä organisaatioita, ei pelkästään julkisia tai yritysorganisaatioita.

Laukkanen ja Vanhala [1994] määrittelevät strategian organisaation perustehtävää (missio) tavoittelevaksi suunnitelmaksi. Strategia on heidän mukaansa julkilausuma, jossa organisaatio ilmaisee toimintansa onnistumisen kannalta tärkeimmät mittarit, ja jossa yritys virallisesti päättää, mitä toimintamuotoja ja -tapoja harjoitetaan, miten se suhtautuu mm. riskeihin, ja mitä arvoja organisaatio erityisesti kunnioittaa.

Yksi nykyisin tunnetuimmista näkökulmista strategiaan on Porterin [1985] yhdistetty geneerinen kilpailustrategia. Porter jakaa strategiat kolmeen kategoriaan: kustannusjohtajuus, erilaistaminen ja keskittyminen. Porterin [1996] mukaan strategia on yhdistelmä maaleja, joita kohti organisaatio pyrkii, ja

keinoja joita sinne pääsemiseksi käytetään. Vaikka Porterin ajatukset sopivat parhaiten kilpailulliseen strategiaan, etenkin jälkimmäinen osa on yleistettävissä strategian perusajatukseksi. Tietotekniikalla voidaan tukea kaikkia Porterin strategiatyyppejä. Strategia on linkki yleisten menettelytapojen ja käytännön välillä. Vaikka strategiasta on tullut monien mielestä aikamme muotisana, voidaan sen perustella olevan edelleen perusolemukseltaan *johtamisen* tärkeä väline.

2.3. Tietohallintostrategia

Tietohallintostrategialle on monta nimeä, jotka periytyvät suomalaisessa kieliympäristössä tietotekniikan eri aikakausilta, ja joista voidaan jo nimen perusteella nähdä erilaisia sisällöllisiä painoituksia. Atk-strategia painottaa tietojenkäsittelyn rutiineja, tietotekniikkastrategia laajemmin käytettävää tekniikkaa ja tietojärjestelmästrategia varsinaisia sovelluksia [Ruuhonen ja Salmela, 1999, ss. 9-10]. Tietohallintostrategiaksi kehityttyään termi sisältää jo paljon hallinnollisen näkökulman mukanaan tuovia elementtejä. Ruuhosen ja Salmelan mukaan tulevaisuuden (ja jo nykyinen) suuntaus on puhua tietostrategiasta tai tietoresurssi-strategiasta.

Tietohallinto on johdon tehtäväkokonaisuus ja strategia johtamisen työkalu. Tietohallintostrategian avulla päätetään tietojärjestelmien tehtävistä, investoinneista, kehitystoimista ja arvioinneista [Ruuhonen ja Salmela, 1999]. Voidaan päätellä, että myös tietohallintostrategian tavoitteena on organisaation yhteinen julkilausuma, jolla kerrotaan olennainen organisaation tavoitteista ja suunnitelluista toimenpiteistä tietoresurssien käytön suhteen myös niille, jotka eivät ole osallistuneet strategian luomiseen.

Tietohallintostrategiassa määritellään organisaation tekninen ja sovellusarkkitehtuuri ja päätetään sen kehittämistä, mutta sen lisäksi otetaan huomioon organisatoriset, taloudelliset ja johtamisen kysymykset. Strategia on siis hyvin monialainen haaste, joka vaatii tietotekniikan, liiketoiminnan ja organisaation kehityksen yhtäaikaista hallintaa. Ruuhonen ja Salmela [1999, s. 125] tiivistävät:

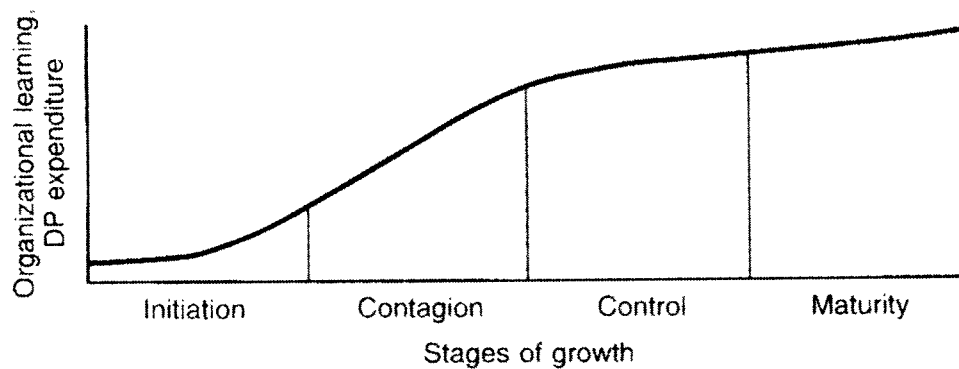
“Kehityksen perusteella voidaan todeta, että tietohallintostrategia onkin perusteltava organisaation rakenteen, sen kulttuurin ja henkilöresurssien tekijöiden avulla.”

Julkisen tietohallinnon alaan vaikuttava suuntaus on tietoyhteiskuntastrategia. Suomen valtiolla on yleinen tietoyhteiskuntaohjelma, jonka tavoitteina voidaan mainita tietotekniikan hyödyntäminen julkishallinnossa, kansalaistaitojen kehittäminen ja tietotyön lisääminen. Myös useat kunnat ovat laatineet tietoyhteiskuntastrategioita. Tietoyhteiskuntastrategiat antavat julkisille tietohallinnoille pääsuuntaviivat ja tavoitteet omien strategioidensa muodostamiseen.

Suomen hallituksen suunnitelmiin kuuluu myös valtion tietohallintolaki, jonka tulisi selkeyttää tietohallintojen vastuuta strategian käytännön toteuttamisesta. [VT, 2005]

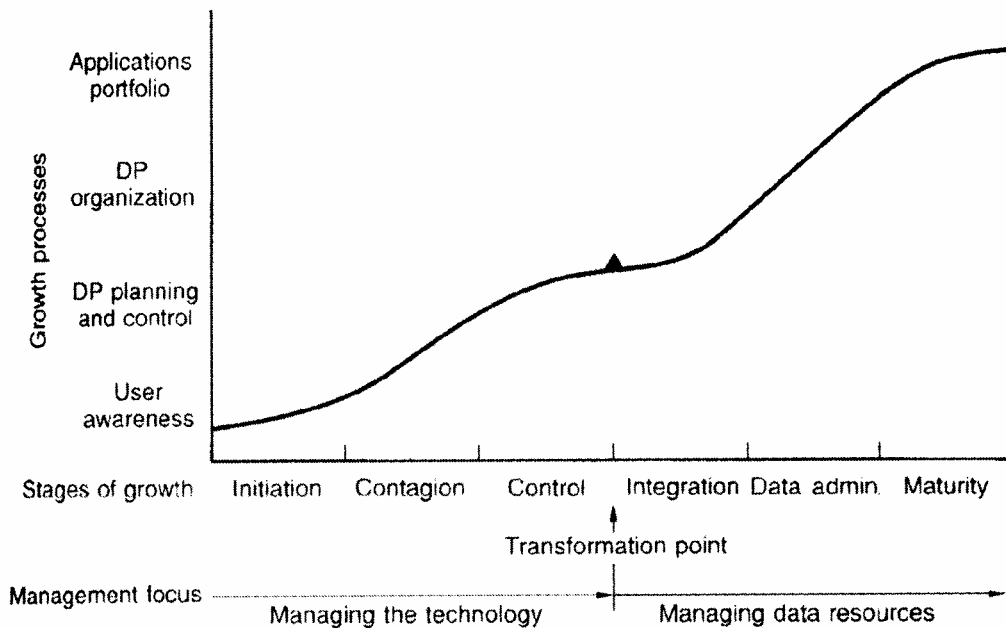
3. Tietojenkäsittelyn kehitys

Tietojenkäsittelyn kehitykselle on esitetty ensin monia vaiheittaisia malleja. Esimerkiksi Nolanin nelivaiheinen ja myöhemmin kuusivaiheinen malli 1970-luvulta kuvasi ensimmäisinä tuolloista näkemystä tietotekniikan hyödyntämisen kehityksestä [Galliers and Sutherland, 2003, pp. 33–35]. Nolanin malli on ollut Suomessakin yleisesti käytössä vielä 1980-luvulla.



Kuva 1. Nolanin nelivaiheinen tietojenkäsittelyn kehityskulku. [Galliers and Sutherland, 2003]

Nelivaiheinen malli ei vielä herättänyt suurta kiinnostusta, koska sen yksinkertaisuus ei tuntunut alan tutkijoiden ja käytännön työtä tekevien mielestä uskottavalta. Myöhemmin Nolan kehitti kuusivaiheisen kehityskulun, jossa otettiin huomioon teknologian ja sisällön kehityksen eriaikaisuus. Tämäkin malli sai osakseen kritiikkiä alan tutkijoilta, mutta käytännön tietotekniikan sovelluksista vastaavien ihmisten mielestä tämä malli vastasi todellisuutta [Ruohonen ja Salmela, 1999]. Molemmat mallit sisältävät kypsän vaiheen, jonka olemassaolo on kyseenalainen.



Kuva 2. Nolanin kuusivaiheinen tietojenkäsittelyn kehityskulku. [Galliers and Sutherland, 2003]

Yleisellä tasolla tietotekniikan hyväksikäytössä organisaatioissa on siirrytty vaiheesta toiseen 1960-luvulta lähtien. Tuolloin tavoitteena oli kustannusten minimointi, ja tietotekniikkaa hyödynnettiin laskukonemaisesti tarkasti rajattuihin tehtäviin. Tätä aikaa kutsutaan nykyisin *atk-ajaksi* (data processing). 1970-luvulla painotukseksi nousi tietojärjestelmien kehitys. Erityisesti johdon tietojärjestelmiä kehitettiin ja pyrittiin kokoamaan informaatiota toiminnan ohjaukseen. Tätä aikaa kutsutaan nykyään *MIS-ajaksi* (Management Information Systems). 1980-luvulla alkoi tietotekniikan vaikutus mm. liiketoimintaan olla huomion keskipisteenä, ja tietotekniikan käytöllä pyrittiin tuottamaan suoraa lisäarvoa. Tätä kutsutaan *SIS-ajaksi* (Strategic Information Systems). [Galliers and Somogyi, 1987; Ruohonen ja Salmela, 1999]

ERA	NATURE OF THE TECHNOLOGY	NATURE OF SYSTEMS OPERATIONS	FOCUS OF SYSTEMS EFFORT	CHARACTERISTICS OF SYSTEMS DEVELOPMENT
DATA PROCESSING	<ul style="list-style-type: none"> • CUMBERSOME MACHINES • LIMITED RELIABILITY • LIMITED INPUT/OUTPUT MEDIA 	<ul style="list-style-type: none"> • CENTRALISED • REMOTE 	<ul style="list-style-type: none"> • PROGRAMMING PROCESS 	<ul style="list-style-type: none"> • EMPIRICAL
MANAGEMENT SERVICES	<ul style="list-style-type: none"> • LARGE MAINFRAMES • OPERATING SYSTEMS • SCREEN-BASED INPUT-OUTPUT • DATABASES • MINICOMPUTERS 	<ul style="list-style-type: none"> • DISTRIBUTED PROCESSING • TRANSACTION DRIVEN OPERATIONS 	<ul style="list-style-type: none"> • SYSTEMS ANALYSIS AND DESIGN • SYSTEMS LIFE CYCLE • INCREASING MAINTENANCE • DATA MANAGEMENT 	<ul style="list-style-type: none"> • ANALYTICAL, STRUCTURED (separate trends for process and data) • PROJECT MANAGEMENT
INFORMATION PROCESSING	<ul style="list-style-type: none"> • CONVERGING TECHNOLOGIES • MINIATURISATION • PACKAGES • PCs/WORKSTATIONS 	<ul style="list-style-type: none"> • USER/BUSINESS DRIVEN • INFORMATION CENTRE • BALANCE BETWEEN DISTRIBUTED/CENTRALISED APPROACH 	<ul style="list-style-type: none"> • INFORMATION MANAGEMENT • COMMUNICATION • END-USER INVOLVEMENT • LINK WITH BUSINESS STRATEGY 	<ul style="list-style-type: none"> • PARTICIPATIVE • END-USER COMPUTING • PROTOTYPING • AUTOMATED TOOLS • IS PLANNING/STRATEGY
TRENDS	Fragmented ↓ Packaged & interconnected; addressing different forms of information (data, voice, text, image)	Unavailable, regulated ↓ Supportive, available	Technical Issues ↓ Business support Strategic issues	Ad hoc ↓ Methodical, social, business related

Kuva 3. Siirtymäilmiön tunnusmerkkejä. *Data Processing* (atk-aika), *Management Services* (MIS-aika), *Information Processing* (SIS-aika). [Galliers and Somogyi, 1987]

Ruuhonen ja Salmela [1999, s. 11] kuvaavat nykyistä aikaa osaamiskeskiksi tietojenkäsittelyksi. Tällä tarkoitetaan osaamisverkkojen johtamista (Knowledge Management Networks, KMN), jossa tietojärjestelmät toimivat organisaation toimijoiden ja prosessien alustana.

KMN-AIKA	OSAAMISVERKKOJEN JOHTAMISEN AIKA
TAVOITTEET	Organisaation osaamisvoimavarojen tunnistaminen ja hyväksikäyttö
ELINKAARI	Jatkuva uusiutuminen mahdollistaa vuosien elinkaaren
TIEDON LUONNE	Sisäistä ja osin ulkoista tietoa organisaation työntekijöissä, prosesseissa ja tietojärjestelmissä
TIEDON LÄHTEET	Intuitiivinen, näkemyksellinen tieto, tietoverkkojen kautta jaettu tulkittu tieto, yhteisesti luotu, sosiaalinen tieto
LOOGISET PROSESSIT	Tutkimusprosessin luonteisia, ristiriitoja sietäviä
KÄYTTÄJÄT	Koko organisaatio ja sen sidosryhmätahot
TEKNIikka	Verkottunut tietojenkäsittely yli organisaatorajojen

Kuva 4. KMN-ajan ominaisuuksia Ruuhosen ja Salmelan [1999] mukaan.

Carr [2003] arvioi it-infrastruktuurin jo saavuttaneen niin sanotun kypsän vaiheen ainakin useimpien liiketoimialojen näkökulmasta. Hänen näkemyksensä ei ole saanut tukea muilta alan tutkijoilta. Carrin päätelmien kritiikkinä voidaan esittää, että tietojenkäsittelyn kaltainen teknologiaintensiivinen ala ei voi saavuttaa varhaisten mallien kaavailemaa kypsää vaihetta aina uusien innovaatioiden muuttaessa rakenteita. Tietojenkäsittelyn käytön kehitystä ei voi Carrin tapaan suoraan verrata esimerkiksi sähkön tai rautateiden käytön kehitykseen, sillä tietojenkäsittely on innovaatioita mahdollistava väline. Teknologiset, kaupalliset ja sosiaaliset innovaatiot synnyttävät uusia strategisia mahdollisuuksia ja vanhat ja nykyiset strategiat joutuvat uudelleen mietittäviksi. Tärkeää on pohtia, voiko tällöin myöskään tietoresurssien johtaminen ja sitä kautta tietohallintostrategia saavuttaa kypsää vaihetta.

4. Strategian muutos

4.1. Atk-aika

Edes atk-strategiasta ei vielä tietotekniikan hyödyntämisen alkuaikoina 1960-luvulla ja 1970-luvun alussa puhuttu strategian osa-alueena, vaan termi on kehitetty myöhemmin kuvaamaan ajan asenteita tietotekniikkaan. Tuolloin tietotekniikka oli käytössä laskimeen verrattavana työkaluna, jota pidettiin takahuoneessa pois päätoimintojen tieltä. Johtamisessa ja organisaatioiden toiminnassa ajateltiin funktionaalisesti, eli esimerkiksi markkinointi, tuotanto ja tietojenkäsittely erotettiin tarkasti ja johtaminen oli henkilö- ja johtamistyyli-vetoista [Järvinen, 2003, s. 19].

4.2. 1970-luku

1970-luvulla keskityttiin informaation keräämiseen ja johdolle raportointiin. Informaatiota käytettiin tuotannonohjaukseen ja muihin tukitoimintoihin aluksi kokonaisjärjestelmin. Tällaisten järjestelmien strategisten tavoitteiden voisi kuvailla olevan melko suoraviivaisen operatiivisia, esimerkiksi tietyn suoritteiden onnistumisen prosenttiosuus. Tällaiset käsittelyjärjestelmät keskittyvät tiedon eli datan varastointiin ja käsittelyyn, ja pystyvät raportoimaan vastuussa olevalle johdolle. MIS-tyyppisten järjestelmien suunnittelun yleisstrategia liittyi sen käytännön palvelevuuteen, eli esimerkiksi olennaisen tiedon jakeluun ja poikkeusraportointiin [Ruohonen ja Salmela, 1999, s.130]. Suunnittelua tehtiin teknologian ja liiketoimintastrategian vuorovaikutuksellisesta näkökulmasta (contingency perspective) [Sabherwal *et al.*, 2003]. Voidaan sanoa, että tällöin ensimmäisen kerran havahduttiin strategiseen suunnitteluun, jossa otetaan yrityksen kokonaisstrategia huomioon. On kuitenkin huomioitava, että tieteilisten viitekehysten luomisesta on kulunut aina jonkin verran aikaa niiden

lopulliseen hyväksyntään ja käytäntöön siirtymiseen. Toisaalta samaan aikaan uskottiin vahvasti vaiheittaisiin malleihin (esim. Nolan), joiden yleispätevyys kaikissa konteksteissa on käytännössä mahdotonta [Sabherwal *et al.*, 2003]. Vaiheittaisia mallejakaan ei ole kuitenkaan täysin unohdettu, vaan vaiheittaisuutta voidaan todeta tarkastetun aina sopivin väliajoin vastaamaan ajankohdaisia vaatimuksia (esim. [Galliers and Leidner, 2003, pp. 34-60]).

4.3. 1980-luku

1980-luvulle tultaessa alettiin todella keskustella tietotekniikan vaikutuksista liiketoimintaan. Strategioiden integrointi tuli ensi kerran vahvasti kuvaan tuona aikana. Aikakausi sai jopa nimekseen strategisten tietojärjestelmien aika. Tuolloin ei kuitenkaan käytännön tietotekniikkastrategiaan liitetty vahvasti hallinnollisia ja sosiaalisia elementtejä, vaan keskityttiin siihen, millaista suoraa lisäarvoa tietotekniikan avulla voidaan saavuttaa (esimerkiksi [Earl 1989]), ja strategiat todennäköisesti siksi olivat vain liikeyritysten kiinnostuksen kohteina. Monet tietohallinnon nykyisin noudatettavat strategiset periaatteet luotiin tänä aikana. Esimerkiksi Earlin [1989] määritelmät tietohallinnon tehtävistä ovat edelleen alan perusmääritelmiä.

Tietojärjestelmien kasvu organisaatioiden toiminnan sisälle aiheutti osaltaan johtamistavan muutoksen, ja 1980-lukua pidetäänkin integroivan johtamistavan alkuna [Järvinen, 2003]. Samalla tietotekniikka oli jo integroitunut suureen osaan liiketoimintaprosesseja, ja se alkoi integroitua osaksi organisaatioita. Aikakauden loppupuolella alettiin selvästi pohtia myös tietotekniikan soveltamisen sosiaalisia näkökohtia (esim. [Kemp and Clegg 1987]).

1980-luvulla myös yritysstrategia sai uusia näkökulmia muun muassa Porterin [1985] ansiosta. Porter jakoi strategiat kolmeen kategoriaan: kustannusjohtajuus, erilaistaminen ja keskittyminen. Porterin ajatukset sopivat tietotekniikan soveltamiseen, ja koska Porterin teorioiden vaikutus oli voimakas yritysten ajatteluun, teknologiaa alettiin soveltaa hänen strategioihinsa.

4.4. 1990-luku

1990-luvulla etsittiin mahdollisuuksia uudelleenorganisointumiseen ja integroiva ajattelu ulottui myös organisaatioiden prosesseihin ja tietojärjestelmiin. Tietotekniikan käytössä havaittiin niin sanottu tuottavuusparadoksi, sillä tutkimukset osoittivat, että lisääntynyt teknologian määrä ei lisännyt tuottavuutta tarkoituksen mukaisessa suhteessa [Ruuhonen ja Salmela, 1999]. Gartner Group julkaisi 1996 tutkimuksen, jonka tuloksena arvio tietotekniikkaan sijoitetun pääoman tuotosta 1985–1995 oli vain yksi prosentti [Kettunen ja Simons, 2001, s. 192]. It-funktion tuottavuuden arviointi ja kehitys on ollut strategioiden huomion kohteena 1990- ja 2000-luvuilla (esim. [Melville *et al.*, 2004]). Tämä on

johtanut tietohallinnon toiminnan alasajoihin ja ulkoistuksiin monissa organisaatioissa.

Tietotekniikkaan tehdyt massiiviset investoinnit saivat tutkijat pohtimaan teknologian roolia organisaatioissa. Swanson [1994] kehitti innovaatioteorian, jolla voidaan luokitella it-funktioon liittyviä innovaatioita hallinnon, teknologian ja prosessien kannalta. Voidaan sanoa, että tällöin on ryhdytty pohtimaan tietotekniikan kokonaiskontribuutiota organisaation toimintaan. Aiemmin tällaista mittaavassa mielessä konvergoivaa lähestymistapaa, jonka perusteella yritykset voivat esittää itselleen kysymyksiä innovaation ja teknologian suhteesta organisaatiossaan, ei ole esitetty.

Integroiva ajattelu ja hajautunut globalisoituva liiketoiminta on johtanut viimeisen kymmenen vuoden aikana ERP-järjestelmien (Enterprise Resource Planning, toiminnanohjausjärjestelmä) suureen suosioon. Toiminnanohjausjärjestelmät toivat uusia mahdollisuuksia strategiseen suunnitteluun, sillä niiden avulla voidaan tukea organisaation kokonaisstrategiaa, tai ne voivat tuoda aivan uusia strategisia mahdollisuuksia. ERP-järjestelmät ovat myös olleet yhtä aikaa vastaus sekä kustannuksellisiin paineisiin että tiedon ja järjestelmäarkkitehtuurien yhteensovittamiseen. [Davenport, 2000]

ERP-järjestelmien vaikutus tietohallintoihin on kaksijakoinen. Toisaalta niiden tarkoitus on poistaa järjestelmien välistä redundanssia, ja siten niiden vaikutus tietohallintoon voi olla alasajava. Toisaalta tietohallinnon tehtäväkenttä voi muuttua toiminnanohjausjärjestelmän myötä, jos sen rooleihin kuuluu implementaatioon liittyviä tehtäviä. Davenportin [2000, p. 190] mukaan esimerkiksi tietohallinnon tehtäväkenttä useimmiten laajenee toiminnanohjausjärjestelmien myötä.

Toiminnanohjausjärjestelmien lisäksi toinen nimittävä tekijä strategioiden muutokselle on ollut internet. Sähköisen liiketoiminnan mahdollisuudet saivat jopa kuplaksi muodostunutta huomiota, mutta yhdessä entistä verkostoituneemman liiketoiminnan ja tehokkuutta tarjoavien toiminnanohjausjärjestelmien kanssa internet on kasvanut 1990-luvulla yhdeksi strategioiden kulmakiviksi. Internet-strategiat ovat olleet monille organisaatioille myös kompastuskivi liian suurien tai vääristyneiden odotusten vuoksi. Porter [2001] toteaa monien yritysten hairahtuneen kustannustehokkuuden etsimiseen unohtaen etsiä markkinoilla asemaa, jossa he voivat olla ainutlaatuisia. Porter toteaa viiden kilpailuvoiman ja arvoketjumallin edelleen pätevän, vaikkakin arvoketjussa internet voi korvata joitakin toimintoja. Myös Agarwal ja Sambamurthy [2002] korostavat informaatioteknologian aseman arviointia strategisen edun tuottajana (strategic differentiator) organisaatioissa.

4.5. 2000-luku

2000-luvun tietohallintostrategian voidaan sanoa olevan suoraa jatkoa edelliselle vuosikymmenelle. Monet 1990-luvulla suuria harppauksia enteilleet muutokset ovat realisoituneet tai realisoitumassa. Esimerkiksi internet ja globaali verkostoituminen ovat kasvaneet täysin eri mittakaavaan kymmenessä vuodessa. 2000-luvulla mobiilikausi ja median sisällön digitalisoituminen on nousnut seuraavaksi varmaksi suuntaukseksi. Toisaalta niin sanotun it-kuplan puhkeaminen on johtanut edelleen kovempiin tietotekniikan tuottovaatimuksiin. Tiedon jalostaminen tietämykseksi ja tietämyksen hallinta (mukaan lukien tietoturva) on eräs nykyisten organisaatioiden ajankohtaisista haasteista.

Globalisaatio on ollut määräävä trendi 2000-luvun tietojärjestelmien, organisaatioiden, liiketoiminnan ja tietohallintostrategian kannalta. Uudet haasteet ovat luoneet tarpeen globaalien tietojärjestelmien (Global Information Systems, GIS) hallintostrategioille. Karimin ja Konsynskin [2003, p. 108] mukaan globaalien tietojärjestelmien hallintostrategian tulisi sisältää (1) keskitetyn ja/tai koordinoitun liiketoiminta-/teknologiastrategia kommunikatioinfrastruktuurin, arkkitehtuurin ja standardien rakentamiseksi, (2) keskitetyn ja/tai koordinoitun tiedonhallinnan strategian yritystietokantojen suunnittelemiseksi ja (3) globaalien liiketoiminnan ja globaalien tietojärjestelmien hallintostrategian yhteensovituksen. Tulevaisuuden haasteina Karimi ja Konsynski eivät pidä it-infrastruktuureja, vaan näkevät tietämyksen muodostamisen ja hyödyntämisen globaalissa mittakaavassa tärkeimpänä tietohallintostrategian tavoitteena organisaatioille. Itse tietohallinnon organisointiin ei kuitenkaan ole yhtä oikeaa tapaa, näin toteavat päätelmissään myös Agarwal ja Sambamurthy [2002] tutkimuksessaan erilaisista tietohallintofunktion organisointimalleista.

Turban olettaa kaikkien organisaatioiden siirtyvän sähköiseen liiketoimintaan ja määrittelee [Turban *et al.*, 2002, p. 677] nykyaikaisen organisaation strategian tasoiksi (1) yritysstrategian (corporate strategy), (2) tietohallintostrategian (IT strategy), (3) sähköisen liiketoiminnan strategian (EC strategy) ja (4) sähköisen liiketoiminnan funktionaaliset strategiat (EC functional strategies). Turbanin mukaan sähköisen liiketoiminnan strategia on alisteinen tietohallintostrategialle tai hyvin läheisessä relaatiossa siihen. Pääsyyt tähän ovat yhteinen infrastruktuuri ja pakollinen integraatio muun muassa toiminnanohjausjärjestelmän kanssa. Myös organisaationaalisesti tietohallinto ja sähköinen liiketoiminta linkittyvät vahvasti tai jopa ovat henkilöityneinä samoihin ihmisiin. Voidaankin todeta, että tietohallintostrategia on saanut lisäksi eräänlaisia alistrategioita, joilla täydennetään pelkän yleisen tietohallintostrategian sisältöä, koska käsite on laajentunut ajan saatossa vaikeasti hahmotettavaksi.

2000-luvun näkemykseksi tietotekniikan optimaalisesta hyödyntämisestä strategiassa on noussut onnistunut päätöksenteko tietämykseksi jalostetun informaation pohjalta. Yrityksissä tätä kutsutaan liiketoimintatiedon hallinnaksi (Business Intelligence, BI). Rouibah ja Ould-ali [2002] määrittelevät liiketoimintatiedon hallinnan seuraavasti:

“Business Intelligence (BI) is a strategic approach for systematically targeting, tracking, communicating and transforming relevant weak signs into actionable information on which strategic decision-making is based.”

Business intelligence on monialainen käsite, joka on kehittynyt markkina- ja kilpailutiedon keräämisestä Knowledge Management -suuntaukseen. BI:n olennainen sisältö on kuitenkin strateginen käyttö päätöksenteossa.

5. Tietohallintostrategian tutkimuksen tyypit

5.1. Elämänkaariteoriat

Sabherwal ja muut [2003, p. 313] lajittelevat strategisen tietojärjestelmä- ja tietohallintojohtamisen tutkimukset kolmeen teoreettiseen koulukuntaan, ja tarjoavat omaa teoreettista malliaan neljänneksi vaihtoehdoksi. *Elämänkaariteoriat* (life-cycle theories) ottavat tietohallintojohtamiseen muutosjohteisen näkemyksen. Niiden olettamuksiin kuuluu, että muutokset organisaatioiden tietojenkäsittelyssä tapahtuvat samojen vaiheiden mukaisesti, ja niillä on samanlainen päämäärä. Esimerkkinä Sabherwal ja muut [2003] käyttävät Nolanin mallia, jossa tavoitteena on kypsä vaihe. Elämänkaariteorioiden ongelma on kontekstin merkityksen väheksyminen, sillä käytännössä esimerkiksi yksittäisen tietohallinnon kehitys ei todennäköisesti täydellisesti noudata vaiheittaista mallia, vaan kehitys etenee osittain eri tahtiin. Malleissa ei myöskään koskaan palata takaisin päin, mikä todellisuudessa on täysin mahdollista, ja niissä on aina jokin päämäärä, johon vääjäämättä pyritään. Tietojenkäsittelyn kehitys on kuitenkin osoittanut, että päämäärän häämöttäessä se todellisuudessa siirtyy aina eteenpäin uusien innovaatiomahdollisuuksien myötä. Vaiheittaisen mallin esittäminen voi johtaa myös usein kannanottoihin hyvästä johtamistavasta ([Galliers and Sutherland, 2003]), mikä lisää malliin tulkinnallisen elementin, joiden yhteisvaikutus on edelleen vaikeammin arvioitavissa. Elämänkaariteoriat ovat varhaisimpia teorioita strategian kehityksestä, mutta niidenkin käyttö on jatkunut ajan vaatimien päivitysten myötä.

5.2. Universalistiset teorit

Alan kirjallisuudessa on paljon *universalistisia teorioita* (universalistic theories), jotka kuvaavat yhtä tapaa johtaa tietohallintoa ja parantaa sen toimintaa. Vaikkakin universalistiset teorit tuottavat arvokasta tietoa tietohallintojohtamisesta ja sen vaikutuksista, voidaan niiden heikkoutena pitää teorioiden yleistystä pätevyyteen kaikissa konteksteissa. Universalististen teorioiden todellinen sovellusala saattaa olla verrattain kapea. Esimerkkinä universalistisesta teoriasta voidaan pitää Karimin ja Konsynskin [2003] mallia, joka esittää yleistäviä näkemyksiä globaalin tietojärjestelmästrategian kehittämisestä. Universalistisia teorioita kehitetäänkin usein uusille alueille, joita ei yleisesti vielä tunneta kovin täydellisesti. Tällaisen teorian suurin ansio on suunnan näyttäminen, jolloin tutkittavasta alueesta voidaan saada tarkempia tutkimustuloksia universalististen teorioiden kontekstissa. Universalistisia teorioita on siis syntynyt aina uusien innovaatioiden myötä siitä lähtien, kun tietotekniikkaa on pidetty merkityksellisenä organisaatioiden vaikuttajana.

5.3. Yhteistoiminnallinen näkökulma

Yhteistoiminnallinen näkökulma (contingency perspective) vastoin universalistista teoriaa tutkii ympäristöllisen, organisaation ja tietojärjestelmällisen kontekstin vaikutusta tietohallintoon tai tietohallinnon ja liiketoiminnan yhteensovittamista. Tämän tyyppinen tutkimussuunta pyrkii osoittamaan liiketoiminnallisen ja tietojärjestelmällisen yhteensovittamisen tulokselliset edut. Yhteistoiminnallisen näkökulman vahvuus on kontekstin tunnustaminen, vaikkakin tutkimukset ovat yleensä aikaan sidottuja ja niiden toimintamallit yleensä pätevät vain lyhyellä aikavälillä. Voimme helposti todeta tietohallinnon haasteiden muuttuneen huomattavasti jo verrattain lyhyellä aikavälillä, joten myös strategioiden voidaan olettaa tarvitsevan muutoksia ajan kuluessa. Yhteistoiminnallisesta näkökulmasta tehdyllä tutkimuksella saadaan universalistisia teorioita konkreettisempia ja ajankohtaisempia tutkimustuloksia, mutta niiden käytettävyys on ajallisesti rajoittunut. Esimerkiksi Agarwal ja Sambamurthy [2002] pyrkivät tutkimuksessaan kasaamaan mallin it-funktion rakentamiselle liiketoimintastrategian tukemana, vaikka päätelmissään toteavatkin yksittäisen oikean organisointitavan mahdollisuuden. Tämä strategiasuuntaus on tullut esiin SIS-ajan myötä liiketoiminnallisen hyödyntämisen tutkimuksissa.

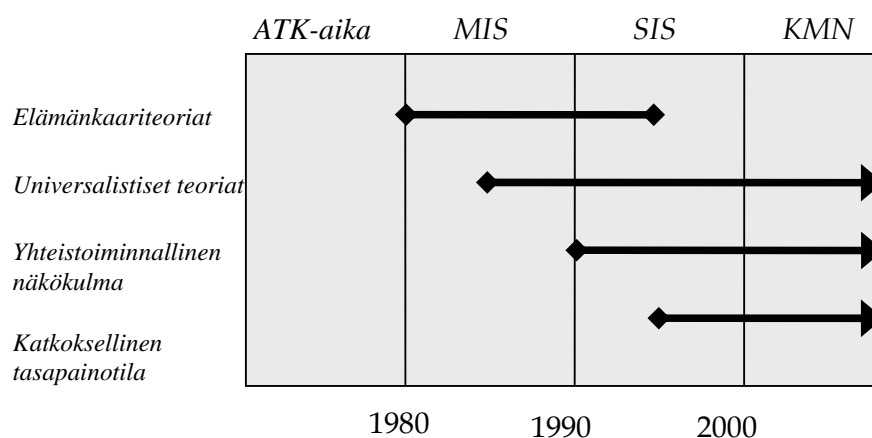
5.4. Katkoksellinen tasapainotila

Darvinististen vaiheittaisten (elämänkaari) mallien lisäksi Sabherwal ja muut löytävät *katkoksellisen tasapainotilan* mallit (punctuated equilibrium model), jotka poikkeavat vaiheittaisista malleista esittämällä, että hidas vaiheittainen kasvu todellisuudessa muuttuu aina äkillisten murroksellisten muutosvaihei-

den myötä. Katkoksellisen tasapainotilan malli on avoin erilaisille kausille tietohallintojohtamisen kulttuurissa, ja se tiedostaa murroksen nopean kehityksen vaiheet. Toisin sanoen se pyrkii välttämään samaan aikaan universalististen teorioiden yksipuolisen ajattelutavan ja yhteistoiminnallisen näkemyksen olemuksen stabiilisudesta. Myös elämänkaariteorioiden pakollinen vaiheittaisuus puuttuu. Mallin hyvä puoli on sen sovitettavuus eri konteksteihin eri organisaatioissa. Sen avulla voidaan luoda strategisia skenaarioita, joihin reagoitua voidaan pohtia etukäteen. Katkoksellisen tasapainotilan mallia ei toisaalta sen vuoksi voi yleistää universalististen teorioiden tapaan. Parhaiten tällaista mallia voi soveltaa yksittäisen organisaation strategian ja tietojärjestelmien hyödyntämisen historialliseen tarkasteluun ja arviointiin. Historiallisen kontekstin tiedostaminen voi olla kuitenkin merkittävä apu tulevaisuuden strategian muodostamisessa. Katkoksellisen tasapainotilan malli ei ole samalla tavalla yleiseen tietojenkäsittelyn kehitykseen sidottu kuin muut teoriatyypit. Katkoksellisen tasapainotilan malli on harvinainen ja tuore teoriatyyppejä, jonka hyödyntäminen riippuu siitä, voidaanko teoriaan tuoda historiaa analysoivien ominaisuuksien lisäksi enemmän tulevia toimintamalleja tarjoavia elementtejä.

5.5. Tietohallintostrategian luokittelu ja siirtymäilmiö

Sabherwalin ja muiden [2003] tietohallintostrategian tutkimuksen luokittelu on yhdistettävissä tietojenkäsittelyn kehityksen siirtymäilmiöön. Kuvassa 5 on malli tietohallintostrategian teoriatyyppeiden vallitsevuudesta suhteessa tietojenkäsittelyn kehitykseen. Suuntauksien muutoksille ei ole annettavissa selkeitä päivämääriä ja kehitys ei ole välttämättä tapahtunut murroksellisesti.



Kuva 5. Tietohallintostrategian teoriatyyppeiden vallitsevuus suhteessa tietojenkäsittelyn aikakausiin.

Kuvassa 6 on yhdistetty tietohallintostrategian luokittelu siirtymäilmiöön ja lueteltu joitakin ajanjaksojen tutkimusten tyypillisimpiä strategisia piirteitä. Eri tutkimustyypeillä on näkökulmansa myös niitä itseään edeltäviin aikakausiin, joita on taulukossa tuotu esille.

	<i>ATK-aika</i>	<i>MIS</i>	<i>SIS</i>	<i>KMN</i>
<i>Elämänkaariteoriat</i>	<ul style="list-style-type: none"> • Kustannussäästöt • Tehokkuus 	<ul style="list-style-type: none"> • Informaation kerääminen ja raportointi 	<ul style="list-style-type: none"> • Liiketoiminta-strategiset järjestelmät 	<ul style="list-style-type: none"> • Osaamisvoimavarojen tukeminen
<i>Universalistiset teoriat</i>	<ul style="list-style-type: none"> • Tietojenkäsittelyn mahdollistaminen 	<ul style="list-style-type: none"> • Järjestelmiin ja teknologiaan investoiminen 	<ul style="list-style-type: none"> • Uudelleen-organisointi 	<ul style="list-style-type: none"> • Globaalien organisaatioiden tietojenkäsittelyn järjestäminen
<i>Yhteistoinnallinen näkökulma</i>	<ul style="list-style-type: none"> • Resurssien tarkkailu 	<ul style="list-style-type: none"> • Liiketoiminnan tukeminen 	<ul style="list-style-type: none"> • Liiketoiminnan mahdollistaminen 	<ul style="list-style-type: none"> • Liiketoimintatiedon hallinta
<i>Katkoksellinen tasapainotila</i>	<ul style="list-style-type: none"> • Evolutionääriset kaudet organisaation tietojenkäsittelyssä • Vallankumoukselliset kaudet organisaation tietojenkäsittelyssä 			

Kuva 6. Teorioiden keskeisimmät näkemykset tietojenkäsittelyn eri aikakausien strategisista suuntauksista.

6. Yhteenveto

Tietohallintostrategian tutkimuksen luokittelu syventää käsitystämme tietohallintostrategian muutoksesta ja suhteuttaa tehtyjä tutkimuksia aikansa ilmiöihin. Luokittelun pohjalta voidaan myös arvioida tulevaa kehitystä ja tutkimuksen suuntaa.

Tietohallintostrategian varhaisimpien mallien voidaan todeta olevan elämänkaarimallien kaltaisia, ja niiden käytettävyyden edellyttävän jatkuvaa päivittämistä niin sanotun kypsän vaiheen siirtämiseksi jälleen sopivalle etäisyydelle. Tietohallintojohtamisen kypsän vaiheen saavuttamattomuuteen ovat päätyneet muiden muassa myös Sabherwal ja muut [2003] ja Ruohonen ja Salmela [1999]. Universalististen mallien vahvuus on niiden suunnan näyttäminen tarkemmalle tutkimukselle ja heikkous liiallinen yleistävyys. Yhteistoinnallinen näkökulma tarjoaa käytännönläheisimpiä toimintamalleja ja erityisesti liiketoimintaan käytännön suosituksia, mutta sen tarjoamat mallit ovat ajan vaatimusten suhteen rajoittuneita. Katkoksellisen tasapainotilan malli on erinomainen tarkasteltaessa yksittäisen kokonaisuuden kehityshistoriaa, mutta ei ole yleistettävissä vastaavalla tarkkuudella.

Tietohallintostrategian tutkimuksia ei ole viime aikoina paljolti luokiteltu muissa tutkimuksissa. Sabherwalin ja muiden tutkimuksen yhteydessä [2003] luoma lajittelu olisi edelleen helposti laajennettavissa laajemman tutkimuksen puitteissa. Strategioiden tunnusmerkistön tarkempia piirteitä voitaisiin luetella huomattavasti lisää, mutta samalla huomattaisiin niissä olevan enemmän yhteneväisyyksiä kuin eroja. Luokittelu on pienistä painotuseroista johtuvaa, ja joidenkin tutkimuksien osalta olisi vaikea vetää tarkkaa rajaa esimerkiksi universalistisen teorian ja yhteistoiminnallisen näkökulman välille. Myös tutkimussuuntausten tarkka rajaaminen ajallisesti on vaikeaa, sillä tutkimus ei ole edennyt aina murroksittain, ja joissakin aikaansa edellä olevissa tutkimuksissa on jo viitteitä tulevasta kehityksestä.

Tietohallintostrategian tutkimuksessa voitaisiin huomioida entistä enemmän kulttuurien vaikutus organisaatioihin. Tällä hetkellä tietohallintostrategian tärkeimpinä suhteina nähdään yritysstrategia ja henkilöstöstrategia, mutta globaalien organisaatioiden ongelmaksi muodostuu usein tietohallintoympäristön epätasainen kehitys. Tähän antaa viitteitä Palvian ja Palvian [2003] toteamat globaalien tietohallintoympäristöjen tietojenkäsittelyn kehittyneisyyden erot suhteessa taloudelliseen ja kulttuurilliseen kehitykseen. Tietohallintostrategia ja kulttuuri ovat sidoksissa toisiinsa ja strategian kehitys vaatii riittävän kypsää kulttuurillista pohjaa edistymiselle. Myös tässä kehityksessä on havaittavissa vaihteellisuus, jota voitaisiin edelleen verrata tietojenkäsittelyn kokemaan kehitykseen.

Viiteluettelo

- [Agarwal and Sambamurthy, 2002] Ritu Agarwal and V. Sambamurthy, Principles and models for organizing the IT function, *MIS Quarterly Executive* **1**, 1 (2002), 1-16.
- [Carr, 2003] Nicholas G. Carr, IT Doesn't Matter, *Harvard Business Review* **81**, 5 (May 2003), 41-49.
- [Davenport, 2000] Thomas H. Davenport, *Mission Critical – Realizing the Promise of Enterprise Systems*. Harvard Business School Press, Boston, 2000.
- [Earl, 1989] Michael J. Earl, *Management Strategies for Information Technology*. Prentice Hall, Hertfordshire, 1989.
- [Galliers and Somogyi, 1987] Robert D. Galliers and Elisabeth K. Somogyi, From data processing to strategic information systems – a historical perspective. In: Robert D. Galliers and Elisabeth K. Somogyi (eds.), *Towards Strategic Information Systems*. Abacus Press, Kent, 1987, 5-25.

- [Galliers and Sutherland, 2003] R. D. Galliers and A.R. Sutherland, Information systems management and strategy formulation: applying and extending the 'stages of growth' concept. In: Rober D. Galliers and Dorothy E. Leidner (eds.), *Strategic Information Management*. Elsevier Butterworth-Heinemann, Oxford, 2003, 33-63.
- [Järvinen, 2003] Pertti Järvinen, *Atk-toiminnan johtaminen*. Opinpajan kirja, Tampere, 2003.
- [Karimi and Konsynski, 2003] J. Karimi and B. R. Konsynski, Globalization and information management strategies, In: Rober D. Galliers and Dorothy E. Leidner (eds.), *Strategic Information Management*. Elsevier Butterworth-Heinemann, Oxford, 2003, 89-112.
- [Kemp and Clegg, 1987] Nigel J. Kemp and Chris W. Clegg. Information technology and job design: a case study on CNC machine tool working. In: Robert D. Galliers and Elisabeth K. Somogyi (eds.), *Towards Strategic Information Systems*. Abacus Press, Kent, 1987, 152-162.
- [Kettunen ja Simons, 2001] Jari Kettunen ja Magnus Simons (toim.), Toiminnanohjausjärjestelmän käyttöönotto pk-yrityksessä. Teknologiahäntöisestä ajattelusta kohti tiedon ja osaamisen hallintaa. VTT Automaatio, Espoo, 2001. Saatavilla: <http://www.vtt.fi/inf/pdf/julkaisut/2001/J854.pdf> (tark. 7.12.2005).
- [Laukkanen ja Vanhala, 1994] Mauri Laukkanen ja Sinikka Vanhala, *Liikkeenjohtamisen perusteet*. KY-Palvelu Oy, Helsinki, 1994.
- [Melville et al., 2004] N. Melville, K.L. Kraemer and V. Gurbaxani, Information technology and organizational performance: An integrative model of IT business value. *MIS Quaterly* **28**, 2 (2004), 283-321.
- [Palvia and Palvia, 2003] P.C. Palvia and S.C.Palvia, Understanding the global information technology environment: representative world issues, In: Rober D. Galliers and Dorothy E. Leidner (eds.), *Strategic Information Management*. Elsevier Butterworth-Heinemann, Oxford, 2003, 151-180.
- [Porter, 1985] Michael E. Porter, *Kilpailuetu: miten ylivoimainen osaaminen luodaan ja säilytetään*. Weilin+Göös, Espoo, 1985.
- [Porter, 1996] Michael E. Porter, What is strategy?. *Harvard Business Review* **74**, 6 (Nov/Dec 1996), 61-78.
- [Porter, 2001] Michel E. Porter, Strategy and the Internet, *Harvard Business Review* **79**, 3 (Mar 2001), 62.
- [Reponen et al. 1995] T. Reponen, T. Auer, J. Pärnistö ja J. Viitanen, Tietoresurssien Johtamisstrategia Kilpailukyvyn Välineenä. Turun kauppakorkeakoulun julkaisuja, **C-8:1995**, Turku, 1995.

- [Rouibah and Ould-ali, 2002], Kamel Rouibah and Samia Ould-Ali, PUZZLE: a concept and prototype for linking business intelligence to business strategy, *The Journal of Strategic Information System* **11**, 2 (2002), 133-152.
- [Ruohonen ja Salmela, 1999] Mikko Ruohonen ja Hannu Salmela, *Yrityksen Tietohallinto*. Edita Publishing Oy, Helsinki, 1999.
- [Sabherwal *et al.*, 2003] R. Sabherwal, R. Hirschheim and T. Goles, The dynamics of alignment: insights from a punctuated equilibrium model. In: Rober D. Galliers and Dorothy E. Leidner (eds.), *Strategic Information Management*. Elsevier Butterworth-Heinemann, Oxford, 2003, 311-346.
- [Swanson, 1994] E. Burton Swanson, Information systems innovation among organizations. *Management Science* **40**, 9 (1994), 1069-1092.
- [Turban *et al.*, 2002] Efraim Turban, David King, Jae Lee, Merril Warkentin, H. Michael Chung. *Electronic Commerce – A Managerial Perspective*. Prentice Hall, New Jersey, 2002.
- [VT, 2005] Valtion tietoyhteiskuntaohjelma 2005. Saatavilla: <http://www.valtionuuvosto.fi/tiedostot/pdf/fi/42828.pdf> (tark. 13.12.2005)

Tietojärjestelmästrategian merkitys pk-yrityksille

Eerik Savolainen

Tiivistelmä.

Tutkielmassa käsitellään tietojärjestelmätieteen tutkimuksessa esiin nousutta huomiota nykyisten tietojärjestelmästrategiamenetelmien soveltumattomuudesta modernien pk-yritysten tarpeisiin. Tähän mennessä kehitytyt menetelmät eivät palvele erityisen hyvin pk-yrityksiä, koska suuri osa niistä on kehitetty suurten konsernitason yritysten tarpeisiin. Menetelmät eivät myöskään vastaa kaikilta osin nykyisen nopeita muutoksia vaativan liiketoiminnan vaatimuksia. Ongelmaa on lähestytty perehtymällä alan julkaisuissa esitettyihin ongelma-kohtiin ja ratkaisuehdotuksiin. Tutkielma pyrkii tunnistamaan nousevan trendin ja selkeyttämään hieman hajanaista tutkimuskenttää.

Avainsanat ja -sanonnat: Tietojärjestelmä, tietojärjestelmästrategia.

CR-luokat: H.1

1. Johdanto

Tietojärjestelmätutkimus on painottunut suurten organisaatioiden tietojärjestelmiin, niiden hyödyntämisen ja johdannaisseurausten tutkimukseen. Pk-yritysten alettua panostamaan enemmän tietojärjestelmiin, on myös tutkimusta suunnattu tätä aluetta kohden. Pk-yritysten lisääntyneet tietojärjestelmäinvestoinnit johtunevat tietotekniikan käytön yleistymisestä ja jokapäiväistymisestä. Tietojärjestelmätoimittajille on kertynyt kokemusta suuryritysten tietojärjestelmämarkkinoilta ja toimittajat ovat alkaneet tuotteistamaan toimintaansa. Asia-kaskohtaisesti räätälöitävä järjestelmiä toimitetaan siis entistä harvemmin ja järjestelmät kootaan pääsääntöisesti liiketoiminnan eri osatoimintoja varten kehitetyistä moduuleista. Tuotteistettujen konfiguroitavien tietojärjestelmämoduulien hankkiminen on houkuttelevampaa ja helpompaa kuin kokonaisen järjestelmän tyhjältä pöydältä luominen. [Davenport, 2000] Enää ei siis vaadita valtavia resursseja tietojärjestelmän luomiseen, joten pk-yritykset ovat voineet ottaa yhä useammassa tapauksessa tietojärjestelmiä käyttöönsä. Liiketoimintaympäristön muutos ja globalisaatio luovat osaltaan myös paineita ja edellytyksiä tietojärjestelmien käyttöönottoon ja kehittämiseen. [Sauer et al., 1997]

Pk-yritysten aktivoituttua hyödyntämään tietojärjestelmiä liiketoiminnassaan, on huomattu tähän mennessä kehitettyjen metodologioiden ja työkalujen soveltumattomuus pk-yritysten tarpeisiin. Mallit ja menetelmät kaipaavat tarkennuksia soveltuakseen pk-yritysten toimintaan. Menetelmät kaipaavat päi-

vitystä myös vastatakseen nykyaikaisen liiketoimintaympäristön vaatimuksia, jossa myös pk-yritykset toimivat. [Levy and Powell, 2000; Smaczny, 2001]

2. Käsitteitä ja taustoja

Tässä luvussa esitellään tutkielman kannalta keskeisiä käsitteitä ja tutkimusongelman taustoja.

2.1. Pk-yritys

Huolimatta muutaman suomalaisen suuryrityksen korostuneesta näkyvyydestä tiedotusvälineissä, Suomen talouselämän perustan muodostavat pk-yritykset, mikä tekee tämän kokoluokan yrityksistä mielenkiintoisen tutkimuskohteen

EU:n virallisen määritelmän mukaan mikroyritysten sekä pienten ja keski suurten yritysten (pk-yritysten) luokka koostuu yrityksistä, joiden palveluksessa on vähemmän kuin 250 työntekijää ja joiden vuosiliikevaihto on enintään 50 miljoonaa euroa tai taseen loppusumma on enintään 43 miljoonaa euroa. [Komissio, 2003] Määritelmä on laadittu vuonna 2003 ja astunut voimaan 1.1.2005. Aikaisemmassa määritelmässä ei määritelty erikseen mikroyrityksiä, jolloin ne sisältyivät pienyritysten luokkaan. Nämä mikroyritykset eivät ole tutkielman kannalta merkittäviä, koska pienimuotoisen toiminnan johdosta niillä ei ole juurikaan tarvetta tietojärjestelmille.

Yritys- luokka	Henkilö- kunta	Liikevaihto	Taseen loppusumma
Mikro	<10	2 milj. euroa (ei aiempaa määritelmää)	2 milj. euroa (ei aiempaa määritelmää)
Pieni	<50	10 milj. euroa (ennen 7 milj. euroa)	10 milj. euroa (ennen 7 milj. euroa)
Keskisuuri	<250	50 milj. euroa (ennen 40 milj. euroa)	43 milj. euroa (ennen 27 milj. euroa)

Taulukko 1. 1.1.2005 voimaan tulleet PK-yritysten raja-arvot (aiemmat määritelmät suluissa). [EK, 2005]

Vuonna 2002 alle 50 työntekijän yrityksiä oli noin 97 %, ja alle 250 työntekijän yrityksiä yli 99 %, kaikista Tilastokeskuksen yritysrekisterin yrityksistä. Pk-yritysten palveluksessa oli vastaavia rajoja käyttäen 43 % tai 61 % kaikkien yritysten henkilöstöstä. Kaikkien suomalaisten yritysten liikevaihdosta kertyi alle 50 työntekijän yrityksissä 33 % ja alle 250 työntekijän yrityksissä 52 %. Kaikista palkoista alle 250 työntekijän yritykset maksoivat yli puolet, 56 %.

Vaikka pk-yrityksillä on merkittävä rooli Suomen taloudessa, työvoimaosuudella mitattuna Suomi määritellään Euroopassa suuryrityksvaltaiseksi

maaksi Hollannin, Saksan ja Iso-Britannian rinnalle. Tämä selittyy Euroopan suurella mikroyritysten määrällä. Euroopassa on noin 19 miljoonaa yritystä, joista 99 % on mikroyrityksiä sekä pieniä ja keskisuuria yrityksiä. Nämä yritykset työllistävät noin 70 % koko Euroopan työvoimasta. Tyypillinen eurooppalainen yritys on alle 10 henkilöä työllistävä mikroyritys. Nämä yritykset toimivat etenkin työvoimavaltaisilla aloilla.

2.2. Yritysten tietojärjestelmistä

Yleisen määritelmän mukaan tietojärjestelmä on ihmisistä, tietojenkäsittelylaitteista, ohjelmistoista ja tiedonsiirtokanavista koostuva kokonaisuus, jonka tarkoitus on tietoa käsittelemällä tehostaa tai helpottaa jotakin toimintaa tai tehdä toiminta mahdolliseksi. [Wikipedia, 2005]

Liiketoiminnassa tietojärjestelmien käytöllä pyritään saavuttamaan kilpailuetua toiminnan tehostamisen kautta. Yksittäisten tietojärjestelmän käytöllä pyritään tyypillisesti tehostamaan esimerkiksi varaston hallintaa (MRP), tilausten käsittelyä (EDI) tai asiakkuuksien hallintaa (CRM). Yksittäisten järjestelmien käyttö ei ole toisiaan poissulkevaa, mutta usean yksittäisen järjestelmän toiminnan yhteensovittaminen voi olla haasteellista ja aiheuttaa usein tehottomuutta. Tyypillinen tehottomuusesimerkki liittyy tehtävän moninkertaiseen suorittamiseen, kun sama tieto syötetään useaan kertaan eri järjestelmiin eri puolilla organisaatiota. [Davenport, 2000]

Tehokkuuden lisäämiseksi ja järjestelmien välisten kommunikaatio-ongelmien poistamiseksi on nykyään yleistä hankkia yksi järjestelmä, joka kykenee hoitamaan keskitetysti useita eri osatoimintoja. 1990-luvulta alkaen suuryritykset ovat ottaneet käyttöön toiminnanohjausjärjestelmiä (Enterprise Resource Planning, ERP). Näiden järjestelmien avulla voidaan hoitaa kaikki edellä mainitut osatoiminnot.

2000-luvulle tultaessa suuryritysten tietojärjestelmä markkinat ovat tasoittuneet ja tietojärjestelmätoimittajat ovat tuotteistaneet toimintaansa. Varsinkin ERP-järjestelmille on tyypillistä modulaarisuus, eli järjestelmä koostuu eri tehtäviä varten luoduista moduuleista, jotka konfiguroidaan asiakkaan liiketoiminnan mukaisiksi. Laajimpana tietojärjestelmä tyyppinä pidetään yritysjärjestelmiä (Enterprise System, ES), jotka kattavat yrityksen kaikki tietotarpeet. [Davenport, 2000] Käytännössä ES-järjestelmät ovat kattavia ERP-järjestelmiä, joten ES-järjestelmän termin käyttö ei ole yleistynyt.

Yritys voi siis nykyään hankkia esimerkiksi ERP-ohjelmiston talous- ja asiakashallinnon moduulit ja laajentaa järjestelmää myöhemmin tarvittavilla moduuleilla. [Davenport, 2000] Järjestelmien luokittelussa tulee kiinnittää huomiota järjestelmän tarkoitukseen ennemmin kuin sen hoitamiin osatehtäviin.

Pk-yrityksien käyttämät tietojärjestelmät vaihtelevat suuresti yrityksen toimialan mukaan. Yleisimmin pk-yrityksillä on käytössä taloushallinnon ja tilausten käsittelyn tietojärjestelmäratkaisuja. [Levy and Powell, 2000]

2.3. Tietojärjestelmästrategia

Yleisesti strategia on suunnitelma, jonka mukaan yritystä johdetaan kohti tavoitteita. *Tietojärjestelmästrategia* (Information System Strategy, ISS) on suunnitelma yrityksen tietojärjestelmien käytöstä ja toteutuksesta yrityksen kilpailukyvyin lisäämiseksi. [Levy and Powell, 2000; Peters et al., 2001]

Alun perin ajatus tietojärjestelmästrategian tarpeellisuudesta syntyi 80-luvulla liiketoimintastrategia-ajattelun kehittymisen ja yleistymisen jälkeen. Strategia-ajattelun mukaisessa ideaalitulanteessa kaikki yrityksessä toteutettavat toiminnot on johdettu liiketoimintastrategiasta. Liiketoimintastrategiaan kuuluu usein myös osastrategioita kuten tuote-, asiakas- ja toimialastrategiat.

Tietojärjestelmästrategia ajattelun syntymistä on edesauttanut tietotekniikan kehittyminen ja tietojärjestelmien liiketoiminnalle luomien mahdollisuuksien ymmärtäminen. Kun tietojärjestelmien liiketoiminnalle mahdollistamia etuja alettiin nähdä, oli luonnollista alkaa kehittämään tietojärjestelmästrategiaa, jotta tietojärjestelmistä saataisiin mahdollisimman suuri hyöty. [Peters et al., 2001]

Tutkijat ja asiantuntijat ovat painottaneet strategia-ajattelun alkutaipaleelta tähän päivään saakka tietojärjestelmä- ja liiketoimintastrategian yhdenmukaisuuden tärkeyttä. Mikäli tietojärjestelmästrategiaa ei muodosteta rinnan liiketoimintastrategian kanssa riskinä on, että kehitetään kaksi erillistä strategiaa, joiden tuotokset eivät tue toisiaan. Tällöin tietojärjestelmäinvestointi tuskin maksaa itseään takaisin. Koska tietojärjestelmien rooli liiketoiminnassa on kasvanut merkittävästi viime vuosikymmeninä, on strategioiden välinen ero käynyt entistä häilyvämmäksi.

2.4. Muuttuva liiketoimintaympäristö

Muutos edustaa epäpysyvyyttä ja asiantilan alituista muuttumista. Antiikin filosofit esittivät, että muutos on jatkuvaa ja läsnä kaikkialla. Muutamat heistä kiistivät kokonaan muutoksen olemassa olon, mutta tästä huolimatta maailman muutos on kiistämätön. Liiketoimintaa harjoitetaan tässä samassa muuttuvassa maailmassa, joten muutos on kiistämättä läsnä myös liiketoiminnassa.

Yrityksen toimintaan vaikuttaa lukemattomat ulkoiset ja sisäiset tekijät, joista muodostuu yrityksen liiketoimintaympäristö. Yrityksen ulkoista liiketoimintaympäristöä laajemmin tarkasteltaessa suuriksi muuttuviksi tekijöiksi voidaan nostaa PEST-analyysin poliittiset, taloudelliset, sosiaaliset ja teknologiset seikat. Nämä suuret muuttujat vaikuttavat kaikkiin yrityksiin. Yksittäisen

yrittäjien tason tarkastelussa liiketoimintaympäristössä korostuu erityisesti asiakkaat, kilpailijat ja yhteistyökumppanit.

Liiketoimintaympäristö muuttuu jatkuvasti. Tähän jatkuvaan muutokseen vastataan päivittäisillä operatiivisen tason toimilla. Operationaalisen tason toimenpiteitä on liiketoimintaprosessien muokkaaminen asteittain muutoksen ohjaamaan suuntaan. Toisinaan kohdataan tilanteita, jolloin yrityksen koko toiminta on järjesteltävä uudelleen liikeidea myöden, jotta toiminta voi jatkua muuttuneessa ympäristössä. [Martola, 1997]

Muutokseen sisältyy aina onnistumisen ja epäonnistumisen mahdollisuus. Se, että vaikuttaako muutos yrityksen toimintaan suotuisasti vai haitallisesti, riippuu muutostoimenpiteiden oikeellisuudesta ja ajoituksesta. Epäonnistumisen syynä voi olla huono käsitys toimintaympäristöstä ja sen lainalaisuuksista tai väärät oletukset muutostoimenpiteiden vaikutuksista. [Martola, 1997; Lehtoranta ja Uusikylä, 2005]

3. Pk-yrityksille tyypillinen liiketoimintaympäristö

Pk-yritykset toimivat usein rajatulla markkina-alueella ja ovat monin tavoin erikoistuneita. Tyypillisiä erikoistumistapoja ei voida määritellä niiden ollessa hyvin moninaiset. Pk-yritysten toiminnasta on kuitenkin selvästi havaittavissa taipumus erikoistua suurten toimijoiden kannalta vähemmän houkutteleville alueille. Usein pk-yrityksen liikeidea on kehitetty täyttämään suurten toimijoiden markkinoilta löytynyt aukko tai puute, jonka täyttäminen ei ole suurille yrityksille taloudellisesti riittävän houkuttelevaa. [Lehtoranta ja Uusikylä, 2005] Tämän tyyppisiä asetelmia esiintyy tyypillisesti suurten teollisuusyritysten tuki- ja oheistoiminnoissa. Tällaisia toimintoja voisi kuvailla suuryrityksen ydinliiketoiminnan kannalta välttämättömänä pahana; Tehtävinä jotka eivät varsinaisesti tuota taloudellista tulosta, mutta jotka täytyy kuitenkin suorittaa, jotta ydinliiketoimintaa voidaan harjoittaa. Varmistaakseen asemansa ja menestyäkseen omalla erikoisosaamisellaan pk-yrityksen on oltava alallaan yliveritaisen hyvä.

Vähemmän erikoistuneet yritykset joutuvat toimimaan usein erittäin tiukassa kilpailutilanteessa, jossa pienetkin erot toimittajien välillä voivat muodostua merkittäviksi. Suuri osa suomalaisista pk-yrityksistä toimii teollisuusyritysten alihankintaketjuissa, jotka on kilpailutettu tehokkaasti. Näissä ketjuissa markkinatilanne on tilaajan kannalta lähes täydellinen. Tällöin tilaajalla on sekä kattava tieto alihankkijoista ja niiden toiminnasta että vahvat neuvotteluasemat.

Toimiala, kilpailu ja erikoistumisaste vaikuttavat osana liiketoimintaympäristöä pk-yrityksen tietojärjestelmästrategian muodostamiseen. Tuotantopainotteiselle yritykselle kriittinen järjestelmä voi olla hyvin erityyppinen, kun

nopeasti reagoivalle palveluorganisaatiolle. Pitkälle viety erikoistuminen taas asettaa tietojärjestelmille omat vaatimuksensa.

Kovan kilpailutilanteen on osoitettu luovan organisaatiolle paineita muuttaa liiketoimintaprosessejaan [Agarwal, 1998]. Liiketoimintaprosesseja muutettaessa tehokkuutta saadaan lisättyä tuotantoa ja varastonhallintaa tehostavilla järjestelmillä. Mielenkiintoisen ja ajankohtaisen näkökulman tarkasteluun tuo e-Business -ilmiö. Sähköinen liiketoiminta mahdollistaa uusia tapoja järjestää yrityksen toiminta ja asettaa yrityksen tietojärjestelmille omat vaatimuksensa. [Kalakota, 2001]

Nykyajan liiketoiminnalle on tyypillistä nopea muutostahti. Muutoksen sykli, eli aikaväli muutosten välillä, on lyhentynyt merkittävästi mm. tietotekniikan luomien mahdollisuuksien myötä. Tietotekniikalla on kaksijakoinen rooli tässä muutoksessa. Uusi tekniikka mahdollistaa uuden ja nopeamman toiminnan, nopeuttaen samalla kehitystä. Tekniikan kehitystahdin nopeutuessa käytössä oleva tekniikka myös vanhenee nopeammin. Tekniikan kehitys ja kehittäminen luo varsinkin teknologiasuuntautuneille yrityksille jatkuvan muutospaineen. On pysyttävä muutoksen mukana tuotanto- ja suunnittelutekniikan osalta ja omaa myytävää tekniikkaa täytyy kehittää ajoissa, jotta sitä ennätetään myydä ja markkinoida riittävästi, ennen kuin se vanhenee. Tietyn tasoinen tekniikan hallitseminen ja käyttäminen on nykyään liiketoiminnan perusedellytys. Tekniikkaa innovatiivisesti hyväksi käyttäen voidaan saavuttaa merkittävää kilpailuetua.

4. Pk-yritysten tietojärjestelmä strategiatyön haasteita

Tässä luvussa käydään läpi tarkemmin pk-yritysten strategiatyöskentelylle tyypillisiä piirteitä, nostetaan esiin nykyisten menetelmien ongelmia SAM-mallin kautta ja esitellään pk-yrityksiä varten kehitetty tietojärjestelmästrategiamenetelmä.

4.1. Nykyiset strategiatyökalut pk-yritysten käytössä

Pk-yrityksillä on huomattu olevan vaikeuksia soveltaa suuryrityksille kehitettyjä menetelmiä tietojärjestelmästrategian luonnissa [Dunhal et al., 2005; Levy and Powell, 2000]. Toimintaympäristö vaikuttaa pk-yritysten toimintaan huomattavasti enemmän kuin suuryrityksiin, jotka voivat tietyn koon saavutettuaan jopa muokata toimintaympäristöään. Jo näistä peruslähtökohdissa olevista eroista on pääteltävissä, että tietojärjestelmästrategian kehittämistä on lähestyttävä eri näkökulmasta. Näin ollen yleisesti käytettävät menetelmät eivät palvele erityisen hyvin pk-yrityksiä.

Seuraavaksi tätä ongelmaa havainnollistetaan strategioiden linjauksen yhteydessä esiin nousseiden seikkojen avulla.

4.2. Tietojärjestelmä- ja liiketoimintastrategioiden linjaus

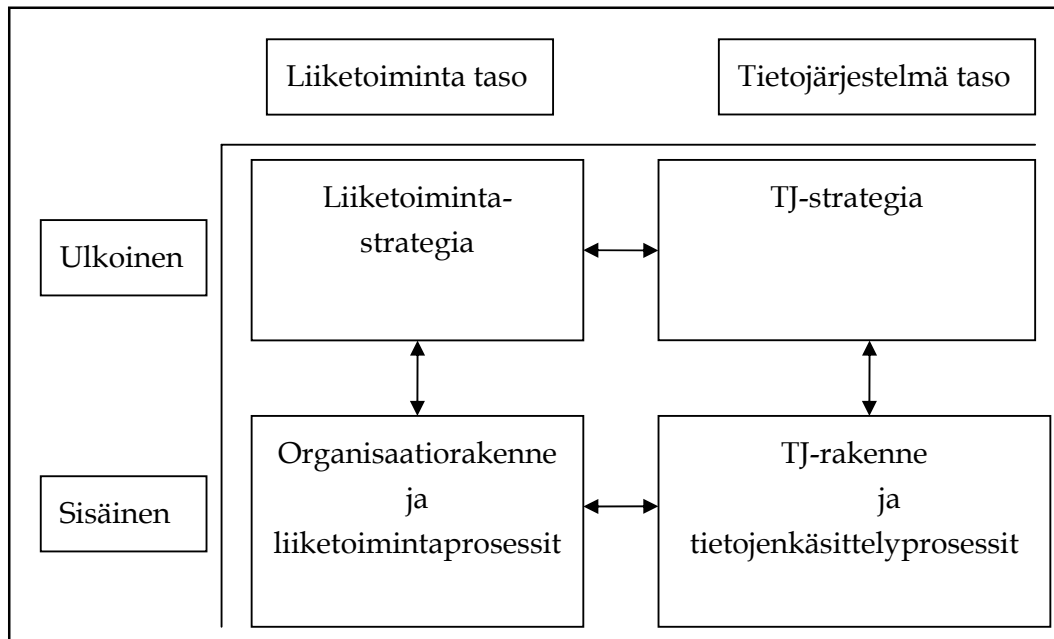
Liiketoiminta- ja tietojärjestelmästrategioiden yhdenmukainen linjaus lisää tietojärjestelmäinvestointien tuottavuutta, auttaa organisaatioita saavuttamaan kilpailuetua tietotekniikan avulla ja helpottaa muutokseen reagoimista ja uusien mahdollisuuksien hyödyntämistä. [Avison et al., 2004; Davenport, 2000; Duhan et al., 2005; Levy and Powell, 2000; Peters et al., 2001; Porter and Millar, 1985; Sauer et al., 1997]

Strategioiden linjauksessa kaksi yleisimmin käytettyä mallia on Mortonin kehittämä MIT90- ja Hendersonin ja Venkatraman [1991] kehittämä Strategic Alignment Model, SAM-malli. SAM-malli on näistä kahdesta ajanmukaisempi tunnistaessaan liiketoiminta- ja tietojärjestelmästrategioiden vuorovaikutuksen. [Avison et al., 2004]

4.2.1. Strategic Alignment Model (SAM)

Perinteisesti tietojärjestelmästrategia on ajateltu yrityksen irrallisen strategiana eikä sen yhteyksistä liiketoimintaan ole juurikaan välitetty. SAM-malli on kehitetty alun perin korjaamaan tätä epäkohtaa, kun tietojärjestelmät ovat käyneet entistä merkittävämmiksi liiketoiminnassa.

Mallin avulla on tarkoitus yhdenmukaistaa liiketoiminta- ja tietojärjestelmästrategioita ja auttaa näkemään tietojärjestelmien ja liiketoiminnan vuorovaikutusta yrityksen sisäisellä ja ulkoisella tasolla. Mallin ulkoisen tason muodostaa tietojärjestelmä- ja liiketoimintastrategia. Tämä taso määrittää mm. yrityksen aseoitumisen markkinoille ja tietojärjestelmien käytön ja vaikutuksen liiketoiminnallisten tavoitteiden saavuttamisessa. Tästä käytetään termiä strategic fit. Sisäinen taso kuvaa yrityksen liiketoimintaprosessien ja tietojärjestelmän prosessien vastaavuutta ja vuorovaikutusta operationaalisella tasolla. Liiketoiminta- ja tietojärjestelmätasojen suhdetta ulkoisella ja sisäisellä tasolla on havainnollistettu kuvassa 1.



Kuva 1. SAM-malli

4.3. Tietojärjestelmästrategian merkitys erityyppisille yrityksille

Tietojärjestelmästrategia johdetaan liiketoimintastrategiasta ja se vaikuttaa tietojärjestelmän toteuttamisen kautta yrityksen liiketoiminnan käytännön toteutukseen. Esimerkkitapaukseksi voitaisiin ottaa tilanne, jossa toteutettu järjestelmä mahdollistaa yritykselle sähköisen liiketoiminnan aloittamisen. Tietojärjestelmästrategian mukaan toteutettu järjestelmä vaikuttaa siis yrityksen liiketoimintaan sen mukaan, kuinka ja mihin toimintoihin sitä käytetään. Järjestelmän käyttäjätyytyväisyys vaikuttaa myös yrityksen liiketoimintaan. Tämän yhteyden ymmärtämistä auttaa kun muistaa, että järjestelmän käyttäjinä voivat olla sekä työntekijät että asiakkaat. [DeLone and McLean, 2003]

Tietojärjestelmästrategian painoarvo strategiakokonaisuudessa riippuu tietoteknisten ratkaisujen käytöstä liiketoiminnan mahdollistajina. Mikäli asiakkaalle yrityksen arvoa tuottavat ydinliiketoimintaprosessit toteutetaan tietotekniikan avulla, on ilmeistä että tietojärjestelmästrategian painoarvo on suuri. Päinvastainen tilanne, jossa tietojärjestelmästrategia ei ole merkittävässä roolissa, voisi esiintyä yrityksessä, jossa on käytössä vain niin sanottuja back-office-järjestelmiä, jotka hoitavat taustalla rutiinitehtäviä kuten palkanlaskentaa tai dokumenttien hallintaa.

4.4. Pk-yritysten strategioista yleisesti

Pk-yritysten liiketoiminta eroaa useissa tapauksissa merkittävästi suuryritysten liiketoiminnasta. Suuret toimijat ovat saavuttaneet ns. kriittistä massaa, jolloin ne eivät ole yhtä alttiita ympäristössä tapahtuville muutoksille kuin pk-yritykset. Pk-yrityksille esimerkiksi yksittäisen asiakkaan tai tavarantoimittajan menettäminen voi olla liiketoiminnan kannalta merkittävä muutos.

Pk-yritysten toimintaa leimaa joustavuus ja kyky tehdä nopeita päätöksiä. Nämä ominaisuudet johtuvat pitkälti näiden yritysten johdon rakenteesta ja toimitavoista. Pk-yritystä johdetaankin useassa tapauksessa epämuodollisesti pienen ydinryhmän toimesta. Tämä ydinryhmä voi tehdä nopeita päätöksiä tarvittavan päätösinformaation löytyessä ydinryhmän sisältä ja muodollisen päätöksentekoprosessin puuttuessa. Liiketoiminnan kannalta olennaisia tietoja ei ole välttämättä saatavissa mistään eksplisiittisessä muodossa vaan tieto on tallessa vain ydinryhmän muistissa. Tällä tavoin johdetuilta yrityksiltä puuttuu usein eksplisiittinen liiketoimintastrategia, josta tietojärjestelmästrategia pitäisi johtaa. [Levy and Powell, 2000]

4.5. Järjestelmien kankeus vs. joustavuus

Pk-yritysten nopeasti muuttuvassa ympäristössä usein strategioiden luontia ja linjausta tärkeämpää on tietojärjestelmien toiminnan ja liiketoimintaprosessien vastaavuus operationaalisella tasolla. Tämä korostuu varsinkin tapauksissa, joissa liiketoimintaprosesseja joudutaan muuttamaan usein. On turhaa työtä kehittää strategiaa liiketoimintatilanteisiin ja strategian pohjalta tietojärjestelmää, jos muutokseen pystytään vastaamaan pienemmillä toimenpiteillä. [Smaczny, 2001] Järjestelmien kehittäminen myös maksaa rahaa ja investointi tarvitsee aikaa maksaakseen itsensä takaisin. Mikäli liiketoimintaprosesseja tarvitsee muuttaa, ennen kuin investointi on maksanut itsensä takaisin, aiheutuu tästä tappiota yritykselle. [Davenport, 2000] Kuitenkin strateginen suunnittelu on tarpeen pitkän tähtäimen toiminnan kannalta. Mikäli järjestelmiä kehitetään hetkellisten tarpeiden pohjalta, investoinneista ei tällöin saada maksimaalista hyötyä. Tällöin järjestelmien liiketoimintaa muokkaava vaikutus ja synergia edut muiden toimintojen tehostumisen myötä jäävät usein vähäisiksi. [Levy and Powell, 2000]

Toiminnan laajasti kattavan järjestelmän kehittäminen ja muutosvalmiuden säilyttäminen on haastava tehtävä. Kalakotan [2001] näkemyksen mukaan nykyaikaiseen nopeaan liiketoimintaan soveltuva tietojärjestelmäarkkitehtuurin perustan voisi luoda ERP-järjestelmä, joka hoitaa back-office-järjestelmille tyypillisiä tukitoimintoja. Tämä perusinfrastruktuuri hoitaa rutiinitehtäviä, joiden ei oleteta muuttuvan radikaalisti, jolloin siihen on turvallista investoida.

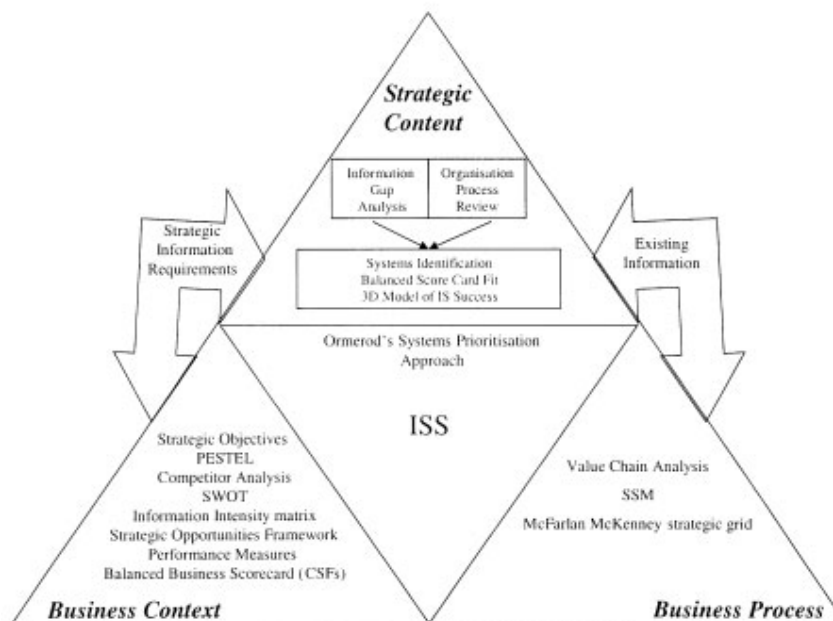
Tämä järjestelmän päälle voidaan luoda esimerkiksi kevyempiä ja joustavampia järjestelmän osia, jotka tukevat kriittisiä asiakkaalle lisäarvoa tuottavia tehtäviä.

4.6. Moniulotteinen lähestyminen pk-yritysten tietojärjestelmästrategiaan ratkaisuksi?

Levy ja Powell [2001] esittelevät tutkimuksen tuloksena syntyneen pk-yrityksille suunnatun tietojärjestelmästrategian luontimenetelmän. Menetelmä on muokattu konsernitason tietojärjestelmästrategian luontimenetelmästä pk-yrityksien tarpeet huomioon ottavaksi ja se on pyritty luomaan joustavaksi, jotta se soveltuisi mahdollisimman monen pk-yrityksen toimintaan. Erityistä huomiota on kiinnitetty järjestelmien tuottaman informaation vaatimuksille. Oikeellisen päätösinformaation katsotaan olevan merkittävän tärkeää nopeasti reagoiville pk-yrityksille.

Menetelmä huomioi yrityksen toiminnasta kolme ulottuvuutta jotka huomioidaan strategian luonnissa liiketoimintaympäristön, liiketoimintaprosessit ja toiminnan strategisen sisällön. Levy ja Powell [2001] viittaavat Earlin viitekehysten viitekehysten ja Walshamin sosiotekniseen organisaationäkemykseen. Earlin viitekehys esittelee yleisesti tunnettuja ja käytettyjä liiketoiminnan analysointimenetelmiä joita käytetään Levyn ja Powellin [2001] menetelmässä. Walshamin sosiotekninen organisaationäkemyks sopii pk-yritys kontekstiin. Levyn ja Powellin [2001] menetelmässä otetaan myös huomioon pk-yrityksille tyypillinen päätöksenteko- ja johtamistapa.

Kuvassa 2 on esitetty tarkasteltavien liiketoiminta ulottuvuuksien yhteydet, sekä käytettävät analysointityökalut jokaisen liiketoiminta ulottuvuuden kohdalla Levyn ja Powellin [2001] mallin mukaan.



Kuva 2. Levyn ja Powellin [2001] menetelmän ulottuvuudet ja analysointityökalut.

Menetelmää on testattu englantilaisilla pk-yrityksillä ja sen on todettu olevan toimiva. Testauksen perusteella menetelmää tulisi vielä kehittää valittujen analysointi työkalujen osalta. Vastaava suomalaisiin pk-yrityksiin kohdistuvaa tieteellistä tutkimusta ei ole tehty.

5. Yhteenveto

Muun muassa Smaczny [2001] kritisoi tietojärjestelmä- ja liiketoimintastrategian linjauksen keskeistä asemaa modernin liikkeenjohdon apuvälineenä. Tutkielman pohjalta voidaan todeta tämä kritiikin olevan aiheellista nykyisessä nopeasti muuttuvassa liiketoimintaympäristössä ja käytettävien menetelmien kaipaavan päivitystä. Kritiikki osuu myös varsin hyvin pk-yritysten tietojärjestelmästrategian käyttöön. Tutkielman perusteella voidaan todeta myös, että tietojärjestelmästrategia työskentely on sulautumassa entistä enemmän liiketoimintastrategia, työskentelyyn tietojärjestelmien liiketoiminnallisen merkityksen kasvaessa. Huomiota tullaan kiinnittämään entistä enemmän tietojärjestelmien ja liiketoimintaprosessien vastaavuuteen operationaalisella tasolla, sekä yritysten kykyyn luoda joustavia tietojärjestelmiä, jotka eivät rajoita yrityksen muutosvalmiutta.

Suomalaisten pk-yritysten käyttämiä tietojärjestelmästrategiamenetelmiä koskevaa tutkimusta ei ole juurikaan saatavilla. TEKES, TE-keskukset, VTT ja Työministeriö tukevat pk-yritysten teknologiahankkeita taloudellisesti ja tiedol-

lisesti. Myös yksityisellä sektorilla on syntynyt liiketoimintaa tietojärjestelmästrategiatyön ympärille, mutta tieteellisiä tutkimustuloksia ei ole alalta juuri saatavilla.

Viiteluettelo

- [Agarwal, 1998] Agarwal, R., Small firm survival and technological activity. *Small Business Economics* **11**, (1998), 215-224.
- [Avison et al., 2004] David Avison, Jill Jones, Philip Powell and David Wilson, Using and validating the strategic alignment model. *Journal of Strategic Information Systems* **13**, 3 (Sept. 2004), 223-246.
- [DeLone and McLean, 2003] William H. DeLone and Ephraim R. McLean, The DeLone and McLean model of information systems success: a ten-year update. *Journal of Management Information Systems* **19**, 4 (Spring 2003), 9-30.
- [Davenport, 2000] Thomas H. Davenport, *Mission Critical: Realizing the Promise of Enterprise Systems*. Harvard Business School Press, 2000.
- [Duhan et al., 2005] Duhan S, Levy M and Powell P, IS Strategy in SMEs using organizational capabilities: the CPX framework. In: Proc. of 13 th. ECIS. Saatavana elektronisessa muodossa: <http://is.lse.ac.uk/asp/aspecis/20050027.pdf>
- [EK, 2005] Elinkeinoelämän keskusliitto. Yrittäjyys ja pk-yritykset. *EK:n www-sivut* (17.11.2005) Saatavana elektronisessa muodossa: http://www.ek.fi/ek_suomeksi/yrittajyys/pk_yritykset/index.php
- [Heir et al., 2000] Björn Heir, Ella Juneja, Tommi Kalilainen, Waltteri Karhusaari, Tommi Nylander ja Tapio Rasimus, *Digitaalinen tarjontaketju: tavara- ja tietovirrat uudessa taloudessa*. WSOY, 2000.
- [Henderson and Venkatraman, 1991] Henderson, John C. Venkatraman, N., Understanding strategic alignment. *Business Quarterly* **56**, 3 (Winter 1991), 72-79.
- [Kalakota, 2001] Ravi Kalakota, *E-business 2.0*. Addison-Wesley, 2001
- [Komissio, 2003] Euroopan yhteisöjen komissio, Suositus mikroyritysten sekä pienten ja keskisuurten yritysten määritelmästä. *Euroopan unionin virallinen lehti* **46**, L124 (20.5.2003). Saatavana elektronisessa muodossa: http://europa.eu.int/eur-lex/pri/fi/oj/dat/2003/l_124/l_12420030520fi00360041.pdf.
- [Levy and Powell, 2000] M. Levy and P. Powell, Information systems strategy for small and medium sized enterprises: an organisational perspective. *The Journal of Strategic Information Systems* **9**, 1 (March 2000), 63-84.
- [Lehtoranta ja Uusikylä, 2005] Toim. Olavi Lehtoranta ja Marjo Uusikylä, Innovaatiot pk-yritysten kasvun vauhdittajina: 25 kertomusta kasvuyrityksistä.

- Teknologiakatsaus*, **166/2005**, TEKES 2005. Saatavana elektronisessa muodossa: http://www.tekes.fi/julkaisut/Innovaatiot_pk.pdf
- [Martola, 1997] Ulla Martola, *Liiketoimintaprosessit: BPR-muutoksen johtaminen*. WSOY, 1997.
- [Peters et al., 2001] S. C. A. Peters, M. S. H. Heng and R. Vet, Formation of the information systems strategy: in a global financial services company. *Information and Organization* **12**, 1 (Jan. 2002), 19-38.
- [Porter and Millar, 1985] M.E. Porter and V.E. Millar, How information gives you competitive advantage. *Harvard Business Review* (1985) 140-160. etsi tarkempi viite ebhostin kautta
- [Sabherwal and Chan, 2001] Rajiv Sabherwal and Yolande E Chan. Alignment between business and IS strategies: A study of prospectors, analyzers, and defenders. *Information Systems Research* **12**, 1 (Mar. 2001), 11-23.
- [Sauer et al., 1997] Christopher Sauer, Philip W. Yetton and Associates, *Steps to the Future: Fresh Thinking of IT-Based Organizational Transformation*. Jossey-Bass, 1997.
- [Smaczny, 2001] Tomasz Smaczny, Is an alignment between business and information technology the appropriate paradigm to manage IT in today's organisations? *Management Decision* **39**, 10 (Dec. 2001), 797-802.
- [Wikipedia, 2005] *Wikipedia – Vapaa tietosanakirja*, tietojärjestelmän määritelmä (23.11.2005). Saatavana elektronisessa muodossa: www.wikipedia.fi

Kotikäyttäjän tietoturva

Atte Sepponen

Tiivistelmä.

Tässä tutkielmassa teen tietoturvaan liittyvän riskikartoitus erilaisista uhkatekijöistä, joita tavallinen kotikoneen käyttäjä saattaa kohdata käyttäessään tietokonetta ja Internetiä. Riskikartoituksen lisäksi pyrin tarjoamaan ratkaisuja näihin ongelmiin kotikäyttäjälle suunnatun ohjeistuksen muodossa.

Avainsanat ja -sanonnat: Tietoturva, kotikäyttäjät, riskikartoitus, Internet-uhat, haittaohjelmien torjunta.

CR-luokat: K.6.5, D.4.6

1. Johdanto

Nopeiden laajakaistayhteyksien hinnat ovat halventuneet ja näin yhä useampi kotikäyttäjät on tullut hankkineeksi itselleen Internet-liittymän ”tiedon valtatielle”. Yhä useampi tietokone on kytketty tähän julkiseen verkkoon ja päässyt osaksi sen valtaisa informaatiossa. Tähän liittyy myös paljon erilaisia ongelmia ja uhkia, etenkin tavallisen kotikäyttäjän kannalta, joka ei niitä kaikkia edes tiedosta. Internet on nimittäin tarjonnut mahdollisuuden rikollisille ja ilkeillä tekijöille sekä erilaisille haittaohjelmille.

Internet sai alkunsa tutkimusprojektista, jossa oli tarkoituksena verkottaa tietokoneita etäkäyttöön ja tiedonvälitykseen. Tämä kaikki rakentui luottamuksen ympärille: käyttäjät noudattivat tarkkoja ohjeita koneiden käytössä eikä kukaan väärinkäyttänyt järjestelmää. Tämän päivän Internetin pohja on edelleenkin tuon projektin verkko, mutta luottamus-ajatus ei enää toimi samalla tavalla. [CERT/CC, 2002]

Tässä tutkielmassa pyrin kartoittamaan ja kuvailemaan riskejä, joita tavallinen kotikoneenkäyttäjät voi kohdata Internetistä. Valitsin Internetin lähtökohdaksi, koska sen tuomat uhat ovat kotikoneen käyttäjälle kaikkein ajankohtaisimpia ja todennäköisimpiä. Lisäksi laadin riskikartoituksen pohjalta ohjeistuksen, jossa esittelen ohjelmia ja menetelmiä näiden ongelmien minimoimiseen.

Tutkielmani etenee niin, että seuraavassa luvussa pyrin avaamaan laajemmin tietoturva-käsitteen ja liittämään siitä keskeiset osat kotikäyttäjään. Kolmannessa luvussa menen itse riskien kartoitukseen ja niissä esiintyviin toimintatapoihin. Varsinainen ohjeistus riskien poistoon ja ehkäisyyn seuraa neljännessä luvussa, jossa esittelen eri tietoturvaohjelmia kotikäyttäjän Windows XP -ympäristössä. Lisäksi esittelen erilaisia tietoturvamenetelmiä ja

tapoja riskien minimoimiseen. Viimeinen luku on tarkoitettu aiheen yhteenvedolle.

2. Tietoturvan käsite

Tietoturvallisuus jaotellaan Valtionhallinnon tietoturvakäsitteistössä [2003] kahteen osaan: tavoitetilaksi ja lainsäädännöllisiksi toimenpiteiksi. Tavoitetilalla tarkoitetaan tietojärjestelmien ja palveluiden suojaamista niin, että niihin kohdistuvat uhat eivät aiheuta vahinkoa yhteiskunnalle tai sen jäsenille. Lainsäädännöllä puolestaan on tarkoitus varmistaa tavoitetilän toteutuminen kaikissa olosuhteissa. Tietoturvallisuuteen liittyy myös kiinteästi käsite tietosuoja, joka tarkoittaa yksityisyyden suojaamista. Jatkossa käyttäessäni tietoturva-käsitettä oletan siihen myös sisältyvän tietosuojan, koska nämä käsitteet ovat sisällöllisesti lähellä toisiaan.

Tavoitetilä voidaan jakaa vielä kuuteen osa-alueeseen: tiedon luottamuksellisuuteen, eheyteen, saatavuuteen, todennukseen, pääsynvalvontaan ja kiistämättömyyteen. Tiedon luottamuksellisuudella tarkoitetaan sitä, että kukaan ulkopuolinen ei pääse käyttämään vain muille tarkoitettua tietoa. Kotikäyttäjän tapauksessa tämä voisi tarkoittaa esim. Internetissä hoidettavia pankkiasioita. Luottamuksellisuus rakentuu kokonaisuudesta, että käyttäjä tunnistetaan ensin pankin asiakkaaksi ja hän voi näin hoitaa pankkitoimensa verkon kautta. Tämä "luottamustunnistus" on pitänyt erikseen rakentaa järjestelmään Internetin historian takia. Tiedon eheydellä puolestaan tarkoitetaan sitä, ettei kukaan ulkopuolinen pääse muokkaamaan tai poistamaan tietoa, kuten kotikäyttäjän tiedostoja. Tiedon saatavuus taas liittyy enemmänkin tietokoneiden toimivuuteen ja siihen, että käyttäjän tulee päästä käsiksi hänelle kuuluvaan tietoon silloin, kun se hänelle sopii. Esimerkiksi kotikoneen tulee toimia normaalisti. Loput osa-alueet, todennus, pääsynvalvonta ja kiistämättömyys, voidaan selittää Internetin pankkiasioimisimerkin avulla. Todennus tarkoittaa käyttäjän tunnistamista tunnus-salasana -parin avulla pankin järjestelmään, kun taas pääsynvalvonta kuuluu pankin puolelle ja heidän järjestelmänsä toimintaan. Kiistämättömyys tarkoittaa esimerkin tapauksessa todistetta siitä, että käyttäjä on suorittanut tilisiirron. [Järvinen, 2002a]

Tietoturva koostuu jaloista toimintaperiaatteista, mutta käytännössä niitä kaikkia on vaikea toteuttaa ja ylläpitää. Tietoturva rakentuu ihmisistä ja koneista. Tämä tarkoittaa sitä, että rakennelman vahvuus riippuu suoraan sen osatekijöistä – rakennelma on niin vahva kuin sen heikoin lenkki. On kuitenkin muistettava, että täydellistä tietoturvaa ei ole mahdollista saavuttaa. Kaikki riskitekijät tuntuvat suurilta, koska ihmiset odottavat uusilta koneilta ja järjestelmiltä täydellistä toimivuutta ja oikeellisuutta. Samalla kuitenkin itse

hyväksyvät oman toimintansa virheet ja ongelmat: ajokorttia tai passia ”vilauttamalla” selviää henkilöllisyyden tunnistuksesta ja samoin henkilöturvatuksen kertomisella. [Järvinen, 2002a]

Tietoturva nostetaan usein uutisotsikoihin, ja varsinkin Internetin ja kotikäyttäjän tapauksessa siitä on tullut iso asia. Tiedotusvälineissä Internet näytetään usein paheiden pesänä, jossa pedofiilit, hakkerit ja huijarit mellastavat. Todellisuudessa asiat eivät ole näin huonosti, sillä Internetissä on paljon järjestelmiä, jotka tarjoavat tietoturvalle hyvän alustan. Uutiset nostavat luonnollisesti esiin poikkeavia asioita. Tietoturvasta itsestään on myös tehty suuri bisnes, jossa pelon tunteella saadaan aikaan ”markkinaliikehdintää”. [Järvinen, 2002a]

3. Riskikartoitus

Internetiin kytkettynä tietokone on monien uhkien kohteena. Tavallinen kotikäyttäjä ei näitä kaikkia edes välttämättä tiedosta tai usko niiden olevan realistisia, sillä tiedotusvälineisiin pääsevät tietoturvauutiset koskevat pääasiassa organisaatioita. Suuret luottokorttitietovarkaudet ja organisaatioiden tietojärjestelmämurrot löytävät helposti tiensä uutisiin. Uutiset eivät kuitenkaan kerro kaikkea, ja kotikäyttäjällä on todellisuudessa vastassaan melkoinen ongelmien vyyhti. Tietotekniikka ja siihen liittyvä tietoturva vaativat paljon ajan hermoilla pysymistä, mikä on jo hankalaa alan asiantuntijallekin. Ei voida olettaa kotikäyttäjän tuntevan kaikkia uhkia, mutta silti ne ovat aitoja ja niistä on puhuttava. Pyrin seuraavaksi kartoittamaan tällä hetkellä keskeisimmät riskit, jotka uhkaavat kotikäyttäjää.

3.1. Virukset ja madot

Tietokoneviruksilla tarkoitetaan ohjelmakoodia, joka pystyy kopioimaan ja levittämään itseään uusiin kohteisiin. Tyypillistä viruksille on se, että ne tarvitsevat aputiedoston, isäntäsolun, jonka kautta ne voivat levitä ja aktivoitua. Virukset ovat ns. haittaohjelmia, jotka ovat ihmisten tarkoituksellisesti tekemiä vahingollisia tietokoneohjelmia. Nämä ohjelmat voidaan jakaa niiden tarttumiskohteen perusteella neljään päätyyppiin: tiedostovirukset, makrovirukset, komentovirukset ja käynnistyslohkovirukset. Tiedostovirukset tarttuvat tiedostoihin, joilla on tietty tunniste, esim. .exe, .com, .scr jne. Makrovirukset puolestaan tarttuvat toimisto-ohjelmilla tehtyihin dokumentteihin ja aktivoituvat, kun dokumentti avataan. Komentojonovirukset ovat erilaisilla komentokielillä tehtyjä, kuten esim. Windows-käyttöjärjestelmän Visual Basic Scripting. Käynnistyslohkovirukset tarttuvat tietokoneen kiintolevyn käynnistyslohkoon, josta tietokone etsii käynnistystiedostoja. [VM, 2004]

Virukset voivat mm. poistaa tai muokata käyttäjän tiedostoja tai käyttöjärjestelmälle tärkeitä tiedostoja aiheuttaen näin tietokoneen hidastumista tai toimimattomuutta. Ne voivat myös lähettää tietokoneesta löytyvää materiaalia eteenpäin verkkoyhteyden kautta. Lisäksi ne voivat heikentää tietokoneen tietoturvaavaamalla takaovia tai portteja muille haittaohjelmille. Virukset voivat olla myös aiheuttamatta minkäänlaista todellista vahinkoa: ne vain leviävät, kuluttavat tietokoneen resursseja tai pilailevat käyttäjän kustannuksella.

Madot ovat viruksien tavoin haittaohjelmia. Ero viruksiin nähden on siinä, että ne eivät tarvitse aputiedostoa levitäkseen. Ne kykenevät omatoimiseen itsensä kopioimiseen ja leviämiseen. Madot ovat hieman viruksia kehittyneempiä, sillä niissä voidaan havaita olevan tietynlaista älykkyyttä, kuten tietojen kerääminen ja analysointi, hyökkäys, leviäminen, itsensä päivittäminen ja etähallinta. Madot siis voivat kerätä tietoja kotikäyttäjän tietokoneesta ja analysoida niitä. Tämän jälkeen ne suorittavat hyökkäyksen kohdekoneeseen tietojen perusteella ja käyttävät mahdollisia aukkoja murtautuakseen sisään. Sisään päästyään ne tekevät samanlaisia tekoja kuin viruksetkin. Mato saattaa yrittää koneelle pelkän avonaisen Internet-yhteyden läpi. [Qing and Wen, 2005]

Virukset ja madot leviävät pääasiassa sähköpostitse, WWW-sivuilla, verkkoyhteyksien kautta, uutisryhmissä, tiedostoissa ja dokumenteissa. Tällä hetkellä on enää vaikeaa vetää selvää rajaa virusten ja matojen välille, koska etenkin kehittyneemmät virukset ja madot muistuttavat toisiaan.

3.2. Troijan hevoset

Troijan hevoset kuuluvat myös haittaohjelmien joukkoon, joita kuvataan usein vakavaksi uhaksi. Ne eivät kuitenkaan viruksista ja madoista poiketen levitä itseään. Ne tarvitsevat siihen käyttäjän apua ja hänen hyväuskoisuuttaan. Ne ovat juuri kreikkalaisen tarinansa kaltaisia, mutta vain puettuina tietokoneympäristöön. Tarun mukaan kreikkalaiset nimittäin onnistuivat valtaamaan Troijan kaupungin suuren puuhevoson ansiosta. Kreikkalaiset jättivät tämän hevosen Troijan kaupungin porttien ulkopuolelle. Troijalaiset luulivat hevosta lahjaksi ja siirsivät sen kaupungin muurien sisään. He eivät kuitenkaan tiedeneet, että suuri joukko kreikkalaisia sotilaita oli piiloutunut hevoseen. Niinpä yön tultua kreikkalaiset astuivat ulos hevosesta ja valtasivat kaupungin. Tämän päivän tietokoneisiin liitettynä Troijan hevoset ovat siis haittaohjelmia, jotka naamioivat itsensä tavallisten ohjelmien sekaan. Troijan hevonen voi olla esim. piiloutuneena jonkin pelin tai hyötyohjelman sisään. Tällainen ohjelma voi muuten toimia normaalisti, mutta käyttäjän aktivoitessa ohjelman se saattaa samalla huomaamattomasti poistaa tärkeitä tiedostoja, avata takaovia uusille haittaohjelmille, levittää viruksen, madon tai päästää uuden troijalaisen valloilleen. [Ford, 1999]

Internetin yleistymisen on osasyynä myös Troijan hevosten lisääntymiseen, vaikka niitä onkin selvästi viruksia ja matoja vähemmän. Verkko on tarjonnut niille leviämislustan, ja Troijan hevosten torjuminen on hankalaa. Tämä johtuu juuri siitä, että nämä haittaohjelmat ovat piilossa, eikä niiden todellinen olemus paljastu heti. Näin käyttäjä on helppo saada käyttämään tavalliselta näyttävää ohjelmaa ilman, että hän olettaisi sen tekevän jotain haitallista. Periaatteessa kaikissa tuntemattomissa ohjelmissa voi olla Troijan hevonen. Mistä tietää, missä ei ole? Kaikki perustuu käyttäjän luottamuksen saamiseen.

3.3. Zombie-verkot

Zombie-verkoilla tarkoitetaan kotikoneista muodostettua hyökkäysverkkoa, jota käytetään erilaisiin haitallisiin tekoihin ilman, että koneiden varsinaiset omistajat tai käyttäjät tietävät sitä. Mediassa muotoutunut zombie-termi kuvaa varsin hyvin tätä ongelmaa. Tyypilliseen zombie-verkkoon kuuluu noin 10 000 konetta, joita ohjaillaan irc-keskusteluohjelmilla. [Kotilainen, 2005a]

Zombie-kone muodostuu, kun käyttäjä saa koneelleen ns. bot-madon (bot, peräisin sanasta robot). Nämä madot poikkeavat muista madoista niiden monipuolisen luonteen takia: ne ovat erilaisia riippuen käyttöjärjestelmästä ja ne voivat olla automaattisia komentojonoja tai binääritiedostoja. Pääasiassa ne ovat kuitenkin Troijan hevosia, jotka avaavat etäkäytön mahdollisuuden. Etäkäyttö tarkoittaa tässä sitä, että joku ulkopuolinen, hakkeri tai rikollinen, voi ottaa Internetistä käsin käyttöönsä kotitietokoneen. Hakkerit ja rikolliset voivat etäkäyttää useita tällaisia zombie-koneita kerrallaan, muodostaen itselleen zombie-armeijan. Tämän armeijan avulla he saattavat tehdä hyökkäyksiä organisaatioiden järjestelmiin, postittaa roskapostia, levittää haittaohjelmia, kerätä tietoa kotikäyttäjien koneilta tai tarkkailla verkkoliikennettä. [Munro, 2004]

Kaikki botit eivät ole haitallisia, vaan niitä käytetään paljon myös järjestelmien automaattiseen ylläpitoon esim. irc-keskustelukanavilla. Irc on tosin myös paikka, josta hakkerit ja rikolliset zombie-koneita ohjailevat, ja josta haitalliset botit pyrkivät leviämään myös kotikäyttäjien koneisiin. [Munro, 2004]

3.4. Vakoilu- ja mainosohjelmat

Vakoilu- ja mainosohjelmat ovat kasvava ongelma. Tietoturvayhtiöt ovat heränneet niiden tarjoamaan uhkaan varsin myöhään, koska vielä jokin aika sitten erilaiset lainsäädännölliset seikat tarjosivat näille ohjelmille melko rauhallisen elintilan. Ajat ovat onneksi muuttuneet ja kehitystä on tapahtunut.

Mainosohjelmat (adware) lasketaan haittaohjelmiksi, vaikka ne eivät varsinaisesti "tee mitään pahaa". Niiden toiminta perustuu siihen, että ne asentuvat käyttäjän koneelle hänen tietämättään ja alkavat lähettää mainoksia mm. eri-

laisten ponnahdusikkunoiden muodossa. Ohjelmilla on usein myös tapana kerätä erilaisia käyttötottumuksia mainosten avulla ja näin esimerkiksi ohjata käyttäjälle tiettyjä mainoksia tai jopa pakottaa hänet selailemaan tiettyjä Internet-sivuja. Ohjelman tekee haitalliseksi juuri tuo ”ilman lupaa asentuminen”. Tästä seuraa huomio, että kaikki mainosohjelmat eivät täytä haitallisuuden tai laittomuuden kriteeriä, sillä esimerkiksi jotkin ilmaiset ohjelmat vaativat toimiakseen tällaisen mainosohjelman asentamisen. Kyse on siis vain siitä, että käyttäjälle pitää selvästi ilmoittaa mainosohjelman asentamisesta ja mahdollisten tietojen keräämisestä, jolloin käyttäjä voi itse päättää, haluaako hän hyväksyä tämän. [Knight, 2004]

Vakoiluohjelmat (spyware) ovat selvästi mainosohjelmia vaarallisempia. Ne asentuvat useiden mainosohjelmien tavoin ilman lupaa, mutta niiden päätarkoitus on vakoilla käyttäjän koneella tapahtuvia asioita ja lähettää näitä tietoja eteenpäin: näppäimistön painallukset, salasanat, luottokorttitiedot, Internet-selailu jne. Lisäksi vakoiluohjelmat tarjoavat usein muille haittaohjelmille pääsyn käyttäjän koneeseen. Näistä seikoista johtuen vakoiluohjelmista on tullut suuri uhka tavalliselle kotikäyttäjälle, koska järjestäytyneen rikollisuuden arvellaan olevan spyware-liiketoiminnan takana ja näin jopa myyvän kaapatuja tietoja tai jopa kaapattuja koneita. Kaapattuja koneita voidaan käyttää esimerkiksi hajautettuihin hyökkäyksiin zombie-armeijoiden tavoin. [Knight, 2004; Lehto, 2005]

Vakoilu- tai mainosohjelman voi saada erilaisten ohjelmien mukana tai tavalliselta Internet-sivulta.

3.5. Ohjelmistojen haavoittuvuudet

Mikään ohjelma ei ole täydellinen tai virheetön, ei myöskään tietoturva-mielessä. Tämä seuraa suoraan siitä, että osaavan ohjelmoijan on helppo tehdä ohjelma, joka suorittaa halutun toiminnon, mutta paljon vaikeampaa hänen on tehdä ohjelma, joka ei missään oloissa tekisi mitään muuta kuin juuri halutun toiminnon.

Erilaisia ohjelmistovirheitä on lukuisia, mutta tietoturvan kannalta katsottuna nämä virheet voidaan jakaa kahteen osaan: tietoisesti ja tiedostamattomasti aiheutettuihin virheisiin. Molemmat voivat aiheuttaa tietoturvaongelmia. Tietoiset ohjelmointivirheet ovat sellaisia, joissa ohjelmoija rakentaa ohjelmaan jotain haitallisia toimintoja tietoisesti, esimerkiksi viruksen tai Trojan hevosen. Tiedostamattomat virheet taas muodostuvat inhimillisen ajattelun seurauksena, esimerkiksi ohjelman loogiset virheet, ohjelmistosuunnittelussa esiintyneet virheet, validointi ja autentikointi. Näitä erilaisia ohjelmien virheitä hyödyntämällä haittaohjelmat ja muut tietoturvauhat saattavat päästä kotikäyttäjänkin koneelle. [Landwehr et al., 1994]

Microsoft on ollut usein julkisuudessa sen ohjelmista löytyneiden virheiden takia, ja samalla se on saanut heikon maineen tietoturvan kannalta. Osa maineesta on perusteltuakin, mutta Microsoft on aina pyrkinyt tekemään ohjelmistaan monipuolisia ja helppokäyttöisiä. Valitettavasti tietoturva ja käytettävyys eivät kulje käsi kädessä. Microsoft on onneksi havahtunut ongelmaan ja Bill Gatesin aloittama trustworthy computing -kampanjan tarkoituksena on ottaa tietoturva keskeisimmäksi asiaksi ohjelmia suunnitellessa ja paranneltaessa. [Järvinen, 2002b]

Suomessa Viestintävirasto ja ulkomailla CERT tiedottavat ohjelmistoista löytyneistä vakavista tietoturva-aukoista.

3.6. Roskaposti

Roskapostia arvioidaan olevan lähes kolme neljäsosaa Internetissä liikkuvasta sähköpostista. Roskapostiksi kutsutaan ei-toivottua, suurina massoina lähetettyä, ei kenellekään erityisesti kohdistettua sähköpostiviestintää. Tällaisia ovat esimerkiksi erilaiset mainokset, ketjukirjeet ja rahankeräykset, joiden määrää ja sisältöä käyttäjä ei voi itse määrätä eikä hän voi vaikuttaa niiden postituksiin. Kotikäyttäjän kannalta roskaposti tukkii sähköpostilaatikoita ja hukuttaa hyödylliset postin roskan peittoon. Varsinaisen tietoturvan kannalta roskapostit ovat pääasiassa erilaisten haittaohjelmien, esimerkiksi virusten ja troijalaisten kuljetuskanava. [Anderson, 2004]

3.7. Huijaukset

Internetin yleistyessä kaikenlainen epärehellinen toiminta on valitettavasti lisääntynyt. Esimerkkinä ovat erilaiset sähköiset asioinnit, joihin liittyy luottokorttitietojen luovuttaminen. Rikolliset yrittävät saada käsiinsä näitä tietoja, joita he myyvät eteenpäin tai käyttävät itse omiin toimiinsa. Etenkin Yhdysvalloissa tämä näyttää yleistyneen. Lisäksi toisenlaisessa huijauksessa pyritään saada käyttäjä lähettämään itse rahaa jollekin tietylle tilille. Tällaista rikollista toimintaa on vaikea havaita ja jäljittää. Toisaalta taas käyttäjät eivät usein tiedä joutuneensa huijatuiksi. [Bequai, 2000]

Henkilö- ja luottokorttitietojen kalastelu (phishing) tapahtuu esimerkiksi siten, että käyttäjälle lähetetään sähköpostiviesti, joka naamioidaan näyttämään jonkin virallisen tahon viestiltä. Tällä viestillä yritetään saada käyttäjää lähettämään jonkin palvelun salasanasensa ja käyttäjätunnuksensa, lähettämään luottokorttitietonsa ja turvakoodinsa, henkilötietonsa jne. Tällainen tietojen kalastelu voi äärimmillään olla jopa henkilökohtaista niin, että käyttäjille lähetetään yksilöityjä viestejä, jotka sisältävät esimerkiksi heidän osoitetietojaan ja nimiään. Samanlaista tietojen kalastelua tehdään myös naamioiduilla Internet-sivuilla. Tietoturvan kannalta ongelmallista on se, että huijaukset perustuvat

käyttäjän hämäämiseen ja hänen hyväuskoisuuteensa - kyseessä ei ole haitta-ohjelma. [Kotilainen, 2005b]

4. Ohjeistus

Tämän ohjeistuksen tarkoitus on tarjota edellä kuvattuihin tietoturvariskeihin ohjelmia ja menetelmiä ennaltaehkäisyyn ja riskien minimointiin. Pyrin tarjoamaan selkeät ohjeet kotikoneen hyvän tietoturvan saavuttamiseksi. En kuitenkaan mene kovin yksityiskohtaisiin ohjelmien asetuksiin tai eri menetelmien kuvauksiin, sillä pääpaino on käyttöönnotolla. Ohjeistus on suunnattu kotikäyttäjille, joilla on käytössään Windows XP -käyttöjärjestelmä ja laajakaistayhteys. Valitsin tuon käyttöjärjestelmän siksi, että se on laajalti yleistymässä juuri kotikäyttäjien keskuudessa ja se on näin ajankohtainen myös tietoturvan kannalta.

Marko Saarijärven [2004] tekemän kyselytutkimuksen mukaan kotikäyttäjät vaikuttaisivat olevan kohtuullisen hyvin suojautuneita erilaisilta tietoturvariskeiltä ja vähintäänkin perustietoturvasta näyttäisi olevan tietämystä. Tätä voidaan selittää osittain laajan tietoturvakampanjan avulla, jossa koteihin jaettiin tietoturvaoppaita.

Kotikäyttäjien tietoturvasta on esitetty paljon myös negatiivisia näkemyksiä. On tehty tutkimuksia, joissa etenkin uusimpien tietoturvariskien tapauksessa kotikäyttäjien suojautuminen olisi todella puutteellista. Tämä liittyy juuri laajakaistaisen Internet-yhteyden päässä oleviin kotikäyttäjiin. [Reiss, 2003]

Tarkoitukseni on tässä ohjeistuksessa ottaa laajempi näkökanta kuin tavanomaisissa tietoturvaoppaissa, koska niissä on usein tarjottu ohjeistus vain keskeisiin tietoturva-asioihin ja jo riskikartoituksen pohjalta tämä ei vaikuta täysin riittävältä.

Olen valinnut ohjelmat niiden suosion, henkilökohtaisen mielipiteeni, käytettävyyden, sekä ilmaisuuden takia. On huomioitavaa, että nämä ohjelmat eivät suinkaan ole ainoat mahdollisuudet vaan on olemassa paljon erilaisia tietoturvaohjelmia sekä ilmaisia, että maksullisia. Keskeisin ero ilmaisten ja maksullisten välillä on niiden monipuolisuus. Maksullisissa ohjelmissa on yleensä monia tietoturvaominaisuuksia ja paljon automaattista toimintaa. Ilmaiset vaativat yleensä hiukan enemmän käyttäjän toimintaa. Valitsin tähän ohjeistukseen kuitenkin vain ilmaisia ohjelmia, koska silloin hinta ei olisi este niiden käyttöönnotolle.

4.1. Virustorjunta

Virustorjuntaohjelman asennus kotikoneeseen on lähes välttämätöntä ja se onkin kuulunut tietoturvaoppaiden ohjeisiin alusta alkaen. Virustorjunta-

ohjelman tarkoitus on etsiä tietokoneesta viruksia, troijalaisia ja matoja. Virustorjuntaohjelmat toimivat siten, että ne etsivät edellä mainittuja haittaohjelmia erilaisten tunnistetietojen avulla. Näitä tietoja tarvitaan, jotta ohjelmat tietävät mitä etsiä. Käytännössä virustorjuntaohjelmien laatu riippuu paljon juuri tunnistetietojen ajantasaisuudesta. Virustorjuntaohjelmille on myös tyypillistä se, että ne eivät päällä ollessaan tutki muuta kuin tietokoneen muistia ja käyttäjän käyttämiä tiedostoja virusten varalta. Ne eivät silloin käy läpi kaikkia koneen tiedostoja, koska se hidastaisi konetta liikaa. Useisiin ohjelmiin onkin tätä varten rakennettu toiminto, jossa käyttäjä voi määrittää tietyn ajankohdan koko koneen virustarkistukseen. [CERT/CC, 2001; CERT/CC, 2002]

Eri valmistajilla on erilaisia virustorjuntaohjelmia. Pääsääntöisesti ne ovat maksullisia, mutta tarjolla on myös paljon ilmaisia ohjelmia. Ilmaiset ohjelmat ovat yleensä riisutumpia versioita saman valmistajan maksullisesta versiosta, jossa saattaa olla esimerkiksi muiden haittaohjelmien poistotyökaluja tai muuta automaattista tietoturvaa. Maksullisten virustorjuntaohjelmien valmistajia ovat mm. F-Secure ja Symantec. Maksullisten ja samalla ilmaisten riisuttujen versioiden valmistajia ovat puolestaan mm. Grisoft ja H+BEDV Datentechnik GmbH. Seuraavaksi käyn esimerkkinä läpi viimeksi mainitun yhtiön yksityiskäytössä ilmaisen AntiVir PersonalEdition Classic -virustorjuntaohjelman asennuksen ja käyttöönoton:

1. Kopioi AntiVir PersonalEdition Classic -virustorjuntaohjelma valmistajan kotisivulta. (<http://www.free-av.com/>)
2. Varmista ettei koneessasi ole ennestään asennettuna virustorjuntaohjelmaa. Useampi koneeseen asennettu virustorjuntaohjelma saattaa haitata toistensa toimintaa.
3. Asenna AntiVir PersonalEdition Classic -virustorjuntaohjelma kaksoisnäpäyttämällä kopioimaasi tiedostoa ja seuraa näyttöön tulevia ohjeita.
4. Asennuksen jälkeen punainen avonainen sateenvarjo tulee näkyviin tehtäväpalkkiin kellon lähelle oikeaan alakulmaan. Vie hiiren kursori tämän päälle ja paina oikeaa hiiren nappia, jolloin avautuu kuvan 1 kaltainen valikko.



Kuva 1. AntiVir PersonalEdition Classic -ohjelman valikko.

5. Kuvan 1 valikosta varmista, että *Activate AntiVir Guard* -kohdan edessä on valinta. Ohjelma on tällöin päällä.
6. Muista säännöllisin väliajoin ajaa kuvan 1 kohta *Start Internet Update*, koska tällöin ohjelmaan päivittyvät Internetistä viimeisimmät tunnistetiedot ja kotikone on näin suojattu viimeisimpiä viruksia vastaan.
7. Toisinaan aja myös kuvan 1 kohta *Start AntiVir Main Program*, jonka jälkeen eteen aukeavasta ikkunasta valitse *folders*-välilehdeltä koneesi kiintolevyt. Tämän jälkeen paina isoa suurennuslasia, jolloin ohjelma suorittaa virustarkistuksen koneen kiintolevyille ja näin myös kaikille tiedostoille.

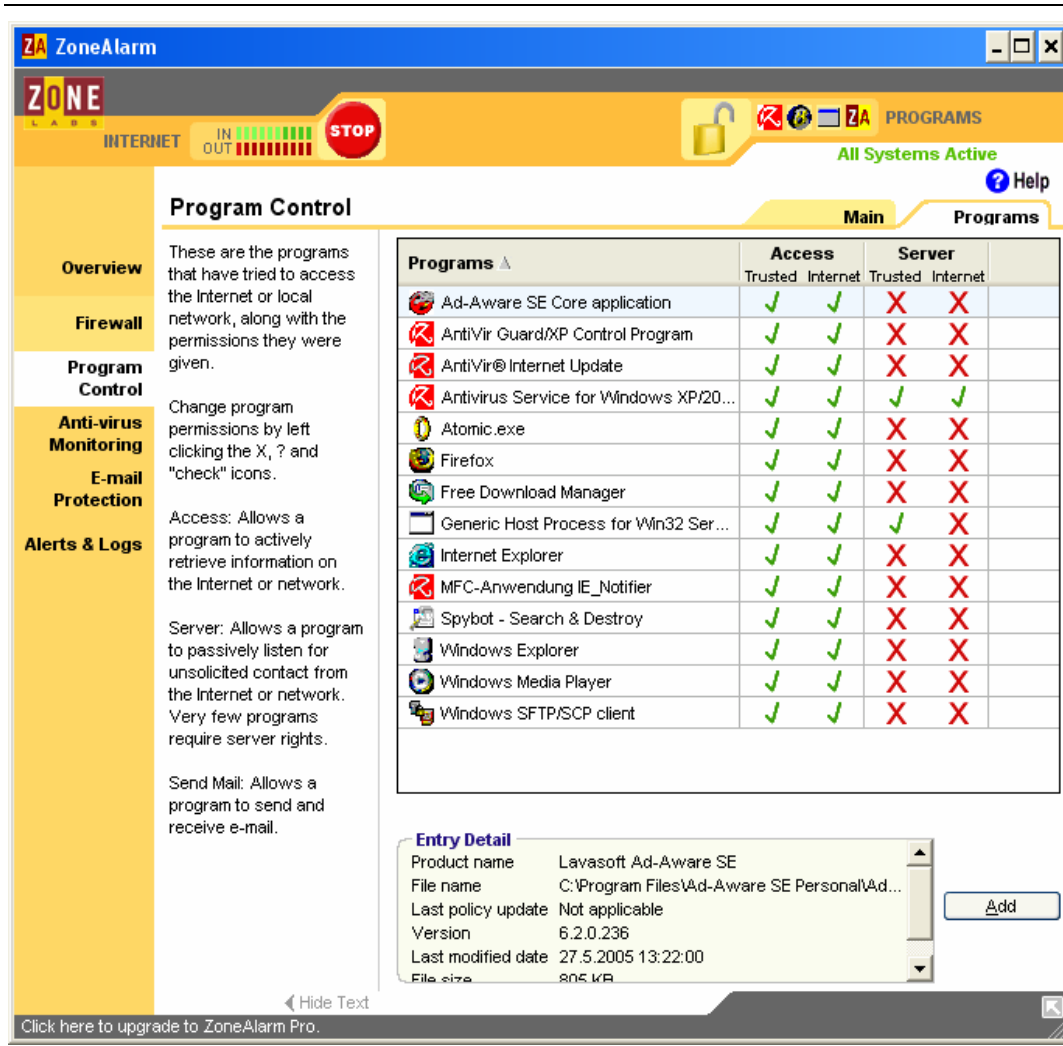
4.2. Palomuuuri

Palomuuuri on myös kuulunut tietoturvaoppaiden ohjeistuksiin jo pitkän aikaa ja sen tärkeys niissä on hyvin perusteltua. Palomuuuri on kahden verkon välissä oleva järjestelmä tai järjestelmäryhmä, joka määrittelee pääsyn kahden verkon välillä. Kotikoneessa palomuurin tehtävänä on estää koneeseen Internetistä tulevat mahdolliset hyökkäykset tai tuntemattomat yhteydenotot. Se siis kontrolloi Internetistä tulevaa liikennettä. Tällaiset ohjelmalliset kotikoneeseen asennettavat palomuurit pystyvät myös hallitsemaan kotikoneesta ulospäin lähtevää liikennettä, mikä auttaa mm. siinä, ettei kotikone muodostu osaksi zombie-armeijaa tai etteivät haittaohjelmat pääse yhdistämään Internetiin käyttäjän huomaamatta. Laitteistopohjaisissa palomuuureissa (kuten joissain ADSL-modeemeissa) ei ole tätä ominaisuutta, koska ne eivät pysty tunnistamaan ohjelmaa, joka yrittää päästä ulospäin. Ohjelmalliset palomuurit pystyvät tähän, koska ne ovat "lähempänä" ja tietokoneen sisällä. Laitteistopohjaiset ovat koneen ulkopuolella ja näin "kauempana". [CERT/CC, 2001; CERT/CC, 2002]

Palomuurien tapauksessa tietoturvyhtiöt tarjoavat virusten-torjuntaohjelmien tavoin kotikäyttäjille myös palomuuriohjelmistoja. Mukaan mahtuu maksullisia ja ilmaisia palomuuureja. Maksullisten palomuurien tapauksessa on tyypillistä se, että ne sisältävät integroituna myös virustorjuntaohjelman. Tällaisia ratkaisuja tarjoavat esimerkiksi F-Secure ja Norton. Ilmaisia riisuttuja palomuuriohjelmiä tarjoavat mm. Kerio, Sygate ja Zone Labs. Varsinkin Zone Labsin yksityiskäytössä ilmainen palomuuriohjelma ZoneAlarm on saanut suurta suosiota erilaisten tietokone-lehtiä vertailuissa ja kotikäyttäjien keskuudessa. Sen takia käyn läpi seuraavaksi juuri ZoneAlarmin asennuksen ja käyttöönoton:

1. Kopioi ZoneAlarm Free -palomuuriohjelma valmistajan kotisivulta. (<http://www.zonelabs.com/>)

2. Tarkista, ettei tietokoneessasi ole ennestään asennettuna palomuu-riohjelmia, koska kaksi palomuuriohjelmaa saattaa haitata toistensa toimintaa. Huomioitavaa on kuitenkin, että laitteistopohjainen palomuu-ri ei haittaa tietokoneeseen asennettavaa ohjelmistollista palomuuria. Se on enemmänkin lisäsuoja. Windows XP:ssä on mukana sisäänrakennettu palomuu-ri, mutta ZoneAlarm on hieman kehittyneempi tähän verrattuna. ZoneAlarmin pitäisi itse asennuksen aikana poistaa automaattisesti käytöstä XP:n oma palomuu-ri.
3. Asenna ZoneAlarm kaksoisnäpäyttämällä kopioimaasi tiedostoa ja seuraa näyttöön tulevia ohjeita. Ohjeissa käydään läpi asennusha- kemisto, käyttäjätiedot, ohjelmalisenssivaihtoehdot, perusasetukset ja ohjelman esittelyvideo. Ohjelmalisenssivaihtoeh- to-kohdassa valitaan käytettävän ZoneAlarm-ohjelman tyyppi: *pro* tai *free*. *Pro*-vaihtoehto on maksullisen ZoneAlarmin 30-päivän kokeiluversio ja toinen on ilmaisen riisutun ZoneAlarmin versio. Valitse siis tässä kohdassa pelkkä ZoneAlarm ilman *pro* päätettä. Mikäli sinulla on useita verk- kokortteja, ZoneAlarm saattaa kysyä, mitkä niistä ovat Internetiin kytkettyjä. Vastaa tähän kohtaa huolellisesti. Muuten asennuksessa muissa kohdissa tarjottavat oletusvaihtoehdot käyvät varsin hyvin. Tosin käyttäjätiedot-kohdassa voit itse miettiä haluatko rekisteröityä tai vastaanottaa uutisia ZoneAlarmiin liittyen.
4. Koneen uudelleenkäynnistyksen ja asennuksen jälkeen ZoneAlarmin pitäisi käynnistyä. Tehtäväpalkin oikeaan alakulmaan kellon lähelle tulee *za*-merkkinen logo, jota kaksoisnäpäyttämällä aukeaa kuvan 2 kaltainen ZoneAlarmin käyttöliittymä.
5. Tarkista käyttöliittymästä, että kohdan *firewall*, välilehdellä *main*, on *Internet Zone Security* -palkki kohdassa *high*. Tämä tarjoaa parhaim- man suojan Internetistä tulevalle liikenteelle. Samasta kohdasta valitse *advanced* ja merkitse aukeavasta ikkunasta valinnat *Allow Outgoing DNS/DHCP in Trusted Zone on High setting*, *Allow Outgoing DNS/DHCP in Internet Zone on High setting* sekä *Disable Windows Firewall*. Kaksi ensimmäistä asetusta on tehtävä, mikäli koneellasi on dynaaminen eli vaihtuva IP-osoite. Tarkista myös, että välilehdellä *zones* on asetettu oikealle verkkokortille *zone* sarakkeeseen Internet. Oikea verkkokortti on tässä se, jonka kautta Internetiin mennään. Tässä kohdassa voit myös asettaa tunnetun (*trusted*) verkon alueelle mahdollisen kotiverkkosi muut tietokoneet ja niiden IP-osoitteet.



Kuva 2. ZoneAlarmin käyttöliittymä.

6. Jos et halua nähdä kaikkia ZoneAlarmin ilmoituksia, vain uhkaavat hyökkäykset, voit tehdä kohdasta *alerts & logs*, välilehdeltä *main*, *Alerts Events Shown* -valinnan *off*.
7. Tärkeä kohta on käyttöliittymässä kohta *program control* ja siinä välilehti *programs*. Täällä on lista ohjelmista, joilla on pääsy (access) tai kieltä Internetiin sekä sieltä tulevaan liikenteeseen (server). Rastit kuvaavat kieltoa ja oikein-merkit sallimista. Merkkejä voi muuttaa painamalla niiden päällä hiiren vasenta painiketta. Puhekuplat, joissa kysytään jonkin tietyn ohjelman yhteydenoton sallimista tai kieltämistä, tulevat tähän listaan. Puhekuplissa kannattaa miettiä, mikä ohjelma on kyseessä ja vastata sallivasti, mikäli tietää ohjelman ja sen toiminnan. Totuus on se, että kaikkien ohjelmien ei tarvitse päästä Internetiin. Lisäksi kannattaa rastittaa kohta puhekuplasta, jossa kyseisen ohjelma aina sallitaan tai kielletään ilman uutta puhekuplaa.

Huomioitavaa on, että Windowsissa on muutamia ”tuntemattomia” ohjelmia, jotka vaativat pääsyn Internetiin, jotta kaikki muut ohjelmat toimisivat normaalisti, esim. Generic Host Process for Win32 Services kuvassa 2. On myös muutamia ohjelmia, jotka vaativat Internetistä tulevan liikenteen sallimista, kuten kuvassa 2 AntiVir. Pääsääntöisesti näitä ohjelmia on vähän ja Internetistä tulevan liikenteen sallimiseen kannattaa aina suhtautua varauksella, koska sillä avataan palomuurista tietoisesti aukko tietokoneelle. Kuva 2 antaa suuntaviivoja siitä, mitä oikeuksia ohjelmille kannattaa antaa.

4.3. Internet-selain ja sähköposti

Internetin-selain on ohjelma, jolla liikutaan Internetissä. Tällaisia ohjelmia ovat mm. Internet Explorer, Firefox ja Opera. Nämä ohjelmat ovat kaikki ilmaisia. Erottavat tekijät kohdistuvat pääasiassa ominaisuuksiin ja käytettävyyteen. Edellä mainitusta kolmikosta on syytä nostaa esille Internet Explorer, joka on saanut varsin negatiivisen kuvan tietoturvan kannalta. Explorerin ongelmana on nimittäin sen läheinen yhteys käyttöjärjestelmään, jonka seurauksena selaimesta löytyvät haavoittuvuudet voivat aiheuttaa varsin ikäviä seurauksia. Ohjelman kehittäjä Microsoft on kyllä havahtunut haavoittuvuuksiin ja tarjonnut päivityksiä niihin, mutta siitä huolimatta eri tietoturvatahot kuten Viestintävirasto tai CERT suosittelevat Internet-selaimeksi kolmikon kahta muuta ohjelmaa. Keskeisin syy tähän on juuri Explorerin läheinen yhteys käyttöjärjestelmään ja se, että varsin laaja osa kotikäyttäjistä käyttää juuri tätä Windows XP:n mukana tulevaa Internet-selainta.

Firefox on saanut paljon suosiota Internetin käyttäjien keskuudessa, varsinkin tietoturvan suhteen. On kuitenkin syytä muistaa, että Firefoxistakin on löydetty erilaisia haavoittuvuuksia eikä se tietoturvamielessä ole millään tavalla täydellinen ohjelma. On siis hyvä pitää mielessä ohjelmien jatkuva päivittäminen, niin Firefoxin kuin Internet Explorerinkin tapauksessa.

Firefoxin suosioon on vaikuttanut myös sen helppo laajennettavuus. Ohjelmaan löytyy nimittäin paljon erilaisia ilmaisia laajennuksia, jotka monipuolistavat ohjelmaa: mainosten esto, irc, pelejä, säätiedotus, sanakirja ja uutisten lukija. (<http://www.mozilla.com/firefox/>)

Sähköpostiin ja sieltä saatuihin viesteihin liittyen kannattaa suorittaa seuraava viisivaiheinen tietoturvatesti:

1. Onko sähköpostitse saamasi viesti peräisin joiltain tuntemaltasi henkilöltä?
2. Oletko saanut sähköpostia viestin lähettäjältä aikaisemmin?
3. Odotitko tätä viestiä tai mahdollista liitetiedostoa tältä lähettäjältä?
4. Onko viestissä mitään järkeä?

5. Havaitisiko virustorjuntaohjelma sähköpostiviestin puhtaaksi?

Mikäli vastasit johonkin kohtaa kielteisesti, on saamasi viesti hyvin todennäköisesti roskaposti tai haittaohjelman levitysyritys, joka kannattaa poistaa heti. Mikäli vastasit kaikkiin myöntävästi, viestiin kannattaa silti suhtautua suurella varovaisuudella ja etenkin sen sisältämien liitetiedostojen avaamista heti tulisi välttää. [CERT/CC, 2002]

4.4. Vakoilu- ja mainosohjelmien torjunta

Vakoilu- ja mainosohjelmien torjuntaan ja poistamiseen tarvitaan erillisiä ohjelmia, sillä muut torjuntaohjelmat, kuten virustorjuntaohjelmat, eivät pääsääntöisesti havaitse näitä uhkia. Lisäksi tämänkaltaiset haittaohjelmat pyrkivät piilottamaan itsensä käyttöjärjestelmän eri paikkoihin, kuten rekisteriin, mikä tekee niiden poistamisesta ja havaitsemisesta vaikeaa.

Markkinoilla on paljon erilaisia ohjelmia, jotka etsivät ja poistavat tietokoneelta vakoilu- ja mainosohjelmia. Mukaan mahtuu sekä maksullisia että ilmaisia ohjelmia. Maksullisissa ohjelmissa on mukana automaattinen päivitys sekä aina päällä oleva torjunta. Ilmaisissa ohjelmissa päivitykset ja tarkistukset joutuu tekemään manuaalisesti. Päivityksiin liittyen on huomioitavaa, että vakoilu- ja mainosohjelmat kuuluvat vielä varsin uusiin uhkiin. Tämän takia ohjelmien tunnistetiedot (vrt. virusohjelmien tunnistetiedot) poikkeavat eri valmistajien kesken usein varsin paljon. Eri ohjelmat löytävät eri vakoilu- ja mainosohjelmat. Tähän liittyvää standardointia ja yhteistyötä ei ole vielä olemassa laajassa mittakaavassa. Tämän takia onkin hyvä asentaa kotikoneelle vähintään kahden eri valmistajan ohjelma, jotta käyttöön saataisiin mahdollisimman laajat eri vakoilu- ja mainosohjelmien tunnistetiedot.

Esittelen seuraavaksi kahden vakoilu- ja mainosohjelmien torjuntaan erikoistuneen ilmaisen ohjelman asennuksen ja käyttöönoton. Ohjelmat ovat SpyBot – Search & Destroy ja Ad-Aware SE Personal, jotka ovat saavuttaneet melko suuren suosion omassa ohjelmakategoriassaan. Ad-Awaresta on myös tarjolla erilaisia maksullisia monipuolisempia versioita.

SpyBot-ohjelman asennus ja käyttöönotto tapahtuvat seuraavasti:

1. Kopio SpyBot valmistajan kotisivulta. (<http://www.safer-networking.org/>)
2. Asenna ohjelma kaksoisnäpyttämällä kopioimaasi tiedostoa ja seuraa näyttöön tulevia ohjeita. Asennuksessa pyydetään lisenssisopimuksen hyväksymistä, asennushakemistoa sekä ohjelman asennuksen tyyppiä. Asennustyyppiin kannattaa valita *Full installation* (täysi asennus).
3. Asennuksen jälkeen käynnistä ohjelma sen kuvakkeesta, mikäli asennuksen jälkeen ohjelma ei käynnisty suoraan.

4. Ensimmäisellä käynnistyskerralla ohjelma käy pienessä ikkunassa läpi muutamia keskeisiä asetuksia, mutta voit sivuuttaa ne painamalla useaan kertaan *next*-painiketta ja lopuksi *start using program* -painiketta.
5. Ohjelman pääikkuna on kuvan 3 kaltainen. Voit valita ylälälikosta (*Language*) ohjelman kieleksi suomen. Tämän jälkeen tarkista, että ylälälikon *Mode*-kohdassa on valinta *Default mode*. Tämä asettaa SpyBot-ohjelman yksinkertaiseen tilaan, jossa kaikkia toimintoja ja asetuksia ei näytetä käyttäjälle. *Advanced mode* -tilassa käyttäjälle näytetään kaikki. Kannattaa siirtyä *Advanced mode* -tilaan, kun ohjelmaa on oppinut käyttämään ja se on tullut tutuksi. Kehittyneempi tila tarjoaa monia säätöjä, jotka lisäävät Windowsin tietoturvaa. Tuohon tilaan kannattaa tutustua ohjelmassa olevan ohjeen avulla.



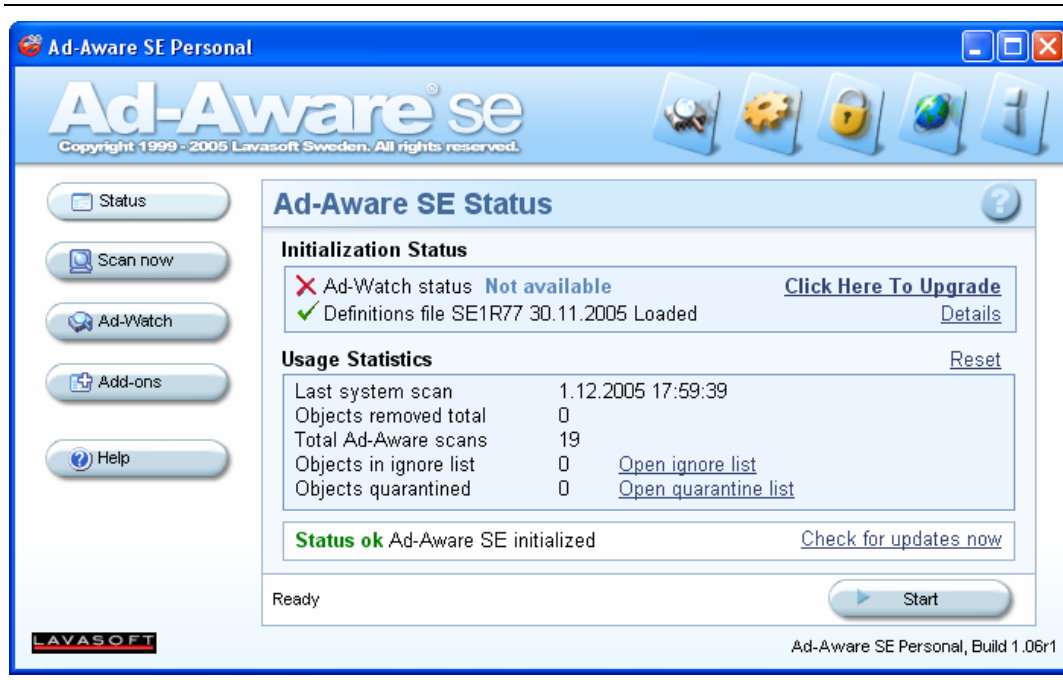
Kuva 3. SpyBot – Search & Destroy -ohjelman pääikkuna.

6. Päivitä ohjelma ja sen tunnistetiedot aina säännöllisin väliajoin valitsemalla vasemmalta *päivitä* ja tämän jälkeen muuttuneesta pääikkunasta *etsi päivityksiä*. Ohjelma etsii näin Internetistä päivityksiä ja ilmoittaa niistä pääikkunaan ilmestyvässä listassa. Mikäli päivityksiä on, rastita tästä listasta kaikki ja paina tämän jälkeen *lataa päivitykset* -painiketta. Ohjelma lataa näin päivitykset Internetistä.

7. Päivityksen jälkeen valitse vasemmalta kohta *immuniteetti*. Paina nyt muuttuneesta pääikkunasta painiketta *ota käyttöön*. Samoin tarkista, että *IEHELPER-BHO* -kohdassa on rasti *Enable permanent blocking of bad addresses in Internet Explorer* -valinnan edessä. Tähän liittyvässä vetovalikossa pitäisi myös olla valittuna kohta *estä automaattisesti*. *Immuniteetti*-kohdan valinnat liittyvät Internet Explorerin erilaisten komponenttien suojaustoimenpiteisiin, jotka kannattaa tehdä, vaikkei tuota selainta käyttäisikään.
8. Valitse pääikkunasta nyt vasemmalta kohta *Search & Destroy* ja paina painiketta *aloita*. Ohjelma alkaa nyt etsiä tietokoneelta mahdollisia vakoilu- ja mainosohjelmia. Tämä toimenpide kannattaa tehdä säännöllisin väliajoin yhdistettynä ohjelman päivittämiseen (kohta 6). Mikäli listaan ilmestyy punaisia merkintöjä, on ohjelma löytänyt koneelta mahdollisia vakoilu- ja mainosohjelmia. Voit rastittaa nämä merkinnät ja valita *korjaa ongelmat*, jolloin ohjelma yrittää poistaa kyseiset ongelmat koneesta. Kannattaa kuitenkin muistaa, että nämä merkinnät eivät aina ole oikeassa, sillä jotkin ohjelmat vaativat toimiakseen jotain SpyBotin punaisiksi merkitsemiä komponentteja. Tämä on esimerkiksi totta F-Securen virustorjuntaohjelman kanssa, joka käyttää BackWeb lite -nimistä ohjelmaa, jonka SpyBot tulkitsee haitalliseksi. Mikäli jokin ohjelma siis ei enää toimi kunnolla SpyBotin korjauksien jälkeen, kannattaa kokeilla palauttaa kohdasta *varmuuskopiot* edellinen toimiva tila takaisin.

Ad-Aware -ohjelman asennus ja käyttöönotto puolestaan tapahtuvat seuraavasti:

1. Kopioi Ad-Aware SE Personal valmistajan kotisivulta. (<http://www.lavasoftusa.com/software/adaware/>)
2. Asenna ohjelma kaksoisnäpyttämällä kopioimaasi tiedostoa ja seuraa näyttöön tulevia ohjeita.
3. Käynnistä ohjelma sen kuvakkeesta, jolloin saat eteesi kuvan 4 kaltaisen ikkunan. Tämä on ohjelman pääikkuna.
4. Päivitä ohjelman tunnistetiedot painamalla oikeasta yläkulmasta maapallon kuvaa. Tämä avaa uuden ikkunan, jossa ohjelma on valmis ottamaan yhteyden Internetiin. Paina tässä ikkunassa painiketta *connect*. Ohjelma ottaa yhteyden Internetiin ja etsii tunnistetietojen päivityksiä. Mikäli päivityksiä löytyi, hyväksy niiden kopiointi ja asennus. Paina *finish*-painiketta sulkeaksesi päivitysikkunan. Muista päivittää tunnistetiedot usein.



Kuva 4. Ad-Aware SE Personal -ohjelman pääikkuna.

5. Paina pääikkunasta painiketta *Scan now*, joka muuttaa ikkunan sisältöä hieman. Ohjelma on nyt valmis etsimään vakoilu- ja haitta-ohjelmia tietokoneesta. Valitse *Select a scan mode* -kohdasta joko *Perform smart system scan* tai *Perform full system scan*. Ensimmäinen valinta tarkoittaa sitä, että tietokoneelta etsitään vakoilu- ja mainos-ohjelmia yleisimmistä tunnetuista paikoista. Toinen valinta etsii näitä koko tietokoneesta ja on huomattavasti hitaampi vaihtoehto. Kannattaa käyttää ensimmäistä nopeampaa vaihtoehtoa usein ja jälkimmäistä hitaampaa vaihtoehtoa hieman harvemmin. Paina *Next*-painiketta, joka käynnistää etsinnän. Etsinnän päätyttyä paina vielä *Next*-painiketta, jolloin ohjelma näyttää tietokoneelta löydetty ongelmat punaisella merkinnällä. Valitse punaiset merkinnät ja paina *Next*-painiketta, jolloin ohjelma pyytää vahvistuksen poistotoimenpiteeseen. Mikäli punaisia merkintöjä ei ole, voit jatkaa painamalla *Next*-painiketta, mikä johtaa ohjelman takaisin etsinnän aloittamisen kohtaan.

4.5. Ohjelmistojen päivitys

Ohjelmistojen päivitys tarkoittaa ohjelmissa esiintyvien haavoittuvuuksien ja vikojen paikkaamista sekä ohjelman parantamista. Tämä tapahtuu asentamalla ns. korjaustiedostoja tai kokonaan uuden version ohjelmasta. Ohjelmistojen valmistajat tarjoavat yleensä kotisivuillaan edellä mainittuja korjaustiedostoja,

jotka ovat pääosin ilmaisia. Usein ohjelmissa on myös automaattisia päivitystoimintoja, jotka hakevat ilman käyttäjän häiritsemistä uudet päivitykset Internetistä. Tosin aina ongelmat eivät tule korjatuksi, vaan korjaustiedosto saattaa aiheuttaa uusia vikoja tai haavoittuvuuksia. Kuitenkin tietoturvan kannalta on todella tärkeää päivittää käyttämiään keskeisimpiä ohjelmia säännöllisin väliajoin. Näihin ohjelmiin kuuluvat oleellisesti virustorjuntaohjelmat ja niiden tunnistetiedot, palomuuriohjelma, vakoilu- ja mainostenpoisto-ohjelmat, Internet-selaimet, sähköpostiohjelmat sekä toimisto-ohjelmat. Käytännössä päivityksen piiriin kuuluvat kaikki tietoturvaohjelmat sekä sellaiset ohjelmat, joita käytetään usein. On siis syytä pitää ohjelmien mahdollinen automaattinen päivitystoiminto päällä tai muutoin käydä säännöllisin väliajoin manuaalisesti tarkistamassa päivitykset. [CERT/CC, 2001; CERT/CC, 2002]

Lisäksi erityisen tärkeää on pitää käyttöjärjestelmää päivitettyinä, koska se toimii kaiken toiminnallisuuden keskuksena ja välittäjänä, ja se on myös haittaohjelmien keskeisimpänä kohteena. Kotikäyttäjän Windows XP -käyttöjärjestelmästä löytyy automaattinen päivitystoiminto, jonka käyttöönotto tapahtuu seuraavasti:

1. Valitse *käynnistä* ja *ohjauspaneeli* Windowsin valikosta.
2. Täältä valitse *automaattiset päivitykset*, jolloin avautuu kuvan 5 kaltainen ikkuna.
3. Valitse kuvan 5 ikkunasta kohta *automaattinen* ja valitse kellonaika, jolloin haluat päivitysten asentuvan. Päivitykset kannattaa lisäksi valita asennettavaksi *joka päivä*.
4. Paina *käytä* ja *OK*. Nyt Windows hakee ja asentaa Internetistä automaattisesti päivityksiä käyttöjärjestelmään.



Kuva 5. Windows XP:n automaattinen päivitys.

4.6. Salaus, salasanat ja varmistus

Tärkeiden työtiedostojen varmuuskopioiminen ja salaus vähentävät merkittävästi tiedostojen väärinkäyttöön kohdistuvaa uhkaa sekä takaavat samalla nopean palautumisen ongelmatilanteissa. Varmuuskopio tarkoittaa tärkeän tiedoston kopioimista toiseen paikkaan, kuten CD- tai DVD-levylle. Salaus puolestaan tarkoittaa tämän tärkeän tiedoston salaamista erityisellä salausalgoritmilla, eli tiedoston käyttöön ja avaamiseen tarvitaan tietty salasana. Salaus koskee alkuperäistä tärkeää tiedostoa, mutta myös varmuuskopiota.

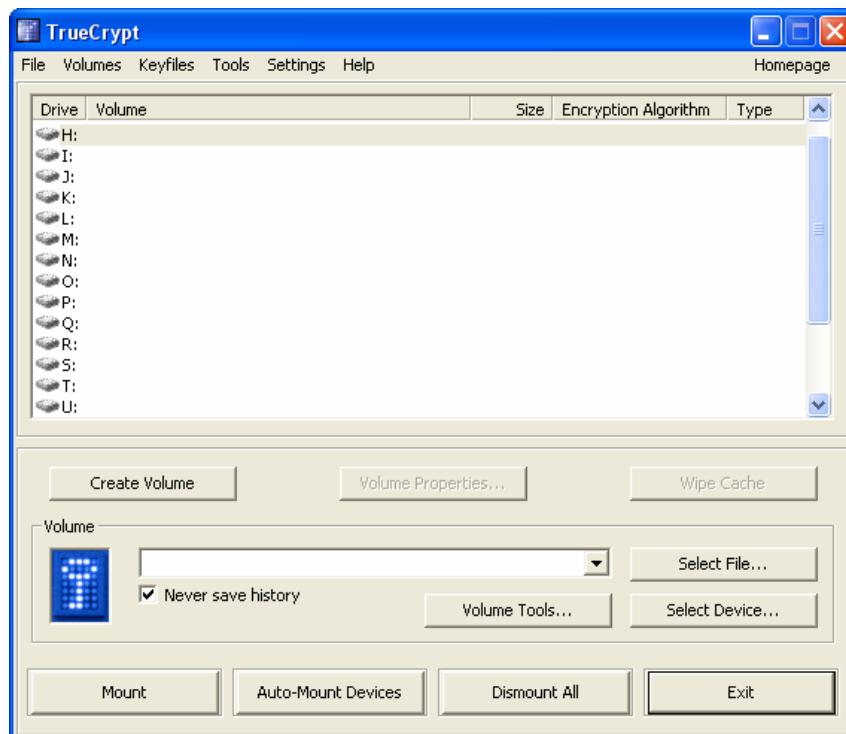
Salaus on tärkeää myös Internetin käytön yhteydessä, sillä esimerkiksi pankkiyhteydet ovat salattuja. Tämä on hyvä pitää mielessä, koska Internet-selaimissa yhteyden salausta on kuvaamassa lukko-symboli, joka sijaitsee tilapalkissa. Luottokorttitietojaan ei tulisi lähettää ilman, että yhteys on salattu. Laajemmin salaus on mahdollista tehdä jopa sähköpostille ja langattomalle WLAN-kotiverkkolle.

Salasanan keksimisessä pitäisi ottaa huomioon sen pituus ja kompleksisuus. Joidenkin lähteiden mukaan hyvän salasanan pitäisi olla vähintään 8 merkkiä pitkä ja toisissa lähteissä sanotaan lukemaksi 14 merkkiä. Hyvänä sääntönä voi

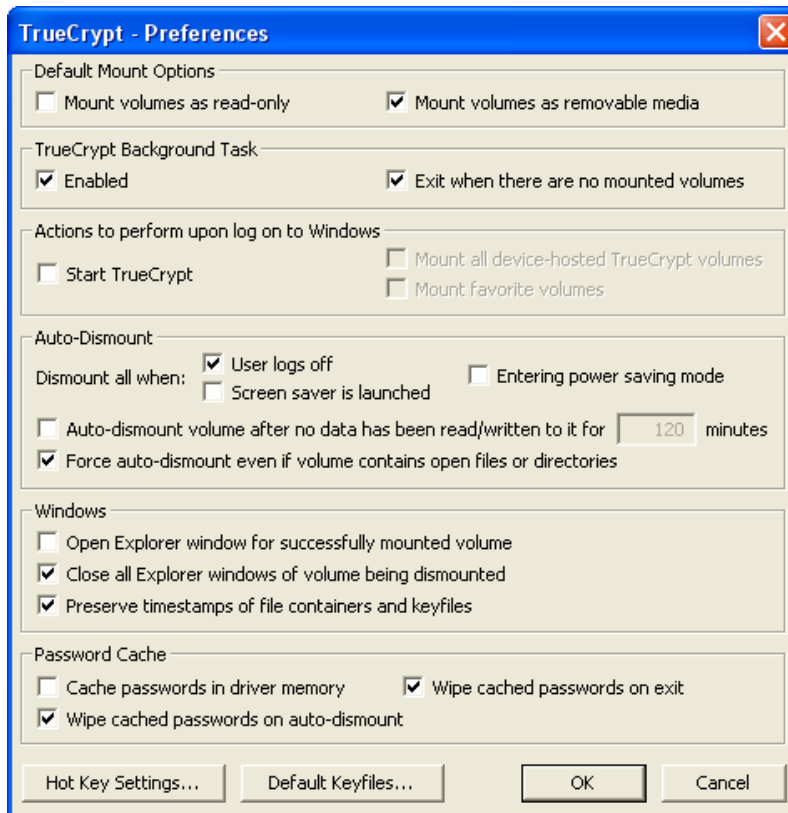
pitää sitä, että mitä pidempi ja monimutkaisempi salasana sitä vaikeampi sitä on arvata. Salasanoja käytetään nykyään monissa järjestelmissä. Hyvänä esimerkkinä on kotikäyttäjän Windows XP -käyttöjärjestelmään kirjautuminen, jossa myös salasanan tulisi olla pitkä ja vaikeasti arvattava. Valitettavasti ihmiset usein laittavat salasanaan helpposti arvattaviksi tai käyttävät samaa salasanaa eri paikoissa. Näin heikoksi kohdaksi tietoturvassa muodostuu juuri käyttäjä eikä itse järjestelmä. [Järvinen, 2002a]

Eräs hyvä ilmainen ohjelma tiedostojen salaamiseen on TrueCrypt. Tämä ohjelman salaamisidea perustuu siihen, että se luo erityisen salatun säiliön tiedostoja varten. Tämä säiliö voi olla käyttäjän valitseman kokoinen, jolloin sinne mahtuu eri määrä tiedostoja. Merkittävä idea säiliöiden käytössä on se, että säiliön saa auki vain salasanalla. Lisäksi sitä on mahdollista käyttää tavallisen massamuistin tavoin. Tämä tarkoittaa sitä, että säiliö tulee näkyviin Windows XP:n Resurssienhallintaan uutena levyasemana, johon on helppo siirtää, käyttää tai tallettaa omia tiedostojaan. Säiliö on myös erityisen helppo varmuuskopioida, koska se on itsessään vain yksi iso tiedosto. Tämä tiedosto on helppo siirtää CD:lle, DVD:lle tai USB:lle, kunhan on vain säiliön luonnin yhteydessä tehnyt tarvittavan kokoisin säiliön. Ohjelmalla voi myös salata kokonaisia kiintolevyjä. TrueCrypt-ohjelman asennus ja käyttöönotto tapahtuvat seuraavasti:

1. Kopioi TrueCrypt-ohjelman Windows XP -versio valmistajan kotisivulta. (<http://www.truecrypt.org/>)
2. Sinun pitää purkaa kopioimasi zip-tiedosto esim. IZArc-pakkausohjelmalla. (<http://www.izarc.org/>)
3. Kaksoisnäpäytä puretusta zip-tiedosta tullutta exe-asennustiedostoa käynnistääksesi asennusohjelman.
4. Seuraa näyttöön tulevia ohjeita, joissa valitaan ohjelman asennushakemisto sekä muita yleisiä asetuksia. *Install*-painike asentaa ohjelman.
5. Käynnistä TrueCrypt-ohjelma kuvakkeesta, jolloin saat eteesi kuvan 6 kaltaisen ikkunan. Tämä on ohjelman pääikkuna.
6. Ohjelman asetukset kannattaa tarkistaa valitsemalla valikosta *Settings* ja *Preferences*. Kuva 7 on esimerkkinä sopivista asetuksista.



Kuva 6. TrueCrypt-ohjelman pääikkuna.



Kuva 7. TrueCrypt-ohjelman asetukset.

7. Salatun säiliön luonti tapahtuu painamalla *Create Volume* -painiketta. Tämä avaa eteen uuden ikkunan, josta valitaan *Create a Standard TrueCrypt volume* ja painetaan *Next*-painiketta.
8. Valitse avautuneesta uudesta ikkunasta *Select File*, jossa valitaan säiliön sijainti ja nimi. Kirjoita nimi ja lisää nimen perään .tc-tunniste (tiedostonnimi.tc). Paina tämän jälkeen *Next*-painiketta.
9. Eteen avautuu jälleen uusi ikkuna, jossa valitaan säiliön salausalgoritmi. Salausalgoritmeista AES ja Blowfish kuuluvat yleisesti käytetyimpiin. Ohjelmalla on myös mahdollista käyttää useita salausalgoritmeja yhtäaikaisesti, joka kyllä lisää tietoturvaa mutta hidastaa hieman säiliön käyttöä. Esimerkkinä voidaan valita salausalgoritmiksi *AES-Twofish* ja ns. avainalgoritmiksi (tämä valinta ei ole niin tärkeä) *SHA-1*. Paina tämän jälkeen *Next*-painiketta.
10. Seuraavaksi valitaan säiliön koko. Koko on hyvä valita sellaiseksi, että sen varmuuskopioiminen uudelle medialle onnistuisi kapasiteetin kannalta, kuten CD-levylle 650 Mt tai 700 Mt ja DVD-levylle 4,3 Gt. Paina *Next*-painiketta valittuasi säiliön koon.
11. Säiliön salasana pitää kirjoittaa seuraavaksi. Kirjoittaminen pitää tehdä salasanalle varattuihin riveihin. Salasanan tulisi olla pitkä ja vaikeasti arvattava. Sen pitäisi sisältää isoja ja pieniä kirjaimia sekä erikoismerkkejä. Ohjelma suosittelee salasanan pituudeksi vähintään 20 merkkiä. Mitä pidempi salasana on, sen parempi! Tässä kohtaa voi myös käyttää ns. avaintiedostoa, joka vaaditaan aina säiliötä avattaessa. Avaintiedosto voi olla mikä tahansa tiedosto. Sen käyttö ei ole pakollinen, joten voit huoletta jättää rastittamasta tuota kohtaa. Salasanan kirjoittamisen jälkeen paina *Next*-painiketta.
12. Seuraavaksi ohjelma tarvitsee satunnaislukuja, jotka muodostetaan hiiren liikkeillä. Hiirtä kannattaa pyöritellä ikkunassa jonkin aikaa, jolloin satunnaisluku olisi mahdollisimman satunnainen. Tämän jälkeen paina *Format*-painiketta, joka alustaa säiliön käyttöä varten.
13. Säiliö on nyt luotu. Voit luoda uuden säiliön painamalla *Next*-painiketta tai palata ohjelman pääikkunaan *Exit*-painikkeen kautta.
14. Säiliön avaaminen tapahtuu ohjelman pääikkunasta valitsemalla ensin levykirjain näkyvillä olevasta listasta. Tämän jälkeen painetaan *Select File* -painiketta, josta valitaan itse säiliö. Näiden valintojen jälkeen painetaan vielä *Mount*-painiketta, jonka jälkeen ohjelma kysyy säiliön salasanan. Salasanan syöttämisen jälkeen säiliön sisältö näkyy Windowsin Resurssienhallinnalla uutena levynä.

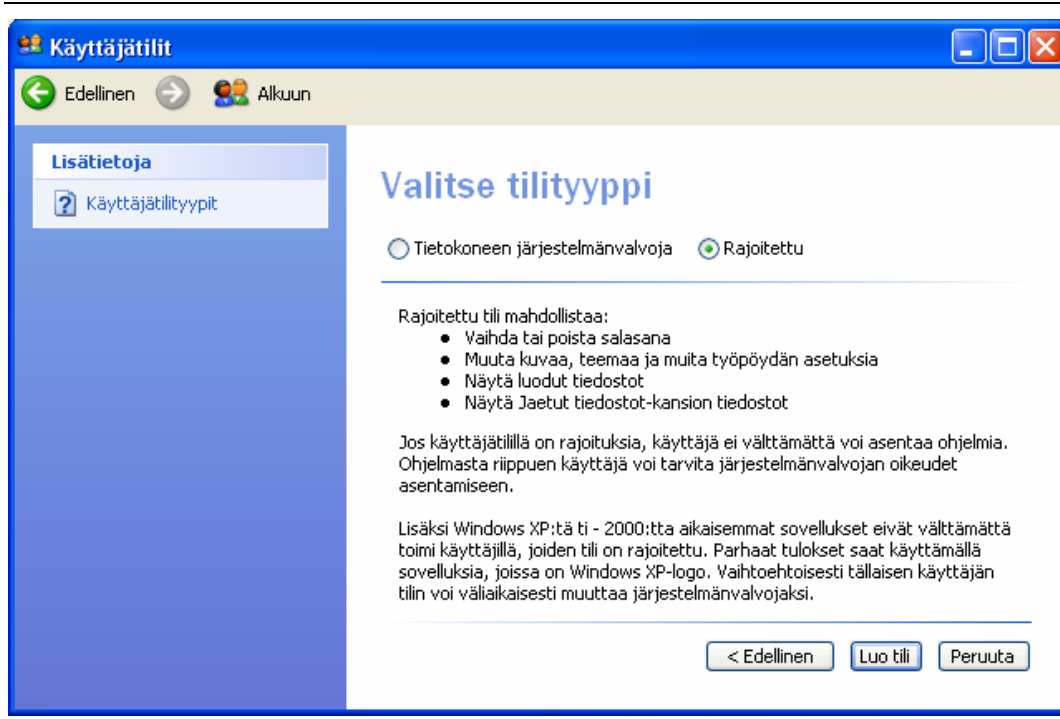
15. Kannattaa myös tutustua ohjelman ohjetiedostoon, joka tarjoaa paljon opastusta ja uusia ominaisuuksia ohjelman käyttöön. Tämä ohje löytyy pääikkunan valikosta valitsemalla *Help* ja *User's Guide*.

4.7. Rajoitettu käyttäjätili

Yritys- ja palvelinkäytössä tunnetuksi tulleissa Unix- ja Linux-käyttöjärjestelmissä on ollut alusta alkaen käytössä ns. käyttäjätilit ja niihin liittyvät käyttöoikeudet. Käyttäjätili on tietokoneiden tapauksessa kuin pankkitili, johon liittyy erilaisia tietoja, asetuksia, tunnus ja salasana sekä erilaisia käyttöoikeuksia; toimintoja, joita tilin käyttäjällä on tai ei ole lupa tehdä. Esimerkkinä voisi olla käyttäjätili, jolla ei ole oikeutta asentaa mitään ohjelmia tietokoneeseen, mutta on kuitenkin oikeus käyttää kaikkia koneeseen asennettuja ohjelmia. Tämän kaltainen käyttäjätiliominaisuus ei kotikäyttäjän Windows XP -käyttöjärjestelmässä ole juuri lainkaan käytössä, vaikka ominaisuus onkin rakennettu siihen mukaan. Suurin osa kotikäyttäjistä käyttääkin Windows XP:tä ns. järjestelmänvalvojan oikeuksin, jolla on kaikki oikeudet poistaa mitä tahansa tiedostoja ja tehdä periaatteessa tietokoneella mitä haluaa. Tästä on seurauksena se, että mikäli koneeseen tulee virus tai muu haittaohjelma, on sillä täysin samat oikeudet kuin käyttöjärjestelmään kirjautuneella käyttäjälläkin. Suojauskeino tähän olisi se, että Windowsissa luotaisiin rajoitettu käyttäjätili, jolla ei olisi muita oikeuksia kuin käyttää ohjelmistoja ja tallentaa niillä luotuja tiedostoja. Tätä tiliä käytettäisiin normaalissa käytössä ja töitä tehdessä. Rajoitetun käyttäjätilin käyttäminen lisää merkittävästi tietoturvaa ja siksi esimerkiksi Linux on saanut turvallisemman maineen verrattuna Windowsiin.

Kotikäyttäjän Windows XP -käyttöjärjestelmässä rajoitetun käyttäjätilin luominen tapahtuu seuraavasti:

1. Valitse *käynnistä*, *ohjauspaneeli* ja sieltä *käyttäjätilit*. Näyttöön avautuu ikkuna, jossa on lueteltu tietokoneessa olevat käyttäjätilit sekä niiden tyypit. Erilaisia tyyppejä ovat esimerkiksi juuri järjestelmänvalvoja ja rajoitettu käyttäjätili.
2. Valitse ikkunasta *luo uusi käyttäjätili*, jonka jälkeen sinulta kysytään luotavan tilin nimeä. Kirjoita haluamasi nimi tähän ja paina *seuraava-painiketta*.
3. Ikkuna muuttuu kuvan 8 kaltaiseksi, jossa valitaan käyttäjätilin tyyppi. Valitse tässä kuvan 8 kaltaisesti *rajoitettu* ja paina *luo tili-painiketta*.



Kuva 8. Rajoitetun käyttäjätilin luonti Windows XP:ssä.

4. Rajoitettu käyttäjätili on nyt luotu ja voit muuttaa sen ominaisuuksia menemällä ensin kohtaan 1 ja tämän jälkeen valitsemalla luodun käyttäjätilin. Ominaisuuksista voit muuttaa tilin nimen, kuvan, salasanan ja tyyppin. Suositeltavaa on asettaa tilille heti salasana, jonka tulisi olla pitkä ja vaikeasti arvattava.
5. Rajoitettu käyttäjätili on nyt valmis käytettäväksi ja voit kokeilla sitä kirjautumalla ensin pois nykyisellä käyttäjällä valitsemalla *käynnistä* ja *kirjaudu ulos*, ja tämän jälkeen kirjautumalla Windowsiin rajoitetulla käyttäjätilillä.

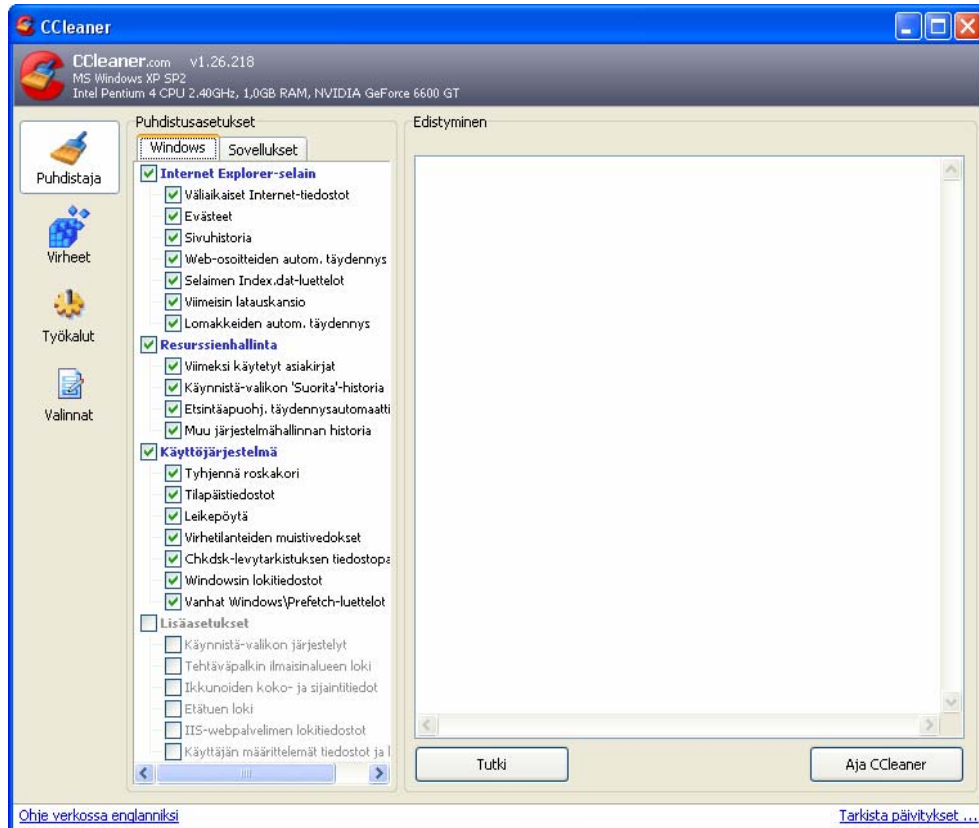
Rajoitettuun käyttäjätiliin liittyy tosin muutamia ikäviä ongelmia, sillä varsinkin Windows XP:ssä jotkut ohjelmat eivät välttämättä toimi lainkaan tiliä käytettäessä. Tämä johtuu usein ohjelmien iästä tai siitä, että niitä ei ole suunniteltu toimivaksi rajoitetuilla oikeuksilla. Onneksi esim. Microsoftin toimisto-ohjelmat ja eri Internet-selaimet toimivat hyvin rajoitetun käyttäjätilin alaisina. Lisäksi rajoitettuun käyttäjätiliin liittyy pieni käytettävyysongelma, koska aina kun käyttäjä haluaa asentaa esim. uuden ohjelman tietokoneeseen, joutuu hän kirjautumaan järjestelmävalvojana ennen kuin hän voi toimenpiteen tehdä. Muutenkin kaikki tietokoneen tärkeiden asetusten muuttaminen, kuten uusien laitteiden asennus ja konfigurointi, vaativat järjestelmävalvojan oikeuksia. Rajoitettu käyttäjätili lisää tietoturvaa, mutta hankaloittaa hieman käytettävyyttä.

4.8. Yksityisyys

Kotikäyttäjän Windows XP -käyttöjärjestelmässä on paljon ohjelmia, jotka jättävät jälkeensä erilaisia tietoja kuten työtiedostojen väliaikaiskopioita, järjestelmän tietoja ja ohjelmien käyttötilastoja. Jotkin näistä tiedoista saattavat sisältää käyttäjälle yksityisiä tietoja, kuten juuri työtiedostojen väliaikaiskopiot. Useat ohjelmat eivät poista näitä tietoja tietokoneen sammuttamisen jälkeen, vaan ne jäävät koneen kiintolevyille. Eri haittaohjelmat yrittävät myös tutkia näitä ohjelmien jälkeen jättämiä tietoja ja mahdollisuuksien mukaan lähettellä niitä eteenpäin. Hyvä esimerkki tästä on Internet-selaimen väliaikaistiedostot, jotka voivat sisältää salasanoja ja käyttäjätunnuksia Internetissä liikkumisen jäljiltä. Tietojen väärinkäytökset ovat selkeä uhka ja sen takia tietokonetta tulisi pitää erityisenä sähköisenä lompakkona. [Hinde, 2003]

Edellä mainittujen väliaikaistiedostojen poistamisen voi tehdä toki täysin manuaalisesti, mutta se on varsin aikaa vievää. Automaattiseen poistamiseen on hyvä käyttää siihen suunnattua ilmaista ohjelmaa, kuten CCleaneriä. CCleaner-ohjelman asennus ja käyttöönotto tapahtuvat seuraavasti:

1. Kopioi CCleaner valmistajan kotisivulta. (<http://www.ccleaner.com/>)
2. Asenna ohjelma kaksoisnäpäyttämällä kopioimaasi tiedostoa ja seuraa näyttöön tulevia ohjeita. Ohjeissa kysytään ohjelman kieli, asennushakemisto sekä haluatko ohjelman esiintyvän roskakorin valikossa.
3. Asennuksen jälkeen käynnistä ohjelma sen kuvakkeesta, jolloin saat eteesi kuvan 9 kaltaisen ikkunan.
4. Ikkunasta kannattaa heti valita kohta *valinnat*, jonka jälkeen painaa ilmestyvää kohtaa *asetukset*. Asetuksista kannattaa rastia kohta *puhdistaa automaattisesti koneen käynnistyessä*. Tämä valinta tekee sen, että ohjelma puhdistaa koneen väliaikaistiedostoista aina käynnistymisen yhteydessä.
5. Valitse jälleen kuvan 9 näkymä valitsemalla vasemmalta kohta *puhdistaja*. Tarkista, että *windows*-välilehdellä on rastitettu kuvan 9 kaltaiset valinnat. Tarkista samalla myös välilehdestä *sovellukset*, että kaikissa kohdissa on rasti. Nämä välilehdet kuvaavat niitä tietoja ja ohjelmia, joita ohjelma puhdistaa tietokoneelta.
6. Paina alhaalta painiketta *Tutki*, joka tutkii tietokoneessa olevia "turhia" tiedostoja ja näyttää löytämänsä tiedostot isossa ikkunassa. Aja CCleaner -painikkeella ohjelma konkreettisesti puhdistaa eli poistaa kaikki löytämänsä edellä mainitut "turhat" tiedostot.



Kuva 9. CCleaner-ohjelma.

7. Ohjelmassa on myös mahdollisuus puhdistaa tietokoneen rekisteri virheellisistä viittauksista valitsemalla ensin kohta *virheet* ja sieltä painamalla painiketta *Etsi rekisterin virheitä*. Tämän jälkeen löydetty virheet tulevat isoon ikkunaan, jonka jälkeen on mahdollista painaa painiketta *Korjaa valitut rekisterin virheet*.
8. Ohjelmaa ei tarvitse itse ajaa, koska se puhdistaa koneen aina käynnistyksen yhteydessä. Ainoastaan rekisterin puhdistus pitää tehdä manuaalisesti käynnistämällä ohjelma ja seuraamalla kohtaa 7.
9. Päivitä CCleaner-ohjelmaa säännöllisin väliajoin painamalla ohjelmassa oikealla alhaalla olevaa *tarkista päivitykset* -linkkiä. Linkki johtaa ohjelman kotisivulle, joka tarkistaa päivitystarpeen ja ilmoittaa pitääkö sinun kopioida sivuilta uudempi versio asennettavaksi.

Muita yksityisyyteen ja yleisesti Windows XP:n turvallisuuteen vaikuttavia ohjelmia ovat esim. xp-AntiSpy ja Safe XP. Ohjelma löytyvät niiden valmistajien kotisivuilta. Ohjelmat on tarkoitettu hieman edistyneemmille käyttäjille, sillä niitä väärin käytettäessä Windowsin normaali toiminta saattaa vaurioitua. Ohjeita näiden ohjelmien käyttöön löytyy niiden kotisivuilta sekä ohjelmien

mukana tulevissa oppaissa. (<http://www.xp-antispy.org/> ja <http://www.theorica.net/safexp.htm>)

5. Yhteenveto

Kotikäyttäjää ja hänen tietokonettaan on vastassa Internetistä tulevat lukuisat tietoturvaan liittyvät uhkatekijät. Monet käyttäjistä eivät edes tiedä tai tiedosta niitä kaikkia. Olen tässä tutkielmassa tehnyt riskikartoituksen liittyen juuri Internetistä tuleviin tietoturvauhkiin, joita kotikäyttäjä voi kohdata sekä pyrkinyt tarjoamaan ohjeistuksen näiden uhkien minimoimiseen.

Kotikäyttäjille suunnatut tietoturvaoppaat eivät tarjoa kovin kattavaa riskikartoitusta tai ohjeistusta tietoturvauhkiin. Tämä selvisi minulle tehdessäni riskikartoitusta. Vakoilu- ja mainosohjelmat ovat yksi uhkatekijä, joka on jäänyt tavallisista tietoturvaoppaista mainitsematta, kuten myös erimuotoiset huijaukset. Osaselityksenä tähän on se, että nämä uhat ovat varsin tuoreita.

Pelkän virustorjuntaohjelman ja palomuurin asentaminen kotikoneeseen ei enää riitä. Tarvitaan muitakin ohjelmia ja suojauskeinoja, kuten vakoilu- ja mainostenpoisto-ohjelmia sekä yksityisyyden ja oman identiteetin suojaamista. Merkittävää on lisäksi se, että kotikäyttäjä on itsessään varsin suuri tietoturvariski hänen hyväuskoisuutensa ja helpon luottamuksen saannin takia.

Ongelmalliseksi kotikäyttäjän tietoturvan tekee se, että Internetistä tulee jatkuvasti erilaisia uusia uhkatekijöitä. Aivan viimeisimpiin kuuluvat rootkit-ohjelmat, joita en tähän tutkielmaan ottanut mukaan, koska ne eivät ole vielä yleisiä kotikäyttäjän Windows-käyttäjärjestelmässä. Valitettavasti ne ovat tosin yleistymässä. Tästä kaikesta seurauksena on se, että kotikäyttäjille suunnatut tietoturvaoppaat tarvitsisivat jatkuvaa päivittämistä.

Tärkeää olisi jatkossa myös analysoida sitä, miten tietoturvaoppaita pitäisi tehdä, jotta tavalliset kotikäyttäjät pääsisivät niihin helposti sisälle. Millainen hyvä ohjeistus olisi, jotta se hyödyttäisi kotikäyttäjää mahdollisimman paljon?

Viiteluettelo

[Anderson, 2004] Ron Anderson, Sick of spam. *Network Computing* **15**, 9 (May 2004), 42-43,46,48,55.

[Bequai, 2000] August Bequai, America's Internet commerce and the threat of fraud. *Computers & Security* **19**, 8 (Dec. 2000), 688-691.

[CERT/CC, 2001] CERT Coordination Center, *Home Network Security*. Software Engineering Institute, Carnegie Mellon University, http://www.cert.org/tech_tips/home_networks.html (30.11.2005).

- [CERT/CC, 2002] CERT Coordination Center, *Home Computer Security*. Software Engineering Institute, Carnegie Mellon University, <http://www.cert.org/homeusers/HomeComputerSecurity/> (30.11.2005).
- [Ford, 1999] Richard Ford, Malware: Troy revisited. *Computers & Security* **18**, 2 (1999), 105-108.
- [Hinde, 2003] Stephen Hinde, Careless about privacy. *Computers & Security* **22**, 4 (May 2003), 284-288.
- [Järvinen, 2002a] Petteri Järvinen, *Tietoturva & yksityisyys*. Docendo, 2002.
- [Järvinen, 2002b] Petteri Järvinen, Microsoftin ristiriita: turvallisuus vai käytön helppous. *Tietokone* **8** (2002).
- [Knight, 2004] Will Knight, PC spies report on your every move. *New Scientist* **182**, 2453 (Jun. 2004), 24.
- [Kotilainen, 2005a] Samuli Kotilainen, Zombie-verkot varastavat jopa roolipelitavaroita. *Tietokone* (15.3.2005), http://www.tietokone.fi/uutta/uutinen.asp?news_id=23344&tyyppi=1 (30.11.2005).
- [Kotilainen, 2005b] Samuli Kotilainen, henkilökohtainen phishing uusin nettiuhka. *Tietokone* (16.5.2005), http://www.tietokone.fi/uutta/uutinen.asp?news_id=23898&tyyppi=1 (30.11.2005).
- [Landwehr et al., 1994] Carl E. Landwehr, Alan R. Bull, John P. McDermott and William S. Choi, A taxonomy of computer program security flaws. *ACM Computing Surveys* **26**, 3 (Sep. 1994), 211-254.
- [Lehto, 2005] Tero Lehto, Kotitietokoneet netissä rikollisten kauppatavaraa. *Tietokone* (16.9.2005), http://www.tietokone.fi/uutta/uutinen.asp?news_id=24713&tyyppi=1 (30.11.2005).
- [Munro, 2004] Jay Munro, Bots march in. *PC Magazine* **23**, 22 (Dec. 2004), 90-94.
- [Qing and Wen, 2005] Sihan Qing and Weiping Wen, A survey and trends on Internet worms. *Computer & Security* **24**, 4 (Jun. 2005), 334-346.
- [Reiss, 2003] Markku Reiss, Laajakaistan käyttäjät suojautuvat surkeasti. *Tietokone* (9.6.2003), http://www.tietokone.fi/uutta/uutinen.asp?news_id=19504&tyyppi=1 (30.11.2005).
- [Saarijärvi, 2004] Marko Saarijärvi, Tietoturva- ja verkkopalvelujen käyttö kodeissa – kyselytutkimus laajakaistayhteyden käyttäjälle. Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, Pro gradu -tutkielma, Toukokuu 2004.
- [VM, 2003] Valtiovarainministeriö, *Valtionhallinnon tietoturvakäsitteistö 4/2003*. Valtionhallinnon tietoturvallisuuden johtoryhmä, <http://www.vm.fi/vm/liston/page.lsp?r=50903&1=fi&menu=3246> (30.11.2005).

[VM, 2004] Valtiovarainministeriö, *Haittaohjelmilta suojautumisen yleisohje 3/2004*. Valtionhallinnon tietoturvallisuuden johtoryhmä, <http://www.vm.fi/vm/liston/page.lsp?r=88080&l=fi&menu=3246> (30.11.2005).

Selainsovelman ja palvelimen välinen viestintä

Bemmu Sepponen

Tiivistelmä.

Tutkimuksessa esitellään kolme alustaa WWW-selaimessa ajettavien sovelmien rakentamiseen. Erityisesti keskitytään niiden mahdollisuuksiin siirtää tietoa reaaliaikaisesti verkkoyhteyden yli. Tutkimuksessa tarkastellaan millaisia mahdollisuuksia Java-sovelmat, Macromedia Flash ja Ajax tähän tarjoavat.

Avainsanat ja -sanonnat: Ajax, Java, Macromedia Flash, Web 2.0, WWW-selain

CR-luokat: D.1

1. Johdanto

WWW-sivustojen käyttö mielletään usein hitaaksi puuhaksi, sillä liikkueessaan sivustolla jokainen käyttäjän valitsema linkki tai lähetetty WWW-lomake avaa uuden yhteyden palvelimelle HTTP-kutsua varten. Yksittäisen kutsun jälkeen yhteys katkaistaan. Kun kyse on dynaamisesta sisällöstä, ei selaimen välimuististakaan ole apua. Tom Muck [2003] vertaa tilannetta puhelinkeskusteluun, jossa soittaisit ystävällesi ja esittäisit kysymyksen, sulkisit luurin ja kohta ystäväsi soittaisi taas sinulle ja kertoisi vastauksen. Soittaisit hänelle uudestaan, sulkisit luurin ja tähän tapaan keskustelu matelisi pätkittäin hitaasti eteenpäin.

Malli, jossa ainoastaan palvelin voi muuttaa näkyvää sisältöä ja selaimen tehtäväksi jää ainoastaan staattisen HTML-sivun esittäminen ruudulla (jatkossa "perinteinen malli"), on rajoittava etenkin silloin, kun haluttaisiin rakentaa työpöytäsovelluksien vuorovaikutustasoa muistuttavia WWW-sovelluksia. Perinteisessä mallissa viestintäyhteyksiä ei voida avata kuin ainoastaan siirryttäessä HTML-sivulta toiselle, eikä sivun ollessa näkyvillä voida samanaikaisesti siirtää tietoa taustalla. Toistuvat elementit, kuten navigointivalikot, joudutaan lähettämään palvelimelta yhä uudestaan ja uudestaan. Uudelle sivulle siirtyminen saattaa myös aiheuttaa selaimen suorittamasta HTML-koodin tulkinnasta ja sivun piirtämiseen kuluva ajasta johtuvia viiveitä tai visuaalisesti epämiellyttäviä hetkiä, joina sivu on vasta osittain latautunut.

Kun otetaan käyttöön tekniikoita, joilla on mahdollista ohjelmoida palvelinsovellusten lisäksi myös selainsovelmia, aukenevat WWW-kehittäjälle uudet mahdollisuudet lyhentää vuorovaikutuksen vasteaikoja. Suosituimmat nykyi-

set selaimet, [HitsLink] eli Microsoft Internet Explorer, Firefox ja Safari, sekä kyseisten selainten liitännäiset (plug-in) tarjoavat tähän monipuoliset mahdollisuudet.

Luvuissa 2-4 esitellään Javan, Macromedia Flashin ja JavaScriptin tarjoamat mahdollisuudet. Lopuksi luku 5 sisältää yhteenvedon käsitellyistä asioista.

2. Java

Java kehitettiin alunperin sulautettujen järjestelmien tarpeisiin, mutta mukautettiin soveltumaan myös Internet-käyttöön [Gosling *et al.*, 2000]. Sanalla "Java" viitataan tässä Java-virtuaalikoneeseen ja Java-kieleen yhdessä. Ne muodostavatkin kokonaisuuden, sillä Java-virtuaalikone suorittaa Java-kielestä käännettyä Java-tavukoodia. Muitakin Java-tavukoodille käännettäviä kieliä on kuitenkin olemassa, itse asiassa erilaisia Java-tavukoodia tuottavia kääntäjiä on olemassa lähes kaksi sataa [Languages for the Java VM]. WWW-selaimessa toimivien sovelmien laatimisen lisäksi Java sopii myös palvelinpuolen ohjelmointiin. Esimerkiksi Tampereen yliopiston Informaatiotutkimuksen laitoksen tiedonhaun opetus- ja tutkimusohjelma QPA (Query Performance Analyser) on laadittu Java-palvelinsovelmana (servlet) [Sormunen and Pennanen, 2004, figure 1.].

Virtuaalikoneetta voidaan ajaa WWW-selaimessa, mikäli selaimeen on asennettu vaadittu liitännäinen. Virtuaalikoneessa ajettavan tavukoodin käyttö selainsovelmien toteuttamiseen on huomattavasti turvallisempaa kuin ratkaisu, jossa sovelman annettaisiin käyttää tietokoneen prosessoria suoraan, sillä virtuaalikoneen oikeuksia esimerkiksi tiedostojen kirjoittamisen osalta voidaan rajoittaa. Virtuaalikoneella on myös oma muistialueensa, jonka ulkopuolelle se ei pääse käsiksi. Tällaiset rajoitteet ovat ensiarvoisen tärkeitä, jotta kolmannen osapuolen laatimia sovelmia voidaan ajaa ilman, että ne saisivat tietoonsa käyttäjän henkilökohtaisia tietoja, tai voisivat vahingoittaa käyttäjän järjestelmää vahingossa tai tahallisesti.

Java-sovelmia voidaan käyttää korvaamaan perinteinen malli, kun HTML-elementein luotu käyttöliittymä korvataan Java-sovelmalla. Teksti, kuvat ja lomakkeet voidaan näin esittää Java-sovelman sisällä. Sen sijaan että käyttäjä navigoisi WWW-sovelluksessa HTML-sivulta toiselle, navigointi tapahtuukin Java-sovelman sisällä, eikä navigointi aiheuta samanlaisia katkoksia kuin perinteisessä mallissa.

Java-sovelma voi viestiä palvelimen kanssa TCP-protokollalla suoraan Javan vastakkeiden (socket) avulla, tai niiden päälle rakennettujen muiden palvelujen, kuten JDBC:n (Java database connectivity) tai RMI:n (remote method invocation) kautta. TCP:n lisäksi myös UDP-pakettien käyttö Java-sovelmasta on mahdollista. Muiden IP-protokollien kuten ICMP:n käyttö ei kuitenkaan

Java-sovelmasta onnistu [Harold, 2004, section 13.1.].

TCP ja UDP ovat useimmiten kuitenkin täysin riittäviä sovellusten tarpeisiin, ja siispä Java-sovelmat tarjoavat ohjelmistokehittäjälle lähes vastaavat mahdollisuudet kuin työpöytäsovelluksenkin kehittäminen tarjoaisi, lukuunottamatta Java-sovelmille asetettua rajoitusta, joka estää niitä ottamasta yhteyksiä muille palvelimille kuin sille, jolta sovelma alunperin haettiin. Rajoitusta voidaan kiertää luomalla välityspalvelin, joka välittää Java-sovelman pyynnön eteenpäin [Chase, 2004].

2.1. Javan vastakkeet (sockets)

Yhteydenotto selaimen Java-sovelmasta palvelimeen tapahtuu luomalla uusi vastake Socket-luokan muodostinta (constructor) käyttäen. Kaikki Javan tarjoamat TCP-yhteyksiä käyttävät palvelut hyödyntävät tosiasiaassa alimmalla tasolla Socket-luokkaa. Yhteyden muodostuttua voidaan luodulta Socket-luokan oliolta pyytää syöte- ja tulostevirtoja (stream) edustavat oliot, joiden kautta varsinainen viestintä sitten tapahtuu. Listauksessa 1. on lyhyt esimerkki, jossa avataan TCP-yhteys Tampereen yliopiston WWW-palvelimelle, ja luetaan sieltä pääsivun sisältö HTTP-protokollaa käyttäen.

```
Socket uta = new Socket("www.uta.fi", 80);

OutputStream raw = uta.getOutputStream();
OutputStream buffered = new BufferedOutputStream(raw);
out = new OutputStreamWriter(buffered, "ASCII");

out.write("GET / HTTP/1.0\r\n\r\n");
out.flush();

BufferedReader in = new BufferedReader(
    new InputStreamReader(uta.getInputStream())
);
String line = null;
while ((line = in.readLine()) != null) {
    System.out.println(line);
}
```

Listaus 1. Yhteydenotto Java-vastakkeella

Esimerkissä avataan palvelimen HTTP-portti, lähetetään sinne pääsivun pyyntö, luetaan tulostusvirrasta palvelimen antama vastaus ja tulostetaan se rivi kerrallaan Javan konsolille. Socket-luokkaa vastaava luokka UDP-yhteyksiä varten on DatagramSocket.

2.2. JDBC ja RMI

Pääsy relaatiotietokantaan on yksi tavanomaisimpia syitä, miksi selainsovelma ylipäättänsäkään tarvitsee verkkoyhteyttä. JDBC on ohjelmointirajapinta (API), joka määrittelee, miten Java-ohjelma voi olla yhteydessä tietokantaan [Wikipedia: JDBC]. Se mahdollistaa kyselyt relaatiotietokantoihin verkon ylitsekin.

RMI on ohjelmointirajapinta ja Javan luokkakirjasto, jolla voidaan kutsua palvelimella sijaitsevia ohjelmametodeja samaan tapaan kuin jos metodit olisivat osa paikallista sovellusta. Tietenkään palvelimella sijaitsevien metodien kutsuminen ei voi olla yhtä nopeata kuin paikallisten metodien kutsuminen; lisäksi on aina varauduttava poikkeuksiin (exception), joita voi syntyä mikäli kutsu epäonnistuu verkkoyhteysongelmien vuoksi. [Harold, 2004, section 18.1.]

Sekä JDBC- että RMI-kutsuja on mahdollista tehdä Java-sovellusten lisäksi myös selaimessa toimivista Java-sovelmista. Koska tekniikat on rakennettu vastakkeiden päälle, koskee niitäkin rajoitus, joka estää ottamasta yhteyttä muihin kuin siihen palvelimeen, jolta itse Java-sovelma on ladattu.

2.3. Java-sovelmien edut ja haitat

Javalla toteutetaan suuriakin ohjelmistokokonaisuuksia, ja virtuaalikoneensa ansiosta myös turvallisesti selaimessa ajettavien sovelmien ohjelmointi on mahdollista. Java on yleisesti hyväksytty alusta, jolle kehittäjiä ja ohjeistusta on helppo löytää. Voikin vaikuttaa siltä, että kaikki selainohjelmointi tulisi tehdä Javaa käyttäen ja muut vaihtoehdot voisi suosiolla unohtaa. Java-sovelmilla on kuitenkin myös heikkoutensa.

Sovelmien integroitavuus WWW-ympäristöön on rajallinen, sillä sovelma varaa ruudulta tietyn suorakulmaisen tilan joka eroaa muusta WWW-sivustosta. Java-sovelma ei siis voi yksinään vaikuttaa WWW-sivujen muuhun sisältöön kuin omaan alueeseensa. Käyttäjälle ei ole kovinkaan vaikeata havaita, missä tavanomainen HTML-sivu loppuu ja Java-sovelma alkaa. Usein siirtymästä tiedottaa ikävä harmaa laatikko, joka käyttäjälle on sovelman tilalta näkyvillä Java-virtuaalikoneen rauhallisesti käynnistyessä. Java-sovelmat ovat usein visuaalisesti ja käyttäytymiseltään yhteensopimattomia muun sivuston kanssa, sillä Java toteuttaa omat käyttöliittymäelementtinsä jotka näyttävät ja tuntuvat erilaiselta kuin mihin WWW-sivujen käyttäjä on ehkä muuten tottunut.

Joidenkin selainten Java-virtuaalikoneen toteutuksissa sovelmat käynnistyivät hitaasti ja käyttäytyivät tahmeasti, välillä myös ruudunpäivityksessä tapahtui virheitä. Näistä syistä käyttäjät saattavat yhä nähdä Java-sovelmat

yleisestikin huonossa valossa. Virtuaalikoneiden toteutuksien parantuessa, sekä tietokoneiden ja verkkoyhteyksien nopeutuessa voidaan odottaa, että vastaavanlaiset hidastukset ja virhetilanteet vähenevät.

3. Macromedia Flash

Vuonna 1993 Jonathan Gay perusti yrityksen nimeltä FutureWave Software, jonka tarkoituksena oli kehittää grafiikkaohjelmia kynätietokoneille. Kuitenkin kynätietokoneiden tulevaisuuden näyttäessä yhä epävarmemmalta, päätettiin kehiteillä oleva piirto-ohjelma sittenkin siirtää Windowsille ja Macintoshille. Käyttäjäläpälautteessa FutureWavea kannustettiin pelkän piirto-ohjelman sijaan laajentamaan ohjelmaansa myös animointiin. Näinä aikoina Internetin suosio oli vahvassa nousussa, ja FutureWavella nähtiin että Internet-selaimessa toimivien esitysten luontiin kykenevä animointiohjelma saattaisi olla menestys. [Flashmagazine]

Alussa Java-sovelmana toteutetulle soittimelle löytyikin kysyntää, ja sitä käyttivät muun muassa Microsoft ja Disney. Myöhemmin soittimesta rakennettiin oma liitännäinen, joka käyttäjien tuli erikseen asentaa selaimen, mutta joka pystyi tarjoamaan Java-sovelmaa sulavampaa animaatiota. Vuoden 1995 joulukuussa Macromedia osti FutureWaven, ja animointiohjelma nimettiin Macromedia Flashiksi [Flashmagazine]. Macromedian teettämän tutkimuksen mukaan [Macromedia] syyskuussa 2005 oli 98.7%:lle eurooppalaisista Internet-yhteydellä varustetuista tietokoneista asennettu vähintään Flash-soittimen versio 5.

Vaikka Flash ei ohjelmointiympäristönä olekaan kielellisesti yhtä monipuolisella tasolla kuin Java, on sen ohjelmoitavuus kuitenkin kehittynyt riittävän pitkälle jotta sillä voidaan toteuttaa suurempiakin ohjelmia. Kun vuonna 1998 Flashin versiossa 3 oli yksinkertaisen ristinollapelin toteuttaminen vielä vaikeaa [Moock, 2001, foreword], on Flashia käyttäen nykyään täysin mahdollista toteuttaa esimerkiksi verkkokauppapaikkoja tai jopa videoneuvottelujärjestelmiä. Flashin skriptikieli ActionScript perustuu samaan ECMA-262 -standardiin kuin JavaScriptkin, ja sitä on mahdollista käyttää oliopohjaiseen ohjelmistokehitykseen.

Tässä luvussa esitellään kolme tapaa hakea palvelimelta tietoa Flash-sovelman käyttöön: loadVariables-funktio, XMLSocket-luokka ja Flash Communication Server. Erityistapauksina käsittelemättä jätetään SWF-, JPEG- ja MP3-tiedostojen lataaminen verkon yli. XML-tiedoston hakemista XML-luokan load-metodilla ei myöskään käsitellä, sillä sama toiminnallisuus sisältyy myös XMLSocket-luokkaan. UDP-pakettien tai muiden IP-protokollien käyttö Flashista ei ole mahdollista.

3.1. Flashin loadVariables-funktio

Yksinkertaisin tapa siirtää tietoa palvelimelta Flash-sovelmaan on Flashin loadVariables-funktio. Palvelin tarjoaa tällöin tiedot HTTP-protokollaa käyttäen sovelmalle, eli sovelma hakee tiedot palvelimella sijaitsevasta tiedostosta tai tiedot dynaamisesti luovalta tai muualta välittävältä palvelinohjelmalta. [Jacobson and Jacobson, 2001, section 7.3.]

Tietojen täytyy olla MIME-tyyppiä application/x-www-form-urlencoded, eli tiedot välitetään samalla tavalla kuin selaimen osoitekentässäkin. Siis jos halutaan välittää palvelimelta vaikkapa muuttujat kosteus = 91% ja paine = 980,4hPa, tulisi tiedot esittää muodossa "kosteus=91%25&paine=980%2C4hPA". Mikäli tiedot tuotetaan dynaamisesti palvelinohjelmasta, on yleensä olemassa jokin funktio jolla muunnos voidaan helposti tehdä, esimerkiksi PHP-skriptikielessä muunnos tapahtuu urlencode-funktiolla [urlencode].

Tietojen hakeminen onnistuu ActionScriptin loadVariables-funktiolla, ja on äärimmäisen yksinkertaista. Funktion toista argumenttia voidaan käyttää halutun nimiavaruuden valintaan. Jos mitään nimiavaruutta ei erikseen valita, olettaa funktio halutun nimiavaruuden olevan this-nimiavaruus, jolloin funktion kutsumiseksi riittää palvelimella sijaitsevan tiedoston tai tiedot tuottavan skriptin nimen antaminen ainoana argumenttina, eli esimerkiksi

```
loadVariables("kelitiedot.txt").
```

Funktio ei palauta mitään arvoa, vaan luetut tiedot asetetaan suoraan valitun nimiavaruuden muuttujien arvoiksi [Jacobson and Jacobson, 2001, section 7.3.]. Esimerkkikutsun jälkeen ActionScript-muuttujan "kosteus" arvoksi tulisi merkkijono "91%", ja muuttujan "paine" arvoksi merkkijono "980,4hPa". Funktio toimii asynkronisesti, eli funktion palatessa ei muuttujien arvoja ole vielä asetettu, vaan tietojen haku on vasta käynnistynyt. Tietojen todellinen latautuminen laukaisee tapahtuman (event) "data", johon voidaan liittää oma ohjelmarutiini, joka voi sitten hyödyntää haettuja tietoja [Jacobson and Jacobson, 2001, section 7.5.].

Flashissa on turvallisuuden vuoksi yleinen rajoitus, joka estää tietojen haun muilta verkkoalueilta (domain) kuin siltä jolta myös itse Flash-sovelma haettiin. Rajoitusta on kuitenkin mahdollista kiertää laatimalla palvelinohjelma, joka voi toimia muilta verkkoalueilta haettujen tietojen välittäjänä Flash-sovelmalle.

Käyttämällä loadVariables-funktion [loadVariables] kolmatta argumenttia on funktiolla mahdollista myöskin lähettää tietoa GET- ja POST-välitystavoin. Tässä tapauksessa ensimmäinen argumentti määrittelee mille palvelinskriptille tiedot lähetetään, toinen argumentti minkä nimiavaruuden tiedot lähetetään ja kolmantena argumenttina on välitystapa joka voi olla joko "GET" tai "POST".

Muihin ohjelmointiympäristöihin tottunutta ohjelmoijaa voi funktion käyttäytyminen yllättää, sillä se lähettää määritellyn nimiavaruuden kaikkien muuttujien sisällön palvelimelle. Jos siis käytetään this-nimiavaruutta, saattaa mukaan eksyä tahattomasti sellaisiakin tietoja joiden lähettäminen ei välttämättä olisi tarpeellista [Plant and Cleveland, 2001, section 22.3.].

3.2. Vastakkeet ActionScriptissä

Kokeneempi kehittäjä on varmasti kiinnostunut millaiset mahdollisuudet Flash tarjoaa vastakkeiden käyttöön, sillä niiden avulla on reaaliaikainen kaksisuuntainen tiedonsiirto mahdollista. Kun käytetään loadVariables-funktiota, on ainoa keino palvelimella olevien tietojen muuttumisen havaitsemiseen niiden toistuva kysyminen palvelimelta. Sen sijaan vastakkeiden avulla voidaan yhteys pitää avoinna, ja palvelin voi kertoa hyvin pienellä viiveellä tietojen muuttumisesta.

Vaikka vastakkeiden käyttö on ActionScriptissä mahdollista, liittyy niiden käyttöön merkittäviä rajoituksia. Jo luvun alussa mainitun verkkoaluerajoitteen lisäksi ei selaimessa toimiva Flash-sovelma voi ottaa yhteyksiä portteihin 0-1023. Rajoite saattaa parantaa Flashin tietoturvaa jossain määrin, mutta estää esimerkiksi telnet-asiakasohjelman toteuttamisen ActionScriptillä [Jacobson and Jacobson, 2001, section 19.2]. Lisärajoituksena vastakeluokka XMLSocket [XMLSocket] on suunniteltu nimenomaan XML-muotoisen tiedon välitykseen, joten lähetettävä tieto tulisi aina paketoita XML-muotoon. Suuri rajoitus on myös se, että palvelinvastakkeita ei ActionScriptissä ole mahdollista luoda. Palvelinta ei siis Flash-sovelmasta voida tehdä, mutta toisaalta on vaikeaa kuvitella tilannetta jossa se olisi tarpeellistakaan.

Hyödyllisimpiä XMLSocket-luokan toimintoja ovat yhteyden avaaminen, XML-muotoisen tiedon lähettäminen vastaketta käyttäen ja yhteyden sulkeminen. Luokkaa voidaan pyytää kutsumaan käyttäjän antamaa funktiota kun yhteys on muodostettu, vastapuoli on sulkenut yhteyden tai kun uutta tietoa on vastaanotettu. Funktioiden avulla on tapahtumapohjainen ohjelmointi mahdollista, ja saapuneeseen tietoon voidaan reagoida välittömästi, ilman että luokan tilaa olisi tarpeen seurata silmukassa.

XMLSocket-luokan käyttö ei sinänsä ole hankalaa, mutta haastavimmaksi osaksi saattaakin osoittautua useiden asiakassovelmien sujuvaan yhtäaikaiseen palvelemiseen kykenevän palvelinohjelman kirjoittaminen. Valmiitakin palvelinohjelmia on saatavilla, yksi tunnetuista on monen käyttäjän Flash-sovelmien kirjoittamiseen soveltuva Unity [Unity]. XMLSocket-luokan käyttö Flashissa on erityisen hyvä vaihtoehto silloin, kun halutaan vasteajaltaan nopeita yhteyksiä selaimesta esimerkiksi useamman pelaajan pelejä varten.

3.3. Flash Communication Server

Flash-soittimissa on nykyisin tuki mikrofoneille ja webbikameroille, jota hyödyntäen on rakennettu esimerkiksi erityisesti yrityksille suunnattu videoneuvottelujärjestelmä Breeze [Breeze]. Monen käyttäjän ympäristössä ääntä ja liikkuvaa kuvaa välittävän palvelinohjelman laatiminen saattaisi olla monille kehittäjille liian suuri kynnys, joten Macromedia tarjoaa avuksi valmista palvelinohjelmaa nimeltä Flash Communication Server (jatkossa "FlashCom"). Ohjelma tarjoaa valmiin pohjan, jonka päälle voidaan rakentaa omia palvelinsovelluksia käyttäen Server-Side ActionScript -kieltä (SSAS) [Giacomo Guilizzoni *et al.*, 2005].

```
out_ns = new NetStream(nc);  
out_ns.attachAudio(Microphone.get( ));  
out_ns.attachVideo(Camera.get( ));  
out_ns.publish(userName); // Name the stream after the user's name.
```

Listaus 2. Äänen ja videokuvan lähetys Flashista

FlashCom ja uudet Flash-soittimen versiot tarjoavat kehittäjän kannalta helpon alustan tehdä videota ja ääntä hyödyntäviä sovelluksia. Tällaisten sovellusten luominen on alustalle suorastaan hätkähdyttävän suoraviivaista, kuten listauksen 2 esimerkki [Giacomo Guilizzoni *et al.*, 2005, section 1.4.] osoittaa. Esimerkissä käynnistetään äänen ja videon lähetys mikrofonilta ja kameralta FlashCom-palvelimelle.

FlashCom skaalautuu aina tuhansille käyttäjille esitettävän videosyötteen suoratoistoon asti, sillä kuormaa on mahdollista jakaa useille toisiinsa yhteydessä oleville palvelimille [Giacomo Guilizzoni *et al.*, 2005, section 1.1]. Kun tiedonsiirtotarve palvelimen ja selaimen välillä koostuu tekstin lisäksi myös kaksisuuntaisesta äänestä ja videosta, lienee Flash-soittimen ja FlashComin tarjoama alusta tällä hetkellä ainoa varteenotettava vaihtoehto.

3.4. Flash-sovelmien edut ja haitat

Tämän luvun aikaisemmissa kohdissa selvitettiin Flashin tarjoamat mahdollisuudet selaimen ja palvelimen väliseen yhteydenpitoon. Flash on tähän suosittu ja toimivaksi havaittu ratkaisu, mutta sillä on myös haittapuolensa. Kuten kohdassa 2.3. mainittiin Java-sovelmista, on myös Flash-sovelmien visuaalinen ilme ja käyttäytyminen tavallisista HTML-sivuista poikkeavaa. Jos selaimessa

halutaan toteuttaa pelejä, ovat Flashin grafiikkamahdollisuudet toki ensiarvoisen tärkeitä, mutta toteutettaessa esimerkiksi verkkokauppaa lienee tärkeämpää että käyttötuntuma ei ole käyttäjälle liian vieras. "Mitä paremmin järjestelmä vastaa käyttäjän odotuksia, sitä tyytyväisempi hän on" [Nielsen, 1999].

4. Ajax

Termin "Ajax" (asynchronous Javascript and XML) esitti ensimmäisen kerran Jesse Garrett [2005] artikkelissaan "Ajax: A New Approach to Web Applications". Artikkelin ilmestymisen jälkeen on käsite levinnyt nopeasti WWW-kehittäjien tietoon, erityisesti Googlen käytettyä Ajax-tekniikoita Gmail- ja Google Maps -sovelluksissaan. Ajax ei varsinaisesti ole mikään yksittäinen tekniikka, vaan se viittaa WWW-sovellusten kehittämiseen siten, että Javascript-kielellä toteutettu sovelma lähettää XML-muotoista tietoa XMLHttpRequest-luokkaa käyttäen. Vaikka kyseistä luokkaa on ollut mahdollista käyttää jo Microsoft Internet Explorer -selaimen vuoden 1998 versioista lähtien, ovat sen mahdollisuudet vasta nyt levinneet yleiseen tietoisuuteen [Hibbs, 2005].

XMLHttpRequest-luokan avulla voidaan toteuttaa kahdenlaisia yhteyksiä. Sitä voidaan käyttää väliaikaisiin yhteyksiin tai vaihtoehtoisesti yhteys voidaan jättää auki siten, että palvelin voi lähettää uusia tietoja virtana. Ikävä kyllä kaksisuuntaisen jatkuvan yhteyden mahdollisuutta XMLHttpRequest ei tarjoa. Mikäli Javascript-selainsovelmasta on tarve lähettää paljon muuttuvaa tietoa palvelimelle, on ainoa mahdollinen ratkaisu useiden lyhytaikaisten yhteyksien ottaminen. Kuten muitakin esiteltyjä tekniikoita, myös XMLHttpRequest-luokan käyttöä koskee rajoitus, joka estää yhteydet muille kuin sille palvelimelle josta selainsovellus haettiin.

4.1. Lyhytaikaiset yhteydet XMLHttpRequest-luokalla

Yhteyden muodostaminen aloitetaan luomalla instanssi XMLHttpRequest-luokasta [XULplanet: XMLHttpRequest]. Ikävä kyllä instanssin luominen vaihtelee selaimen mukaan, Mozillassa ja Safarissa se tapahtuu new-operaattorilla, mutta Microsoft Internet Explorerissa ActiveXObject-luokan välityksellä. Kun instanssi on luotu, asetetaan seuraavaksi onreadystatechange-tapahtumafunktio, jonka avulla voidaan valvoa milloin XMLHttpRequest-objektin tila muuttuu. Seuraavaksi kutsutaan objektin open- ja send-metodeja, ja tämän jälkeen yhteydenotto käynnistyy.

Open-metodin argumenteilla voidaan valita, halutaanko käyttää POST- vai GET-metodia tietojen välittämiseen, mille palvelimelle tiedot lähetetään ja kolmantena argumenttina voidaan valita halutaanko että yhteys otetaan asynkronisesti vai synkronisesti. Käytännössä käytetään aina asynkronista kutsutapaa, sillä synkroninen kutsutapa aiheuttaisi hetkellisiä taukoja Ajax-sovelmaan, ja juuri niitä näillä tekniikoilla yritetään mahdollisimman pitkälle välttää. Listauksessa 3 ovat äsken kuvaillut askeleet Javascript-koodina Mozilla- tai Safari-selainta varten.

```
x = new XMLHttpRequest();  
x.onreadystatechange = stateChanged;  
var url = "palvelinohjelma.php";  
x.open("POST", url, true);  
x.send(null);
```

Listaus 3. Asynkroninen XMLHttpRequest-kutsu

Kun palvelin on vastannut pyyntöön, tulee asetettu onreadystatechange-tapahtumafunktiota kutsutuksi; XMLHttpRequest-objektin readyState-attribuutti on tällöin asetettu arvoksi 4. Tapahtumafunktiota kutsutaan myös muissa tapauksissa, joten readyState-attribuutin arvo on tarkistettava. Saapuneet tiedot ovat saatavilla XMLHttpRequest-objektin attribuuteissa responseText ja responseXML (sama DOM-dokumenttiobjektina) [XULplanet: XMLHttpRequest]. Palvelinohjelmalta ei vaadita tietojen lähettämiseen muuta kuin normaali HTTP-vastaus joka sisältää halutut tiedot. Binäärimuotoinen tieto ei välity oikein, ellei sitä muuteta siirron ajaksi merkkijonomuotoiseksi esimerkiksi base64-koodaamalla se.

Esimerkkilistauksessa ainoastaan pyydettiin tietoja. Mikäli selaimesta haluttaisiin lähettää myös palvelimen suuntaan tietoja, ei send-metodin argumentin tulisi olla null, vaan merkkijonojoka sisältää lähetettävät tiedot. Merkkijono voi olla XML-muotoinen tai vapaata tekstiä. Jos valittu lähetystapa oli GET, tulevat send-metodin argumentiksi annettavat tiedot HTTP-kutsun alun GET-kenttään. POST-lähetystavalla tiedot tulevat HTTP-kutsun lopun sisältöosuuteen, eli aivan samaan tapaan kuin HTTP-kutsuissa yleensäkin.

4.2. Jatkuva yhteys XMLHttpRequest-luokalla

Vähemmän tunnettu XMLHttpRequest-luokan tarjoama mahdollisuus on jatkuvan yksisuuntaisen yhteyden ylläpitäminen palvelimeen. Tiedot välittyvät siten, että palvelin lähettää selaimelle peräkkäisiä XML-muotoisia tietopaketteja, ja jokainen vastaanotettu paketti aiheuttaa uuden kutsun XMLHttpRequest-oliolle asetettuun onload-tapahtumafunktioon. Tällaista yhteyttä käyttäen kykenee palvelin periaatteessa lähettämään selaimelle jopa reaaliaikaista videokuvaa, tosin tällöin videovirran yksittäiset kuvat on lähetettävä XML-muotoisina base64-koodattuina JPEG- tai PNG-kuvina, eikä ratkaisua voida sen vaatiman lähetyskaistan vuoksi pitää käytännöllisenä.

Kun halutaan vastaanottaa jatkuvaa tietovirtaa, tapahtuu yhteydenotto selaimesta palvelimelle hyvin samaan tapaan kuin edellisessäkin kohdassa, mutta tällä kertaa asetetaan XMLHttpRequest-luokan multipart-attribuutti todeksi, ja määritellään myös onload-tapahtumafunktio jolla tietojen saapumista voidaan seurata. Onload-tapahtumafunktio tulee kutsutuksi aina uuden tietopaketin saapuessa, ja paketin sisältö löytyy tuttuun tapaan XMLHttpRequest-olion responseText- tai responseXML-attribuuteista.

Palvelinohjelman puolella tilanne on myös muuttunut, sillä nyt XML-muotoisia tietopaketteja lähetetään useampia. HTTP-vastauksen MIME-tyypiksi on asetettava multipart/x-mixed-replace;boundary="erotin", jossa erotin on vapaasti valittava erotinmerkkijono, jonka ainoa vaatimus on ettei sitä esiinny varsinaisessa sisällössä. XML-paketit voidaan sitten lähettää, erotellen ne toisistaan kahdella viivalla ja valitulla erotinmerkkijonolla. Erottimien tulee olla omilla riveillään, eli esimerkiksi "\n—erotin\n". Myös ennen ensimmäistä XML-tietopakettia lisätään erotin. Jos yhteys halutaan katkaista, tulostetaan viimeisen erottimen jälkeen kaksi ylimääräistä viivaa. Multipart-vastausta on havainnollistettu listauksessa 4. [XULplanet: Server Push]

```
Content-type: multipart/x-mixed-replace;boundary="erotin"
--erotin

Content-type: application/xml
<?xml version='1.0'?>
<tietoja>Halutut tiedot</tietoja>
--erotin

Content-type: application/xml
<tietoja>Peräkkäisiä XML-osia voi tulla miten monta tahansa</tietoja>
--erotin--
```

Listaus 4. Jatkuva HTTP-vastaus

4.3. Ajaxin edut ja haitat

Muista esitellyistä tekniikoista poiketen Ajax sulautuu erityisen hyvin muun WWW-sivuston sisällön joukkoon. Käyttäjälle saattaa olla jopa täysin näkymätöntä että mitään tavallisuudesta poikkeavaa tekniikkaa käytetään; ainoa vihje että WWW-sovellus on toteutettu Ajaxilla ovat lyhentyneet vasteajat tai muuttuva sisältö jo latautuneen sivun sisällä. Nämä ovat merkittäviä etuja, joiden ansiosta WWW-sovellukset alkavat muistuttaa yhä enemmän tavanomaisia työpöytäsovelluksia [Garrett, 2005].

Koska Ajax-tekniikkaa hyödyntäen on mahdollista ottaa selaimesta palvelimeen yhteyksiä käyttäjän sitä havaitsematta, tarjoaa tekniikka myös täysin uudenlaiset mahdollisuudet käyttäjien valvontaan. Ylläpitäjien on ollut tähänkin asti mahdollista tietää millä sivulla käyttäjä liikkuu, mutta nyt voidaan jopa käyttäjän hiiren liikkeitä valvoa jos niin halutaan. Tekniikalla ei ole kuitenkaan mahdollista valvoa mitä käyttäjä tekee muilla sivuilla, sillä Javascript-ohjelmat voivat tarkastella ainoastaan samalta palvelimelta ladattujen dokumenttien sisältöjä [Wikipedia: Same origin policy]. Sama rajoitus koskee myös tietojen lähettämistä XMLHttpRequest-luokalla, eli yhteyksiä voi ottaa vain omaan palvelimeensa.

Ajax-tekniikkaa on usein moitittu siitä, että yleensä edelliselle sivulle vievä selaimen back-nappi lakkaa toimimasta odotetulla tavalla. Tietenkin vastaava ongelma koskee myös Java- ja Flash-sovelmia, mutta niissä käyttäjän on helpompi erotella sovelman sisäinen navigointi ja WWW-selaimen navigointi toisistaan, sillä sovelmien ulkoasu poikkeaa tavanomaisista HTML-sivuista. Vaikka back-nappi helposti rikkoutuukin toteutettaessa WWW-sovellusta Ajax-tekniikalla, on se mahdollista palauttaa jälleen toimintakuntoiseksi käyttämällä piilotettuja IFrame-alueita, joilla voidaan luoda back- napille kohteita, joiden avulla Ajax-sovelma voi huomata back-nappia painettaneen, ilman että selain poistuu sovelmasta täysin.

Kritiikkiä on kohdistunut myös siihen, että lisättäessä sivua selaimen kirjanmerkkeihin (bookmark/favorites) eivät Ajaxilla toteuttujen WWW-sovelmien tilat tallennu kuten käyttäjä saattaisi odottaa. Tähänkin ongelmakohtaan on kuitenkin löydetty ratkaisu, tässä tapauksessa asettamalla Javascriptiä käyttäen tämänhetkistä Ajax-sovelman tilaa osoittamaan kirjanmerkkeihinkin tallentuva ankkurilinkki. [Wikipedia: Ajax]

5. Yhteenveto

Esiteltyjä tekniikoita käyttäen WWW-sovelluksille tyypillisiä viiveitä on mahdollista merkittävästi lyhentää, mahdollistaen työpöytäsovelluksia

muistuttavien WWW-sovellusten kehittämisen. Tulevaisuudessa yhä useampia sovelluksia käytettäneenkin WWW-selaimesta, sillä WWW-sovellukset tarjoavat useita etuja työpöytäsovelluksiin verrattuna. Sovellusta voidaan käyttää miltä tahansa WWW-selaimella varustetulta tietokoneelta paikasta riippumatta, ja ohjelmat ovat jatkuvasti ajan tasalla ilman ohjelmapäivitysten asentamisen tarvetta.

Ohjelmistojen siirtyminen verkkoon tekee niistä enemmänkin palveluja kuin muuttumattomia tuotteita. Tämä muuttaa perinteistä ohjelmistojen kehitystapaa, sillä ohjelmiin voidaan julkaista uusia ominaisuuksia sitä mukaa kun ominaisuudet valmistuvat, sen sijaan että ohjelmalla olisi tiettyjä vain harvoin julkaistavia uusia versioita. Onkin mielenkiintoista nähdä, miten vakiintuneet ohjelmistoyritykset tulevat selviämään tästä muutoksesta. Myös ohjelmoijille muutos aiheuttaa haasteita, sillä nopeutuvassa kehityksessä mukana pysyminen vaatii entistä enemmän jatkuvaa uuden oppimista.

Tässä tutkimuksessa käsiteltiin ainoastaan joitakin tämän hetken suosituimmista tekniikoista, täysin käsittelemättä jätettiin esimerkiksi Macromedia Shockwave. Muidenkin tekniikoiden mahdollisuuksien kartoittamisesta olisikin aiheita jatkotutkimukselle.

Viiteluettelo

- [Breeze] Macromedia Breeze. <http://www.macromedia.com/software/breeze/> (16.12.2005).
- [Chase, 2004] Nicholas Chase, Use a Java applet to access remote Web services. <http://www-128.ibm.com/developerworks/java/library/x-jappws/> (17.12.2005).
- [Flashmagazine] The Flash history – Flashmagazine. <http://www.flashmagazine.com/413.htm> (18.12.2005).
- [Garrett, 2005] Jesse James Garrett, Ajax: A New Approach to WebApplications <http://www.adaptivepath.com/publications/essays/archives/000385.php> (18.12.2005).
- [Giacomo Guilizzoni *et al.*, 2005] Giacomo Guilizzoni, Brian Lesser, Joey Lott, Robert Reinhardt and Justin Watkins, *Flash Communication Server*. O'Reilly, 2005.
- [Gosling *et al.*, 2000] James Gosling, Bill Joy, Guy Steele and Gilad Bracha, *The Java Language Specification*. Addison Wesley, 2000.
- [Harold, 2004] Elliotte Rusty Harold, *Java Network Programming*. O'Reilly, 2004.
- [Hibbs, 2005] Ajax on Rails. http://www.onlamp.com/pub/a/onlamp/2005/06/09/rails_ajax.html (17.12.2005).
- [HitsLink] Market share for browsers, operating systems and search engines. <http://marketshare.hitslink.com/report.aspx?qprid=0> (16.12.2005).

- [Jacobson and Jacobson, 2001] Dov Jacobson and Jesse Jacobson, *Flash and XML: A Developer's Guide*. Addison Wesley Professional, 2001.
- [Languages for the Java VM] Languages for the Java VM.
<http://www.robert-tolksdorf.de/vmlanguages.html> (17.12.2005).
- [loadVariables] Macromedia Flash Support Center, loadVariables.
http://www.macromedia.com/support/flash/action_scripts/actionsript_dictionary/actionsript_dictionary425.html (17.12.2005).
- [Macromedia] Macromedia Flash Player Version Penetration.
http://www.macromedia.com/software/player_census/flashplayer/version_penetration.html (16.12.2005).
- [Moock, 2001] Colin Moock, *ActionScript: The Definitive Guide*. O'Reilly, 2001.
- [Muck, 2003] Tom Muck, *Flash Remoting: The Definitive Guide*. O'Reilly, 2003.
- [Nielsen, 1999] The Top Ten *New* Mistakes of Web Design
<http://www.useit.com/alertbox/990530.html> (17.12.2005).
- [Plant and Cleveland, 2001] Darrel Plant and Robert Cleveland, *Special Edition Using Macromedia Flash 5*. Que, 2001.
- [Sormunen and Pennanen, 2004] Eero Sormunen and Sami Pennanen, The challenge of automated tutoring in Web-based learning environments for information retrieval instruction. *Information Research* 9 2 (2004).
 Available: <http://informationr.net/ir/9-2/paper169.html>.
- [Unity] moock>> unity. <http://moock.org/unity/> (17.12.2005).
- [urlencode] PHP Manual, urlencode. <http://fi2.php.net/urlencode> (16.12.2005).
- [Wikipedia: Ajax] Ajax – Wikipedia, the free encyclopedia-
<http://en.wikipedia.org/wiki/AJAX>(18.12.2005).
- [Wikipedia: JDBC] Java Database Connectivity – Wikipedia, the free encyclopedia. <http://en.wikipedia.org/wiki/JDBC> (17.12.2005).
- [Wikipedia: Same origin policy] Same origin policy – Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Same_origin_policy (18.12.2005).
- [XMLSocket] Macromedia Flash Support Center, XMLSocket.
http://www.macromedia.com/support/flash/action_scripts/actionsript_dictionary/actionsript_dictionary860.html (16.12.2005).
- [XULplanet: Server Push] XULplanet – Server Push and Server Sockets
<http://xulplanet.com/tutorials/mozsdk/serverpush.php> (18.12.2005).
- [XULplanet: XMLHttpRequest] XULplanet – XMLHttpRequest
<http://www.xulplanet.com/references/objref/XMLHttpRequest.html> (18.12.2005).

Tietotekniikan hyödyntäminen pienten pirkanmaalaisten hoiva-alan yritysten liiketoiminnassa

Janne Siltanen

Tiivistelmä.

Tietoteknologian räjähdysmäinen kasvu viime vuosien aikana on asettanut haasteita yritysmaailmalle. Etenkin pienten ja keskisuurten yritysten kohdalla nopean kasvun mukana pysyminen on vaikeaa. Tässä tutkielmassa käsiteltiin pienten pirkanmaalaisten hoiva-alan yritysten tämänhetkistä tietoteknistä tilaa ja yritysten näkemystä tietotekniikan hyödyntämisestä omassa toiminnassaan. Tutkimuksessa pyrittiin kartoittamaan menestyksekkääseen tietotekniikan hyödyntämiseen johtavia tekijöitä. Hoiva-alaryittäjien näkemyksiä oman yrityksensä tietoteknisestä tilasta kartoitettiin sähköpostin välityksellä lähetetyllä kyselyllä, johon vastasi noin viidesosa kohdeyrityksistä.

Avainsanat ja -sanonnat: Tietotekniikan hyödyntäminen, hoiva-ala

CR-luokat: J.3, H.4.1

1. Johdanto

Tietoteknologia on kehittynyt Suomessa suurin harppauksin viime vuosina. Hänninen ja muut [2001] toteavat, että hyvinvointialan osalta muutos tarkoittaa tietotekniikan laaja-alaista hyödyntämistä ja uusien välineiden kehittämistä tukemaan muun muassa palveluntarjontaa ja palvelujen laatua.

Jos kysymyksessä olisi vain uusien tietoteknisten välineiden käyttöönotto, hyvinvointipalvelut voitaisiin tuottaa perinteisin toimintatavoin. Meneillään oleva teknologinen ja yhteiskunnallinen muutos merkitsee kuitenkin rakenteellisesti suurta toimintatapojen ja -edellytysten muutosta myös sosiaali- ja terveysalalla. Merkittävä osa muutosta on toimintaprosessien ja palvelukäytäntöjen muutos. [Hänninen et al., 2001].

Tietojärjestelmien levinneisyydestä ja käytöstä suomalaisissa sosiaali- ja terveydenhuollon yksiköissä on saatavana hyvin vähän selvityksiin perustuvaa tietoa. Suomessa on kuitenkin tehty terveydenhuollon tietojärjestelmiin liittyviä kartoitustutkimuksia, esimerkiksi Suomen Kuntaliitto ja Sosiaali- ja terveysalan tutkimus- ja kehittämiskeskus (Stakes) toteuttivat vuoden 2001 lopulla Sosiaali- ja terveydenhuollon tietojärjestelmäkartoituksen, joka osaltaan korjaa tiedonpuutetta tietojärjestelmien käytön osalta. [Hartikainen et al., 2002].

Tietoteknisten valmiuksien kehittämiseksi Suomessa on käynnistetty useita terveydenhuollon tietoteknologiaan liittyviä hankkeita. Esimerkkinä tällaisesta hankkeesta voidaan mainita Satakunnan Makropilotti, jota voidaan pitää

edelläkävijänä laajemmille terveydenhuollon aluetietojärjestelmähankkeille. Hankkeen tärkeänä tavoitteena oli saumattomien palveluketjujen toimintamallien ja näitä tukevan tietotekniikan kehittäminen. [Liikanen, 2002].

Puhuttaessa saumattomasta hoito- ja palveluketjusta tarkoitetaan tilannetta, jossa asiakkaan liikkuminen ja tietojen siirtyminen sosiaali- ja terveydenhuollon palvelujärjestelmässä on organisaatorajat ylittävää ja se tapahtuu sujuvasti ja saumattomasti [Nykänen, 2003].

Kokonaisvaltainen hoito ja palvelu saavutetaan, kun tietotekniikka yhdistää paitsi terveydenhuollon myös sosiaalihuollon ja -vakuutuksen sekä mahdollisesti yksityisetkin palveluntarjoajat [Liikanen, 2002].

Pirkanmaalla ei ole tehty pieniin hoiva-alan yrityksiin kohdistuvaa tarkempaa tietoteknistä kartoitusta. Esimerkiksi alueella tehdyt tietojärjestelmäkartoitukset koskevat pääosin massiivisia sairaalajärjestelmiä, jotka eivät tule kysymykseen pienten yritysten kohdalla. Tässä tutkimuksessa käsitellään mahdollisten tietojärjestelmien käyttämisen lisäksi myös kohdeyritysten henkilökunnan atk-taitoja ja tulevaisuuden näkemyksiä tietotekniikan osalta.

2. Aiemmat hankkeet ja tutkimukset

Vuonna 1995 julkaistiin sosiaali- ja terveydenhuollon tietoteknologian hyödyntämisstrategia, jossa kuvattiin sosiaali- ja terveystalouden visioita tietoteknologian hyödyntämisestä sosiaali- ja terveydenhuollossa ja linjattiin tietoyhteiskunnan kehittämisen kannalta keskeisiä toiminnallisia tavoitteita. [Sosiaali- ja terveysministeriö, 1995].

Vuoteen 2001 mennessä oli jo käynnistetty lähes sata sosiaali- ja terveydenhuollon tietoteknologian hyödyntämisstrategiaa toteuttavaa teknologian hyödyntämishanketta [Hänninen et al., 2001].

Eräs merkittävä, Suomen oloissa poikkeuksellisen suuri sosiaali- ja terveydenhuollon kehittämishanke oli vuonna 1998 käynnistetty Satakunnan makropilotti [Liikanen, 2002]. Hankkeen toimeenpanijoina olivat Sosiaali- ja terveysministeriö, Sosiaali- ja terveystalouden tutkimus- ja kehittämiskeskus (Stakes), Kansanterveyslaitos, Työterveyslaitos, Kauppa- ja teollisuusministeriö, teknologian kehittämiskeskus (Tekes), Kela ja Suomen Kuntaliitto. Hankkeeseen osallistui muun muassa seitsemän kuntaa, terveyskeskuksia ja sairaanhoitopiirejä. [Ohtonen, 2002]. Satakunnan makropilotti koostui noin 20 erillisestä osaprojektista. Sen tavoitteena oli kehittää uusiin teknisiin ratkaisuihin perustuvia innovaatioita. [Hänninen et al., 2001].

Vuoden 2001 syksyllä käynnistettiin valtakunnallinen Hoiva-alan palveluverkko -hanke (HOPE) Kauppa- ja teollisuusministeriön rahoittamana. Valtakunnallisesti HOPE-hankkeen koordinoinnista vastaa Terveystalouden ja Sosiaalialan

Yrittäjät TESO ry. Hankkeen tavoitteeksi on asetettu tietoverkkojen monipuolinen hyödyntäminen hoiva-alan yrittäjien ja yrittäjäksi pyrkivien liiketoiminnassa. Hankkeessa kootaan internetiin Hoivayrittäjät-verkkopalvelu, (<http://www.hoivayrittajat.com>). Kyseinen valtakunnallinen palveluhakemisto käsittää koko Suomen kattavan hoiva-alan yritysrekisterin sekä hoivayrittäjien tietopankin. HOPE-hankkeessa käynnistetään laatutyö yrityksissä. Hanke sisältää myös atk-koulutusta. Täten vahvistetaan uusien hoivayritysten liiketoimintaa, erityisesti markkinointia, verkostoitumista ja henkilöstön osaamista. [Viitaniemi 2001; Sinervo, 2004].

Sosiaali- ja terveydenhuollon tietojärjestelmien käyttöön liittyviä tietoja on saatavilla kaiken kaikkiaan hyvin vähän. Vuoden 2001 lopulla Suomen Kuntaliitto ja Stakes, Osaavien Keskusten Verkostosta (OSVE), toteuttivat Sosiaali- ja terveydenhuollon tietojärjestelmäkartoituksen. [Hartikainen et al., 2002]. Kysely oli melko laaja; siihen vastasi 160 perusterveydenhuollon ja erikoissairaanhoidon yksikköä ja 158 kunnan sosiaalitoimi.

Vuoden 2001 Sosiaali- ja terveydenhuollon tietotekniikkakartoitus on jatkoa samojen tahojen vuonna 1999 tekemille vastaaville kartoituksille. Kartoituksen tarkoituksena on saada tietoa sosiaali- ja terveydenhuollon tietotekniikan arkkitehtuurista, ohjelmistokannasta sekä toiminnan organisoinnista tietotekniikan avulla. Tavoitteena on saada syntymään tietoteknologian käyttöä ja levinneisyyttä kuvaava aikasarja toistamalla kartoituksia säännöllisin väliajoin. [Hartikainen et al., 2002]

3. Tutkimuksen tarkoitus

Nopea kehitys luo uusia mahdollisuuksia tietotekniikan hyödyntämiselle. Pienten yritysten kohdalla tietoteknisten mahdollisuuksien hyödyntäminen ei ole kuitenkaan itsestään selvää.

Teknologian nopeassa kehityksessä monet pienet hoiva-alan yritykset ovat muutospaineiden alla. Ylemmiltä tahoilta tulee erilaisia tietotekniikkaan liittyviä ohjeita ja määräyksiä, joita yrittäjät ovat velvoitettuja noudattamaan, esimerkkeinä muun muassa siirtyminen sähköisen laskutuksen piiriin määräaikaan mennessä. Lisäksi hoiva-alan yritysten ja julkisen sektorin yksiköiden välillä pyritään yhä enenevässä määrin siirtymään sähköiseen tiedonvälitykseen.

Tällaiset muutokset vaativat yritykseltä paljon resursseja, jolloin erityisesti pienet yritykset ovat vaikeuksissa. Tietotekniikan hyödyntäminen tuottaa kuitenkin tuloksia vasta myöhemmin. Päivittäiset toiminnot nopeutuvat, kun uusiin metodeihin totutaan. Prosessin alkuun saattaminen saattaa kuitenkin olla työlästä ja vaikeaa monestakin syystä. Vaikuttavia tekijöitä voivat olla

muun muassa yrityksen henkilökunnan muutosvastarinta. Aiemmin totutuista, tutuista ja turvallisista metodeista ei haluta luopua. Tietotekniikka voi olla pelottava asia ja sen käyttöönotto vaikeaa ja hidasta.

Monessa pienessä hoiva-alan yrityksessä käytetään päivittäin tietokonetta, mutta varsinaisen hoitotyön apuvälineenä tietotekniikan käyttö on vielä vähäistä. Tätä on pyritty edistämään tekemällä yritysten käyttöön muun muassa erilaisia sähköisiä lomakkeita. Markkinoilla on myös olemassa valmiita ohjelmistoja, jotka on suunniteltu helpottamaan päivittäistä työskentelyä, esimerkiksi potilastietojen ylläpitämistä.

Tässä tutkielmassa tarkoitukseni oli kartoittaa pirkanmaalaisten hoiva-alan yritysten tietoteknistä tilannetta ja selvittää, mitkä asiat vaikuttavat onnistuneeseen tietotekniikan hyödyntämiseen yrityksen liiketoiminnassa. Huomioitavia seikkoja olivat yrityksen sekä tekniset, että henkiset valmiudet. Lisäksi tavoitteenani oli kartoittaa yritysten tulevaisuuden näkymiä tietotekniikan osalta.

Käsittämäni aihe liittyy läheisesti omaan työhöni. Olen työskennellyt kesällä 2005 yrityksessä, jonka tehtävänä on toimia asiantuntijana muun muassa pienten hoiva-alan yritysten ohjelmistohankinnoissa. Yrityksellä on merkittävä rooli myös yritysten tietotekniikan koulutukseen liittyvissä asioissa.

4. Aineisto ja sen keräys

Tässä tutkimuksessa kartoitettiin pienten pirkanmaalaisten hoiva-alan yritysten tietoteknistä tilannetta ja tulevaisuuden näkymiä. Kohdeyritykset saavutettiin Hoivayrittäjät-verkkopalvelun kautta. Tämä valtakunnallinen hakemistopalvelu sisältää luvanvaraisia ja ilmoitusvelvollisia sosiaali- ja terveystalvyrityksiä. Palvelusta valittiin kohdealueeksi Pirkanmaa, josta yrityksiä löytyi yhteensä 75 kappaletta. Tästä määrästä ajantasaiset tiedot oli 60 hoiva-alan yrityksellä.

Tutkimusmenetelmänä käytettiin kyselyä, joka toteutettiin sähköpostin välityksellä. Kyselyä varten laadittiin viiden kysymyksen kysymyssarja, joihin pyydettiin lyhyitä vastauksia. Oletuksena oli, että vastausaktiivisuus ei ole erityisen korkea, koska kohdeyrityksissä on usein rajalliset resurssit ylimääräiseen toimistotyöhön. Tästä syystä kysely pyrittiin pitämään mahdollisimman yksinkertaisena ja vaivattomana vastata. Kysymykset liitettiin sähköpostiviestin runkoon, liitetiedostoja tai erillisiä www-lomakkeita ei kyselyyn liitetty. Kysymykset toteutettiin tarkoituksella avoimina, vaikka tähän sisältyi riski, etteivät vastaukset olisi täysin vertailukelpoisia. Avoimilla kysymyksillä mahdollistettiin kuitenkin monipuolisemmat vastaukset.

Tutkimusta varten laadittu kysely lähetettiin Hoivayrittäjät-verkkopalvelun ylläpidon luvalla postituslistalle, joka sisälsi Pirkanmaan alueella toimivat HOPE-projektissa mukana olevat hoiva-alan yrittäjät.

Yrityksille lähetettiin seuraavat kysymykset:

1. Kerro lyhyesti yrityksestäsi (palvelutyyppi, henkilökunnan määrä, asiakasmäärä).
2. Miten hyödynnätte tietotekniikkaa yrityksenne toiminnassa ja koetteko tietotekniikasta olevan hyötyä näiden kokonaisuuksien hoidossa? (hallinnolliset asiat; laskutus, sähköposti jne., ja varsinainen hoitotyö; asiakasrekisterit, asiakaskontaktit ym.)
3. Käyttääkö yrityksenne valmiita ohjelmistoja (rekisterit, lomakkeistot yms.) hoitotyön apuna? Jos ei, miten kirjaaminen tapahtuu?
4. Millaiseksi koette henkilökunnan atk-aidot ja yrityksenne valmiudet ohjelmistojen käyttöön?
5. Minkälaiset ovat yrityksenne tulevaisuuden suunnitelmat ja tarpeet tietotekniikan suhteen? (esim. laitehankinnat, ohjelmistohankinnat, atk-koulutukset)

Ensimmäisellä kysymyksellä pyrittiin kartoittamaan yrityksen toimialaa ja henkilöstön ja asiakkaiden määrää. Tämä tieto helpottaa tietojen analysointia muiden kysymysten osalta. Yritysten tietotekniset tarpeet eroavat jonkin verran toisistaan eri palvelutyyppien välillä. Palvelutyyppin perusteella voidaan myös kartoittaa yrityksen tietoteknisiä tarpeita. Tämän avulla voidaan tehdä päätelmiä siitä, miten tietotekniikkaa hyödynnetään ja miten sitä voitaisiin edelleen hyödyntää.

Toisella kysymyksellä haluttiin tiedustella, miten tietotekniikka ylipäätään käytetään yrityksessä ja koetaanko sen käyttö hyödylliseksi vai hankalaksi. Tämä kysymys osoittaa, pystytäänkö tietotekniikkaa hyödyntämään mahdollisesti myös varsinaisessa hoitotyössä.

Kolmas kysymys kertoi, millaisia menetelmiä yritykset käyttävät päivittäisessä kirjaustyössään. Tässä kysymyksessä selviää myös se, kuinka moni käyttää valmisohjelmistoja ja kuinka moni turvautuu muihin keinoihin.

Neljännellä ja viidennellä kysymyksellä pyrittiin kartoittamaan yrityksen omaa näkemystä henkilökuntansa tietoteknisestä osaamisesta ja tulevaisuuden näkymistä. Kysymykset ovat tulkinnanvaraisia, sillä mielipiteet voivat vaihdella yrityksittäin. Toisen mielestä hyvä taso saattaa olla toisen mielestä vain kohtalainen. Tästä syystä tulokset eivät ole suoraan vertailukelpoisia, mutta niistä nähdään yrityksen oma suhtautuminen kyseiseen asiaan.

Kyselyyn annettiin vastausaikaa kaksi viikkoa ja vastaukset pyydettiin sähköpostitse. Kyselystä saatavilla vastauksilla pyrittiin saamaan kuva pienten

pirkanmaalaisten hoiva-alan yritysten tämänhetkisistä tietoteknisistä valmiuksista sekä tulevaisuuden näkymistä.

5. Tulokset

Sähköpostikyselyyn vastanneita yrityksiä oli yhteensä 14 kappaletta kuudestakymmenestä. Vastausaktiivisuus oli näin ollen 23 %. Vastanneet yritykset olivat vastanneet melko kattavasti kaikkiin kysymyksiin. Vastanneiden yritysten koot vaihtelivat yhden hengen yrityksestä 16 henkilöä työllistävään yritykseen. Vastauksista koottu yhteenveto on liitteessä 1.

5.1. Hoiva-alan yrittäjien vastaukset kyselyyn

Ensimmäisen kysymyksen osalta voidaan todeta, että kyselyyn vastanneiden yritysten osuus kattoi hyvin lähes kaikki hoiva-alan eri palvelutyypit. Tämä on tutkimuksen kannalta hyvä asia. Vastanneiden yritysten joukossa oli muun muassa lastensuojeluun, sosiaalihuollon asumispalveluihin, lasten päivähoidon sekä terapiapalveluihin erikoistuneita yrityksiä. Yritysten asiakasmäärä vaihteli myös palvelutyypin mukaan. Luonnollisesti esimerkiksi asumispalveluita tarjoavien yritysten kohdalla asiakasmäärä on pysyvämpi, kun taas esimerkiksi hierontapalveluita tuottavien yritysten kohdalla asiakasmäärät vaihtelevat kuukausittain. Asiakasmäärät vaihtelivat kuukauden ajanjaksolla muutamasta henkilöstä aina 250 henkilöön, riippuen yrityksen palvelutyypistä.

Toisella kysymyksellä yrityksiltä tiedusteltiin tietotekniikan hyödyntämistä sekä sitä, kokevatko yritykset tietotekniikan olevan hyödyllistä. Kaikkien kyselyyn vastanneiden yritysten mielestä tietotekniikka koetaan hyödyllisenä. Sitä käytetään yrityksissä eri tavoin. Alle puolet vastanneista yrityksistä hyödyntää tietotekniikka päivittäisessä hoitotyössä, osa käyttää tietotekniikkaa pelkästään hallinnollisiin tehtäviin. Vastauksista voidaan päätellä, että tietoteknistä koulutusta olisi syytä lisätä. Tietotekniikka koetaan hyödylliseksi, ja sen käyttämisestä koetaan olevan hyötyä. Monella yrityksellä taidot ovat riittämättömiä tietotekniikan hyödyntämiseen laajemmin.

Seuraavassa erään kyselyyn vastanneen yrityksen vastaus kysymyksen numero kaksi:

”Tietotekniikan hyödyntäminen on suhteellisen vähäistä vielä. Laskutus tehdään tietokoneella ja sähköposti on käytössä yhteydenpitoa varten. Asiakastietojärjestelmä ei ole vielä koneella kokonaisuudessaan, nyt kylläkin ollaan tällaista ohjelmaa juuri hankkimassa. Aukkaiden hoitotiedotteet ovat koneella. Stakesin HILMO-asiakaslaskenta tapahtuu tietokoneen kautta. Työvuorolistat tehdään myös tietokoneen avulla. Mielestäni jollain tapaa tietokone on apuna hoitotyössä ja se on ihan hyövä juttu, vaikka ikinähän tietokone ei voi ihmistä täysin korvata hoitotyössä, että rajansa silläkin on

missä sitä käytetään ja missä ei. Mutta juuri tällaiset esim. laskutus ja työvuorolistojen teko koneella on ihan tarpeellista.”

Kolmannella kysymyksellä haluttiin kartoittaa yritysten tilannetta valmiiden ohjelmistojen suhteen. Hoiva-alan yrityksissä tehdään paljon asiakkaiden tietoja käsittäviä kirjauksia. Tässä kohdassa selvitettiin myös kirjausmetodeja, mikäli valmiita ohjelmistoja ei ole käytössä.

Kyselyyn vastanneiden yritysten joukko jakaantuu lähes kahtia ohjelmistojen käytön suhteen: noin puolet käyttää, toinen puoli ei. Hoitotyötä tukevien ohjelmistojen käyttömahdollisuus riippuu suuresti yrityksen palvelutyypistä. Kaikkien yritysten osalta ohjelmistoista ei ole suoranaista hyötyä. Osa yrityksistä oli kehittänyt itse omia kirjauksiaan helpottavia ratkaisuja, esimerkiksi Excel-lomakkeita, jotka soveltuvat kyseisen yrityksen tarpeisiin.

Seuraavassa esimerkkejä vastauksista kysymykseen numero kolme:

”Kirjaaminen tapahtuu seuraavasti: Jokaiselle asiakkaalle on viikko, johon päivittäiset tapahtumat kirjataan. Viikosta kootaan kerran kuukaudessa kk-raportti.”

”Emme käytä. Meillä ei ole siihen tarvetta työn luonteen vuoksi. Olemme laatineet itsellemme sopivat kirjaamisen apuvälineet.”

Neljäs kysymys käsitteli henkilökunnan atk-taitojen ja -valmiuksien arviointia yrityksen omasta näkökulmasta, eli tässä kysymyksessä pyydettiin nimenomaan yrityksen omaa arviota. Vastauksia analysoitaessa tulee muistaa, että yrityksen näkemykset omasta osaamistasostaan eivät suoranaisesti ole vertailukelpoisia, mutta ne kertovat, miten luottavaisesti yritykset suhtautuvat omiin taitoihinsa.

Erään kyselyyn vastanneen yrityksen vastaus kysymykseen numero neljä:

”Henkilökunnan atk-aidot ovat hyvin vaihtelevat. Lähinnä nuoremmat osaavat enemmän ja ovat rohkeampia tekemään kaikkea koneella. Vanhemmilla on vähän arkuutta koneen käytössä. Koulutuksia on pidetty yleisesti koneen käytöstä ja esim. sähköpostin käyttämisestä. Uuden ohjelman käyttöönottoon joutuu tietysti uhraamaan koulutustunteja, jotta kaikille työntekijöille on selvää miten ohjelmaa käytetään. Ainoa vaihtoehto on mielestäni se, että jokainen joutuu tekemään töitä koneen kanssa, kaikki oppii ajan kanssa.”

Viidennellä kysymyksellä haluttiin selvittää yrityksen tulevaisuuden suunnitelmia ja tarpeita tietotekniikan suhteen (esimerkiksi mahdolliset laite- ja ohjelmistohankinnat, atk-koulutukset). Iso osa yrityksistä koki atk-koulutuksen olevan tarpeellista, ja moni yritys olikin suunnitellut järjestävänsä koulutuksia tulevaisuudessa. Monet yritykset kokevat atk-taitonsa riittäviksi perusasioihin, mutta silti koulutus ajateltiin tarpeelliseksi. Yhdessä vastauksessa myös tietoturvan merkitystä korostettiin.

Erään yrityksen vastaus kysymykseen numero viisi:

”Ohjelmiston hankkiminen on ollut harkinnassa. Kahdelta alan toimittajalta on tullut esitteitä ja tarjouksia. Laitehankinnat yms. tulee ratkaistavaksi siinä kohtaa jos ohjelma hankitaan. Ja ohjelmatoimittajilla on olemassa sitten koulutusta, jota varmasti käytämme hyväksi jos ohjelman hankimme. Muuta yleistä atk-koulutusta emme ole harkinneet.”

6. Pohdintaa

Sähköpostikysely lähetettiin 60 hoiva-alan yrittäjälle, joista 14 yritystä vastasi haastatteluun. Täten kyselyn vastausaktiivisuus jäi siis 23 prosenttiin. Kyseinen määrä vastauksia antaa kyllä kuvaa pirkanmaalaisten hoiva-alan yritysten tietoteknisestä tilanteesta, mutta koko maata koskevia johtopäätöksiä siitä ei voi tehdä.

Yritysten melko heikkoa vastausaktiivisuutta voidaan selittää pienten hoiva-alan yritysten vähäisillä resursseilla. Suurin osa henkilökunnan ajasta kuluu hoitotyössä, joten kyselyihin vastaamiseen ei jää aikaa. Vaikka vastausten täyttämiseen ei mene paljon aikaa, koetaan se todennäköisesti kuitenkin työläänä. Tutkimuksesta mahdollisesti saatavia hyötyjä ei osata arvostaa. Myös kyselyn toteutustapa saattoi osaltaan karsia vastaajien määrää. Sähköpostitse välitetty kysely ei ole toimiva ratkaisu ainakaan niiden yritysten osalta, joilla sähköpostin käyttö on vähäistä. Vastausten määrää ei onnistuttu kasvattamaan, vaikka kysely lähetettiin Hoivayrittäjät-verkkopalvelun postituslistan kautta ja mukaan liitettiin vielä kyseisen verkkopalvelun ylläpitäjien kehoitus kyselyyn vastaamiseksi. Yrityksille lähetettiin lisäksi muistutusviesti viikkoa ennen määräajan täyttymistä, minkä jälkeen vastauksia tuli vielä muutaman yrityksen osalta.

Kyselyn vastausaktiivisuus olisi saattanut olla parempi, mikäli kysely olisi toteutettu toisella tavalla. Esimerkiksi haastattelu puhelimitse tai käynti jokaisessa yrityksessä oli tuottanut todennäköisesti paremman lopputuloksen, mutta ne olisivat vaatineet liikaa aikaa ja resursseja tutkimuksen toteuttajilta. Tutkimuksen toteutuksessa käytettiin siis avoimia kysymyksiä, jotka saattavat kuitenkin aiheuttaa hajontaa vastauksissa. Ne ovat osittain tulkinnanvaraisia, eikä niitä tästä johtuen voida pitää kaikilta osin vertailukelpoisina. Tässä tutkimuksessa tärkeää oli kuitenkin saada yleiskuva tutkimuksen kohdeyritysten tietoteknisestä tilasta, joten tähän tarkoitukseen menetelmä oli sopiva.

Myös huolellisesti laadittu www-lomake olisi saattanut olla toimivampi ratkaisu avoimiin kysymyksiin verrattuna. Toisaalta www-lomake olisi saattanut karsia vastauksia entisestään, sillä tietotekniset taidot ovat osalla yrityksistä vielä melko alkeellisella tasolla. Kaikki ylimääräinen ponnistelu vastauksen jättämiseksi oli syytä karsia pois. Siksi lopulta tutkimuksessa päädyttiin

esittämään kysymykset yksinkertaisesti sähköpostin viestin rungossa. Kyselymenetelmää valittaessa oli syytä huomioida myös se, että ko. menetelmiä ei voida sekoittaa keskenään. Täten valitaan ainoastaan yksi menetelmä, jota käytetään kaikkien kohdeyritysten osalta. Mikäli joihinkin kohdeyrityksiin olisi käytetty eri menetelmää kuin toisiin, eivät kyselystä saatavat vastaukset olisi olleet keskenään vertailukelpoisia.

Koska tutkimusten tuloksia ei voida kattavasti verrata valtakunnallisesti, tutkimuksen tieteellinen arvo ei näin ollen ole paras mahdollinen. Pirkanmaalaisten yritysten osalta päädyttiin kaikesta huolimatta melko hyvään arvioon. Vastanneita yrityksiä oli hyvässä suhteessa monesta eri palvelutyyppistä ja koska eri palvelutyypeillä on luonnollisesti erilaiset tietotekniset tarpeet, yritysten tietoteknisestä tilanteesta saatiin monipuolinen kuva.

Ajatus tietotekniikan hyödyntämisestä hoitotyössä tuntui joistakin yrityksistä vieraalta. Käsite *"hyödyntäminen hoitotyössä"* ei ollut kaikkien vastaajien yksiselitteinen. Osa vastaajista ei ajatellut ohjelmistojen hyödyntämisen, kuten muun muassa lääkityksen seuranta, hoitosuunnitelman laadintaa ym. sisältävien asiakasrekisteriohjelmien käytön olevan hoitotyöhön luokiteltavaa tietotekniikan käyttöä. Sitä voidaan kuitenkin pitää hoitotyötä tukevana apuvälineenä, jonka myötä perinteiselle hoitotyölle jää enemmän aikaa. Heidän mielestään lääkkeiden antaminen, läheisyys ym. ovat hoitotyön ydintä, jota ei voida koneella korvata.

Palvelutyyppi määrittää melko pitkälti, koetaanko tietotekniikka hyödylliseksi. Yritysten toimenkuvat vaihtelevat, samoin näkemykset tietotekniikan hyödyntämismahdollisuuksista.

Laitteistot olivat vastanneilla tarpeita vastaavia. Osa yrityksistä oli valmiita tekemään uusia laitehankintoja tarvittaessa. Ainakin niiden yritysten kohdalla, jotka näin vastasivat, voidaan sanoa, että tietotekniikka koetaan tarpeelliseksi. Yritysten tietokoneen käyttötaidot vaihtelivat. Nuoret ovat keskimäärin rohkeampia ja sen myötä myös taitavampia koneen käyttäjiä. Vanhemmat henkilöt ovat puolestaan arempia koneen käytössä. Toisaalta riittävällä koulutuksella myös vanhemmat ihmiset tulevat hyvin toimeen tietotekniikan kanssa. Vanhemmilla ihmisillä tietotekniikkaan siirtymiseen vaikuttavat myös henkiset arvot. Tätä niin sanottua muutosvastarintaa esiintyy joidenkin yritysten henkilökunnassa. Jos ei ennenkään ole koneita tarvittu, miksi nyt?

Edellä mainitussa tilanteessa täytyy henkilökunnalle pitää esimerkiksi motivointipäiviä, joissa tietokoneen käyttöön tutustutaan ja kerrotaan siitä saatavista hyödyistä. Kun muutosvastarinta voidaan murtaa, mitään suurempia esteitä tietotekniikan tehokkaalle hyödyntämiselle, ainakaan henkisellä puolella, ei ole.

Lähes kaikkien vastanneiden yritysten mielestä tietotekniikan koulutus on tärkeää ja sitä tarvittaisiin lisää. Koulutukset ovat tarpeen myös hieman tietoteknisesti edistyneimmille yrityksille. Koulutusten avulla yritys voi tehostaa entisestään tietotekniikan hyödyntämistä. Työskentely yrityksessä on tehokkaimmillaan, kun tietotekniikkaa voidaan luontevasti käyttää hoitotyötä tukevana apuvälineenä, ilman että sen koetaan haittaavan työskentelyä.

Ohjelmistot ovat melko hintavia, joten tästä syystä ne jäävät monilta pieniltä hoiva-alan yrittäjiltä hankkimatta vaikka ne koettaisiinkin käyttökelpoisiksi ja yrityksen kannalta hyödyllisiksi. Toisaalta kaikkien kyselyyn vastanneiden yritysten osalta ei hoitotyötä tukevista ohjelmista voida sanoa olevan hyötyä työn erilaisen luonteen vuoksi.

Hartikaisen ja muiden [2002] mukaan tietotekniikan käyttö on merkittävästi lisääntynyt sosiaali- ja terveydenhuollossa. Kaikesta huolimatta tietotekniikan hyödyntäminen on kuitenkin vielä melko puutteellista ja standardoitumaton. Sosiaali- ja terveydenhuollossa olevat lukuisat, keskenään yhteen sopimattomat järjestelmät hidastavat merkittävästi sosiaali- ja terveydenhuollon toiminnan kehittämistä ja verkottumista. Tietojärjestelmien keskittämällä voitaisiin vapauttaa voimavaroja oleellisempaan käyttöön. [Sosiaali- ja terveystieteiden ministeriö, 2003].

Teknologia saattaa muuttaa omaehtoisesti työkäytäntöjä, jos tietoteknologian käyttöönottoa ei tueta asianmukaisella, riittävällä koulutuksella ja ohjauksella. Tällöin jo valmiiksi kuormittuneelle henkilöstölle aiheutettaisiin lisää paineita. [Hartikainen et al., 2002]. Myös Nykäsen [2002] mukaan koulutus on ratkaisevassa roolissa, etenkin uusia työtapoja ja käsitteitä käyttöönotettaessa.

7. Yhteenveto

Nopeasti kehittynyt tietoteknologia mahdollistaa tietotekniikan tehokkaan hyödyntämisen muun muassa hoiva-alalla. Pienet yritykset ovat muutospaineiden alla, sillä ylemmät tahot asettavat laite- ja ohjelmistohankintoja edellyttäviä vaatimuksia. Tässä tutkielmassa käsiteltiin pienten pirkanmaalaisten hoiva-alan yritysten tietoteknistä tilannetta. Pienten yritysten on usein lähes mahdotonta vastata haasteisiin vähäisten resurssien vuoksi.

Tämän tutkimuksen kohdeyritykset valittiin valtakunnallisen Hoivayrittäjät-verkkopalvelun kautta ja Pirkanmaan alueelta löytyi 60 ajantasaisin yhteystiedoin listattua yritystä. Kohdeyrityksille lähetettiin Hoivayrittäjät-verkkopalvelun postituslistan kautta sähköpostikysely, jonka perusteella pyrittiin kartoittamaan yritysten tämänhetkistä tilannetta sekä tulevaisuuden

näkymiä muun muassa tietokonelaitteistojen, -ohjelmistojen ja atk-taitojen suhteen.

Kyselyn vastausaktiivisuus jäi melko alhaiseksi. Vastanneiden yritysten osuus oli 23 % kaikista kyselyn saaneista. Vastanneet yritykset kattoivat kuitenkin hyvin lähes kaikki hoiva-alan palvelutyypit, joten kyselystä saatujen vastausten perusteella oli mahdollista arvioiden yleistää tilannetta Pirkanmaan osalta. Valtakunnalliseen vertailuun vastausaktiivisuus olisi voinut kuitenkin olla korkeampi kattavamman kuvan saamiseksi.

Yritykset kokivat henkilökuntansa tietokoneen peruskäyttötaidot keskimäärin kohtalaisiksi. Vastanneiden yritysten osalta myös laitteistot olivat ajanmukaisia. Aktiivisimmat olivat valmiita tekemään tarvittaessa myös uusia laitehankintoja. Hoitotyötä tukevia ohjelmistoja ei vielä hyödynnetty kovin kattavasti. Tähän suuresti vaikuttava tekijä on ohjelmistojen korkea hinta suhteessa pienten yritysten vähäisiin resursseihin. Kaikkien vastanneiden yritysten osalta koulutusta pidettiin tärkeänä ja osalla yrityksistä olikin suunnitelmissa hankkia lisäkoulutusta lähitulevaisuudessa.

Vastaavanlaisen tutkimuksen voisi tehdä myös valtakunnallisesti, jolloin saataisiin kattavampaa ja ajankohtaista tietoa tietotekniikan hyödyntämisestä.

Viiteluettelo

- [Hartikainen et al., 2002] Kauko Hartikainen, Sirpa Kuusisto-Niemi ja Elisa Lehtonen, Sosiaali- ja terveydenhuollon tietojärjestelmäkartoitus 2001. Osaavien keskusten verkoston julkaisuja 1/2002. Helsinki 2002.
- [Hänninen et al., 2001] Esko Hänninen, Mari Koivunen ja Piia Paaso, Hyvinvointia teknologiahankkeilla. Hyviksi arvioitujen toimintamallien ja teknisten ratkaisujen käyttöönotto ja juurrutus. Sosiaali- ja terveysministeriön julkaisuja 2001:11. Helsinki 2001.
- [Liikanen, 2002] Hanna Liikanen, *Tietotekniikka kehittää sosiaali- ja terveysalaa*. Tampere University Press, 2002.
- [Nykänen, 2002] Pirkko Nykänen (toim.), Satakunnan Makropilotin tietoteknologian arviointi. Tampereen yliopisto, Tietojenkäsittelytieteen laitos, Raportti **B-2002-3**, Syyskuu 2002.
- [Nykänen, 2003] Pirkko Nykänen (toim.), Terveydenhuollon tietojärjestelmät. Tampereen yliopisto, Tietojenkäsittelytieteen laitos, Raportti **B-2003-7**, Joulukuu 2003.
- [Ohtonen, 2002] Jukka Ohtonen (toim.), Satakunnan Makropilotti: Tulosten arviointi. FinOHTA. Raportti 21/2002. Stakes, Helsinki 2002.

- [Sinervo, 2004] Leini Sinervo (toim.), Saumattomien toimintojen juurruttaminen. Juuria-hankkeen loppuraportti. Stakes, Aiheita 26/2004. Helsinki 2004.
- [Sosiaali- ja terveysministeriö, 1995] Sosiaali- ja terveysministeriö, Sosiaali- ja terveydenhuollon tietoteknologian hyödyntämisstrategia. Sosiaali- ja terveysministeriön työryhmämuistioita 1995:27. Helsinki 1995.
- [Sosiaali- ja terveysministeriö, 2003] Sosiaali- ja terveysministeriö, Saumattoman palveluketjun ja sitä tukevien hallintoratkaisujen ohjausryhmä. Suosituksia. Sosiaali- ja terveysministeriön työryhmämuistioita 2003:18. Helsinki 2003.
- [Viitaniemi, 2001] Marjo Viitaniemi, Hoiva-alan palveluverkkohanke, projekti-hakemus. ESR:n osittain rahoittamat projektit, 31.8.2000.

	Palvelutyyppi	Henkilökunnan määrä	Asiakasmäärä	Tietotekniikan hyödyntäminen hallinnollisissa asioissa	Tietotekniikan hyödyntäminen hoitotyössä	Ohjelmistojen käyttö	Henkilökunnan arvioidut atk-taidot	Tulevaisuuden näkymät
Yritys 1	kotisairaanhoido	1	*	sähköposti	oma systeemi	ei	vähäiset	koulutusta
Yritys 2	lasten päivähoito	6	40	sähköposti, myös palkanlaskuun, tilastointiin ja osittain laskutukseen käytetään tietokonetta	lapsilistat	ei, tarve vähäinen	hyvät	ei erityisiä suunnitelmia
Yritys 3	terveydenhuollon palvelut, mm. fysioterapia	14	250/kk	kattavasti ajanvarauksesta, asiakirjahallintoon ja potilaskortteihin, tiedottamiseen, sisäiseen tiedottamiseen, kalenteriohjelmiin, kännykkäyhteyksiin, kulunvalvontaan, varmuuskopioihin, arkistointiin	kts. hallinnollisista tähän kuuluvat	kyllä	melko hyvät, aktiivista/jatkuvaa kouluttautumista	atk-lisäkoulutusta lähitulevaisuudessa
Yritys 4	vanhusten asumispalvelut	7 + 6	13 + 10	laskutus, työvuorolistat, sähköposti	asiakastieto-ohjelmisto tulossa, asukkaiden hoitotiedotteet koneella, käytössä lisäksi Stakesin hoitoilmoitus- ja asiakaslaskenta-ohjelma (HILMO)	Stakesin hoitoilmoitus- ja asiakaslaskenta-ohjelma (HILMO), päivittäiset kirjaukset tehdään nykyisin vielä vihkoon	taidot vaihtelevat henkilökunnan keskuudessa, nuoret parempia, vanhemmat arempia	asiakastieto-ohjelmisto hankinnassa
Yritys 5	perhekoti	3 + sijaisia	3	laskujen maksuun, sähköpostiin, asiakasrekisteriin, kk-raporttien ja hoito- ja kasvatussuunnitelmien tekemiseen	kirjaukset tehdään asukaskohtaisiin vihkoihin, joista koostetaan kuukausittain kk-raportti	ei	atk-taidot riittävät perustoiimiin, kuten tekstinkäsittelyyn, se koetaan riittäväksi	ei koulutussuunnitelmia, laite- ja ohjelmistohankintoja tarvittaessa

	Palvelutyypit	Henkilökunnan määrä	Asiakasmäärä	Tietotekniikan hyödyntäminen hallinnollisissa asioissa	Tietotekniikan hyödyntäminen hoitotyössä	Ohjelmistojen käyttö	Henkilökunnan arvioidut atk-taidot	Tulevaisuuden näkymät
Yritys 6	sosiaalihuollon asumispalvelut	8 vakituista	24	sähköposti, verkkopankki, asiakatietojärjestelmä, aluetietojärjestelmä	HILKKA-asiakastietojärjestelmä, aluetietojärjestelmä	HILKKA-asiakastietojärjestelmä	hyvä peruskäyttötaito, koulutus kuitenkin tarpeellista	koulutusta aluetietojärjestelmän käyttöön, nykyisen ohjelmiston käyttökoulutusta
Yritys 7	koulutus- ja terapiapalvelut	1 + 2 osa-aikaista	yli sata vuodessa	sähköpostin käyttö yhteydenpidossa mm. yhteistyökumppaneihin ja henkilökunnan sisäinen informointi	ei varsinaisesti	ei, ei tarvetta työn luonteen vuoksi	kohtuulliset käyttötaidot, lisäopille tarvetta ja ennen muuta uudelle orientoitumiselle, lisäksi tietoa siitä miten tietotekniikka voisi hyödyntää (tehokkaammin)	sähköiseen laskutukseen siirtyminen, kotisivujen saattaminen helposti päivitettävään muotoon
Yritys 8	lastensuojelulaitos	7	6	sähköposti, asiakasrekisterit, raportointi	mm. itse laadittuja lomakkeita	ei juurikaan	tydyttävät / hyvät	ei erityisiä suunnitelmia
Yritys 9	sosiaalihuollon asumispalvelut	18	59	sähköposti, laskutus, laskujen maksu, tekstinkäsittely, työvuorosuunnittelu	omat lomakepohjat, ohjelmiston hankinta harkinnassa	työvuorosuunnittelu-ohjelmisto	ei erityisiä atk-taitoja	ohjelmiston hankkiminen suunnitelmissa, sen myötä mahdollisesti myös laitehankintoja ja koulutusta ko. ohjelman käyttöön
Yritys 10	vanhusten asumispalvelut	7	16	sähköposti, laskutus, asiakkaiden tiedot	esim. lääketieteellinen tiedonhaku	ei ohjelmistoa, käytetään tarpeita vastaavia lomakkeita sen sijaan	suurin osa henkilökunnasta hallitsee tarvittaessa atk-taitoja	laitehankintoja

	Palvelutyyppe	Henkilökunnan määrä	Asiakasmäärä	Tietotekniikan hyödyntäminen hallinnollisissa asioissa	Tietotekniikan hyödyntäminen hoitotyössä	Ohjelmistojen käyttö	Henkilökunnan arvioidut atk-taidot	Tulevaisuuden näkymät
Yritys 11	hoito- ja hoiva-ala	7	n. 150	normaalit toimistotyöt	asiakasrekisteri on "paperisessa muodossa"	ei	hyvät perustaidot	laitehankintoja ensi vuoden aikana
Yritys 12	sosiaalihuollon asumispalvelut	8	16	sähköposti, tekstinkäsittely (hoito- ja palvelusuunnitelmien laadinta), lomakkeet	lomakkeet	Stakesin hoitoilmoitus- ja asiakaslaskenta-ohjelma (HILMO)	nuoremmat rohkeampia ja taitavampia, vanhemmat eivät yhtä taitavia	lisää koulutusta ja siirtymistä entistä enemmän sähköiseen "toimintaan"
Yritys 13	psykoterapia	1	n.15	sähköposti, laskutus	asiakasrekisteri	ei, käyttää itse laadittuja lomakkeita	heikot	lisäkoulutusta
Yritys 14	vanhusten asumispalvelut	25	18	sähköposti, laskutus, maksuliikenne	asiakasrekisteri	hallinnon puolella käytössä ohjelmia	atk on lähinnä hallinnon työkalu, henkilökunta ei juurikaan käytä tietotekniikkaa työssään	ei erityisiä suunnitelmia

*) Ei osannut arvioida

Tasograafin duaalin etsintäalgoritmi

Harri Smått

Tiivistelmä

Tässä tutkielmassa esitellään duaaligraafin etsintäalgoritmiin ja graafin pintojen määrittämiseen liittyvää problematiikkaa, sekä lineaarisessa ajassa toimiva algoritmi 2-yhtenäisen tasograafin duaalin etsimiselle. Graafin duaaliin ja graafin pintoihin viitataan useissa algoritmeissa implisiittisesti, mutta näiden etsimistä varten koneellisesti ei tiettävästi ole esitelty algoritmia kirjallisuudessa. Tutkielmassa esitettävä algoritmi muodostaa duaaligraafin graafin pintojen etsimisen yhteydessä. Algoritmi on toteutettu risteämäärvon etsintäalgoritmia varten, mutta se on laajennettavissa yleisemmäksi duaalin määrittäväksi algoritmiksi. Vastaavasti pintojen hakeminen on irrotettavissa omaksi kokonaisuudekseen muita käyttötapoja varten.

Avainsanat: graafi, tasograafi, duaaligraafi, tasotestaus

CR-luokat: G.2.2, F.2.2

1. Johdanto

Duaaligraafia ja duaaligraafin määrittämiseen liittyviä graafin pintoja hyödynnetään useissa graafiteoreettisissa algoritmeissa, mutta kummankaan määrittämiselle ei ole esitetty koneellisesti suoritettavaa algoritmia kirjallisuudessa. Tässä tutkielmassa esitettävä algoritmi on toteutettu Projektityökurssin yhteydessä graafin risteämäärvon minimointia [Gutwenger *et al.*, 2001; Liebers, 2001] varten. Toteutettua risteämäärvon minimointia varten tulee graafille muodostaa duaali, ja duaalin solmuista tulee saada haettua tietoa sen edustamista alkuperäisen graafin pinnan muodostavista solmuista ja kaarista. Näillä tiedoilla on mahdollista etsiä lyhyin polku annetun graafin duaaligraafin solmujen välillä esimerkiksi risteämäärvon minimoinnin yhteydessä. Lisäksi muodostettavasta duaaligraafista on löydettävissä alkuperäisen graafin kaari, jonka duaaligraafin kaari ylittää, etsimällä duaaligraafin vierekkäisten solmujen edustamien pintojen yhteinen kaari. Esitettävä algoritmi etsii graafin pinnat annetulle tasoesitykselle, ja muodostaa graafin duaalin sen pintojen etsimisen yhteydessä.

Projektityökurssin toteutus piti toteuttaa avoimeen lähdekoodiin perustuvien ohjelmien, eikä missään löydytyissä ollut duaalin etsintää toteutettuna

valmiina. Kaupallisessa LEDA-graafikirjastossa on duaalin etsintä toteutettu, ja manuaalisissa sen kerrotaan toimivan lineaarisessa ajassa. Tarkempaa kuvausta algoritmin toiminnasta ei anneta, mutta tämän pohjalta projektityön yhteydessä pyrittiin löytämään kuin myös lineaarisessa ajassa toimiva duaalin etsintä GTL-graafikirjaston vanhahkoa vapaasti levitettävää versiota käyttäen.

Duaalin etsintä on jaettu osiin siten, että 2-yhtenäisen graafin duaalin määrittäminen olisi mahdollisimman suoraviivainen operaatio syvyysuuntaisen haun yhteydessä. Syvyysuuntainen haku ja sen tuottama informaatio graafista mahdollistaa monien graafeihin liittyvien ongelmien ratkaisemisen lineaarisessa ajassa [Tarjan, 1972]. Tutkielmassa esitettävä algoritmi pyrkii tunnistamaan graafin pinnat syvyysuuntaisen haun aikana löytyvien takautuvien kaarien avulla. Kun pinnat on tunnistettu oikein, duaalin muodostamiseen liittyvien erikoistapauksien käsittely on mahdollista suorittaa haun aikana. Syvyysuuntainen haku käy kaikissa graafin solmuissa ja mahdollistaa kaikkien solmuista lähtevien kaarien käsittelyn tarvittaessa. Duaaligraafin etsintäalgoritmi on mahdollista toteuttaa lineaarisessa ajassa toimivana, kun syötegraafi on 2-yhtenäinen.

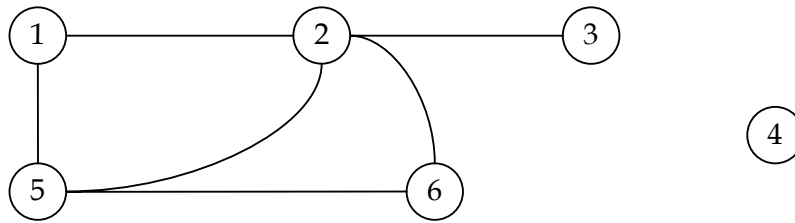
Esitettävä duaalin etsintäalgoritmi hyödyntää tasotestauksen yhteydessä määritettäviä tasoesityksen järjestettyjä vieruslistoja. Tämä sitoo algoritmin tasotestausalgoritmeihin, joista nykyään yleisimmin käytössä olevat toimivat lineaarisessa ajassa.

Tutkielman toisessa luvussa esitellään graafeihin ja duaalin etsimiseen liittyviä peruskäsitteitä, ja kolmannessa luvussa esitellään varsinainen algoritmi duaalin määrittämiseksi graafille.

2. Peruskäsitteitä

2.1. Graafi

Tutkielmassa käytetään Diestelin [2005] merkintöjä graafeille. Graafi $G = (V, E)$ on solmujen V ja niitä yhdistävien kaarien E joukko, missä $E \subseteq V^2$. Kaarien joukko on siis joukko solmupareja solmujen joukosta V . Tässä tutkielmassa käsitellään ainoastaan suuntaamattomia graafeja, eli jos $v_1, v_2 \in V$ ja $e_1 = \{v_1, v_2\} \in E$, niin myös $e_2 = \{v_2, v_1\} \in E$, ja toinen suunta jätetään merkitsemättä ja $e_1 = e_2$. Tutkielmassa ei myöskään käsitellä luppeja, eli jos $v_1, v_2 \in V$ ja $\{v_1, v_2\} \in E$, niin $v_1 \neq v_2$.



Kuva 2.1: Esimerkki graafista.

Kuvassa 2.1 on esitetty esimerkki suuntaamattomasta graafista. Kuvan graafi $G = (V, E)$ koostuu numeroiduista solmuista $V = \{1, 2, 3, 4, 5, 6\}$, sekä solmuja yhdistävien kaarien joukosta $E = \{\{1, 2\}, \{1, 5\}, \{2, 3\}, \{2, 5\}, \{2, 6\}\}$.

Graafin solmujen lukumäärää merkitään $|V|$ ja kaarien lukumäärää vastaavasti $|E|$, kuvan 2.1 graafissa $|V| = |E| = 6$.

Graafin solmuille merkitään lisäksi solmuun liittyvien kaarien joukko $E(v) = \{(v, x) \in E \vee (x, v) \in E \mid x, v \in V\}$, sekä solmuista koostuva vieruslista $N(v) = \{x \in V \mid (x, v) \in E(v)\}$. Graafin solmuille määritellään lisäksi asteluku, solmuun liittyvien kaarien lukumääränä $d(v) = |E(v)|$.

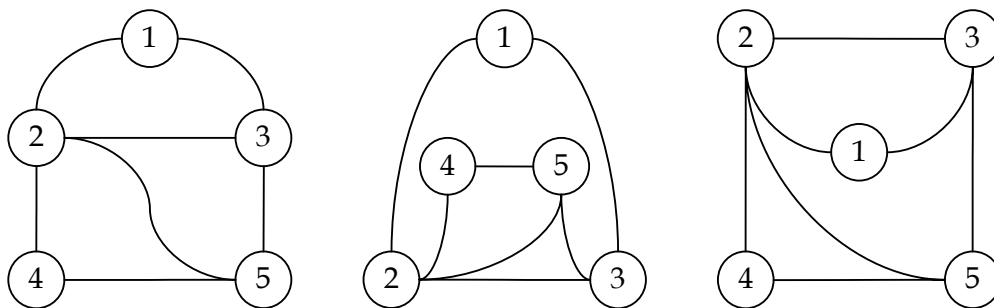
2.2. Tasograafi ja järjestetty vieruslista

Graafi on tasograafi, jos sen solmut voidaan sijoittaa tasoon siten, että kaaret eivät leikkaa toisiaan. Kun graafin solmut on sijoitettu tasoon, voidaan solmusta lähteviä kaaria lukea myötä- tai vastapäivään kiertäen – näin kiertämällä saatuja solmujen järjestettyjä vieruslistoja kutsutaan tasoesityksen järjestetyiksi vieruslistoiksi.

Shih ja Hsu [1999] esittävätkin tasotestauksen ongelmana löytää solmusta lähteville kaarille tasoesitystä vastaava järjestys; jos jokaiselle solmulle on olemassa sellainen järjestetty kaarien vieruslista, että graafi on piirrettävissä tasoon ilman että kaaret leikkaavat, on graafi tasograafi.

Graafin taso-ominaisuuden testaamiselle on esitetty kirjallisuudessa useita menetelmiä, joista nykyisin yleisesti käytettävät testausalgoritmit toimivat lineaarisessa ajassa. Ensimmäisen lineaarisessa ajassa toimivan tasotestauksen esittivät Hopcroft ja Tarjan [1974]. Vaikka heidän saavuttama aikavaatimus oli uraa uurtava, pidetään heidän tasotestaustaan hyvin hankalana toteuttaa ja käytännössä sen toteutus usein on hitaampi kuin uudemmat samassa asymp-tootisessa ajassa toimivat tasotestaukset. Tunnettuja lineaarisessa ajassa toimivia tasotestauksia ovat lisäksi Evenin ja Tarjanin [1976] sekä Boothin ja Luekerin [1976] esittelemät tasotestaukset.

Edellä mainitut tasotestaukset perustuvat monien mielestä hankalien tietorakenteiden hyödyntämiseen. Shih ja Hsu [1993, 1999] esittelivät ensimmäisen ainoastaan syvyysuuntaisen haun tuottamaa informaatiota hyödyntävän tasotestausalgoritmin, joka yksinkertaisti huomattavasti testausta ja tasograafin järjestettyjen vieruslistojen määrittämistä verrattuna aiemmin esitettyihin menetelmiin. Boyer ja Myrvold [1999, 2004] ovat esitelleet tällä hetkellä yksinkertaisimman ja helppotajuisimman tasotestauksen, joka hyödyntää syvyysuuntaisen haun ominaisuuksia vahvasti.



Kuva 2.2: Graafi piirretty tasoon.

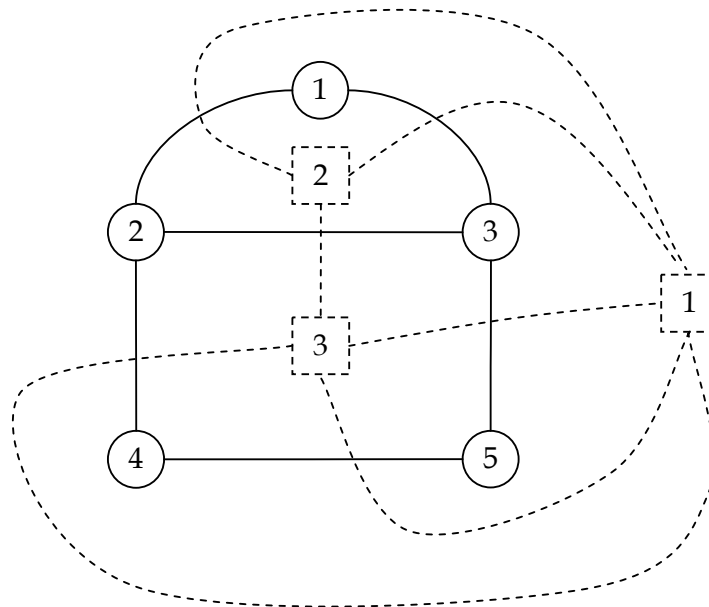
Kuvassa 2.2 on esitetty tasograafi piirrettynä kolmella eri solmujen sijoittelulla tasoon. Graafin tasoesitys ei ole yleisesti yksikäsitteinen, eivätkä vastaavasti graafin tasoesityksen järjestetyt vieruslistat ole yksikäsitteisiä. Esimerkiksi vasemman puoleisessa sijoittelussa solmun 2 järjestetty vieruslista myötäpäivään kiertäen on $\{1,3,5,4\}$, keskimmäisessä $\{1,4,5,3\}$ ja oikeanpuoleisessa sijoittelussa vieruslista on $\{3,1,5,4\}$.

Tasograafin solmujen järjestetyt vieruslistat määräävät graafin pinnat, jotka kuvataan joukkoina graafin solmuja tai vastaavasti kaaria. Graafin pintojen määrittäminen tapahtuu kiertämällä pinnan solmut kaaria pitkin myötä- tai vastapäivään. Valitaan ensiksi aloitussolmu v ja siitä lähtevä kaari e aloituskaareksi, sekä kaarien kiertosuunta. Tämän jälkeen toistetaan kunnes palataan takaisin aloitussolmuun; lisätään solmu v ja kaari e pinnan esitykseen, valitaan solmu v kaaren e vastakkaiseksi solmuksi ja valitaan kaari e uudessa solmussa v e :tä seuraavaksi kaareksi kiertäen valittuun kiertosuuntaan. Esimerkiksi kuvan 2.2 vasemmanpuoleisessa graafissa solmusta 5 kaarta $\{5,2\}$ pitkin aloittaen ja myötäpäivään kiertäen löydetään solmujen $\{5,2,4\}$ määräämä pinta. Kaarta $\{5,4\}$ pitkin aloittaen löydetään pinta $\{5,4,2,1,3\}$, joka edustaa graafin ulkotasoa kyseisellä tasoon sijoittelulla.

Tasoon piirretyn graafin pinnat nähdään helposti silmämääräisesti ja kirjallisuudessa pintoihin viitataan usein implisiittisesti ilman että tarkemmin määrätään kuinka kaikki pinnat löydetään graafista. Kolmannessa luvussa esitettävä algoritmi käy graafin solmut läpi siten, että pinnat, ulkotaso mukaankulkien, tunnustetaan oikein, eikä samaa pintaa kierretä kahta tai useampaa kertaa.

2.3. Duaaligraafi

Tasograafin G duaaligraafi D on graafi, joka saadaan asettamalla solmu kuvaamaan jokaista G :n pintaa, sekä yksi solmu kuvaamaan G :n ulkotasoa. Tämän jälkeen duaaliin lisätään kaari pintoja kuvaavien solmujen väliin jos pinnat ovat vierekkäiset, eli ne jakavat kaaren. Jos pinnoilla on useampi kuin yksi yhteinen kaari, lisätään duaaliin kaari kuvaamaan jokaista näistä.



Kuva 2.3: Duaaligraafin muodostaminen.

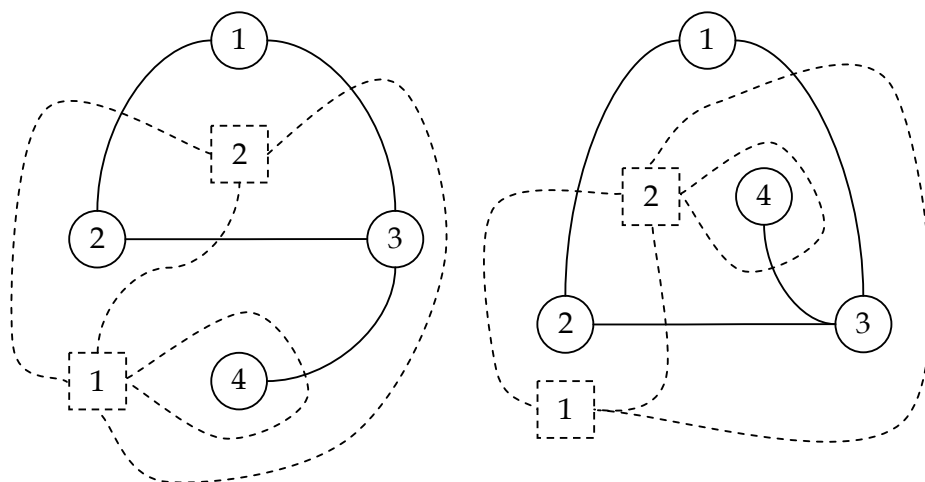
Kuvassa 2.3 on esitetty 2-yhtenäiselle graafille, ympyrät solmuina ja yhtenäiset viivat kaaria, muodostettu duaali. Kuvassa neliöt ovat duaaligraafin solmuja ja katkoviivat sen kaaria.

Graafin sanotaan olevan yhtenäinen, koska kaikkien sen solmuparien $v_1, v_2 \in V$ ($v_1 \neq v_2$) välillä on olemassa yhteys graafin kaaria pitkin. Graafi on 2-yhtenäinen, jos minkä tahansa kaaren $e \in E$ poistamisen jälkeen se on edelleen

yhtenäinen. Yleisemmin graafi on k -yhtenäinen, jos kaarien $e_1, e_2, \dots, e_{k-1} \in E$ poiston jälkeen graafi on yhtenäinen.

Kuvassa 2.3 nähdään lisäksi, että graafin duaalin duaali on alkuperäinen graafi; alkuperäisen graafin solmut 1,3,4 ja 5 ovat duaalin pinnoilla ja solmu 2 edustaa duaaligraafin duaalin ulkotasosolmua. Duaalin etsiminen ei kuitenkaan ole ongelmattonta. Esimerkiksi graafissa olevat alipuut tai puuttuva 2-yhtenäisyys aiheuttavat hankaluuksia duaalin määrittämisessä. Yleisesti edes 2-yhtenäisille graafeille ei ole löydettävissä yksikäsitteistä duaalia, kuten ei yksikäsitteisiä pintojakaan. Vasta 3-yhtenäinen graafi määrää yksikäsitteisen tasoesityksen ja siihen liittyvät pinnat [Servatius and Servatius, 1996].

Luvussa 3 esitettävä algoritmi ei huomioi useampia, mahdollisesti toisistaan poikkeavia, 2-yhtenäisten graafien tasoesityksiä. Useamman tasoesityksen huomioiminen vaatisi tasotestauksen muokkaamista, jotta kaikkia mahdollisia tasoesityksiä vastaavat järjestetyt vieruslistat voitaisiin käydä läpi. Tasotestausalgoritmit eivät itsessään mahdollista eri tasoesitysten hakemista, vaan nimensä mukaisesti tarkistavat onko graafi piirrettävissä tasoon ilman kaarien leikkaamisia jollain solmujen järjestetyillä vieruslistoilla.



Kuva 2.4: Duaali graafille jossa alipuu.

Kuvassa 2.4 on esimerkki alipuusta graafissa, kaari $\{3,4\}$, juurenaan solmu 3. Alipuut aiheuttavat ongelmia paitsi pintojen määrittämisessä, myös optimointiongelmissa joissa etsitään esimerkiksi lyhyintä polkua annetun graafin duaalin solmujen välillä. Kuvassa 2.4 kaaren $\{3,4\}$ sijainti järjestetyssä vieruslistassa ei vaikuta taso-ominaisuuteen, mutta saattaa muuttaa lyhyintä löytyvää polkua duaalin solmujen välillä. Duaalin lyhyimmän polun etsimisessä tuli-

sikin huomioida kaikki mahdolliset alipuiden sijoittamiset järjestetyissä vieruslistoissa. Tämä voidaan toteuttaa esimerkiksi lisäämällä kaikki alipuun kaaret jokaiseen pintaan mihin se on kierrettävissä. Alipuu aiheuttaa myös vastaavan päätösongelman duaalin määrittämisessä duaalille, jonka pitäisi olla alkupe-
räinen graafi, ja edellä kuvattu menetelmä puukaarien lisäämiseksi pintojen kaarilistoihin hukkaa palautusinformaation kokonaan.

Jos alkuperäinen graafi ei ole 2-yhtenäinen, voisi solmu 4 edustaa toista 2-yhtenäistä komponenttia, joka kokonaisuudessaan voi sijaita pinnan $\{1,2,3\}$ ulko- tai sisäpuolella. Vastaavasti pinta $\{1,2,3\}$ voisi sijaita toisen 2-yhtenäisen komponentin sisällä. Jokainen tapaus saattaa muuttaa duaalista löytyvän lyhyimmän polun pituutta. Tämä ei ole alipuiden tapaan triviaalisti ohitettavissa, koska molemmilla 2-yhtenäisillä komponenteilla on hyvin määritelty duaali. Pinnan sisään jäävää 2-yhtenäistä komponenttia ei voi lomittaa alipuiden tapaan ympäröivän pinnan osaksi ja eri sijoitteluvaihtoehdot on käsiteltävä muilla menetelmillä.

Seuraavassa luvussa esitettävässä algoritmossa ensiksi piilotetaan alipuut ja ne käsitellään erikseen pinnan kiertämisen yhteydessä. Algoritmi ei vaadi, että alipuiden poiston jälkeen graafin on oltava 2-yhtenäinen, mutta se toimii joissain tapauksissa väärin, jos näin ei ole. Alipuiden poiston jälkeen 2-yhtenäisille graafeille algoritmi löytää niiden tasoesitystä vastaavan duaaligraafin.

3. Duaalin etsintä

Graafin pintojen ja duaalin etsintä esitetään neljään funktioon jaettuna; alipuiden piilottaminen, haun alustus, syvyysuuntainen haku pintojen tunnistamiseksi ja löydetyn pinnan kiertäminen. Varsinainen duaaligraafin muodostaminen tapahtuu pinnan tunnistamisen yhteydessä ja sen kiertämisen yhteydessä yksinkertaisen tietorakenteen avulla. Etsintään kuuluvat algoritmit kuvataan pseudokoodina, joissa käytetään kuvassa 3.1 esiteltyjä luokkia ja niiden funktioita. Vastaavat funktiot ja luokat graafien käsittelyyn ovat käytössä LEDA- ja GTL-graafikirjastoissa.

```

/* Graafia kuvaava luokka */
class Graph

/* Graafi, jossa solmulle mahdollista listata alkuperäisen
   graafin solmuja ja kaaria (pinnat) */
class Dual : Graph

/* Tasotestaus ja siihen liittyvät vieruslistat */
class Planarity

    /* Tarkistaa onko G tasograafi ja jos on, muodostaa
       tasoesityksen järjestetyn vieruslistan */
    function IsPlanar(Graph G) : boolean

    /* Palauttaa solmun n kaarta e seuraavan kaaren
       järjestetyssä vieruslistassa (myötäpäivään) */
    function NextEdge(Node n, Edge e) : Edge

    /* Palauttaa solmun n kaarta e edellisen kaaren
       järjestetyssä vieruslistassa (vastapäivään) */
    function PrevEdge(Node n, Edge e) : Edge

class Node

    /* Palauttaa tämän solmun vastakkaisen solmun kaarta
       e pitkin */
    function Opposite(Edge e) : Node

    /* Palauttaa solmun yksilöivän id-numeron */
    function Id() : int

class Edge

    /* Palauttaa kaaren yksilöivän id-numeron */
    function Id() : int

```

Kuva 3.1: Pseudokoodissa käytettävät luokat.

3.1. Alipuiden poistaminen

Graafin solmu v on alipuun lehti jos ja vain jos sen aste $d(v) = 1$. Tämän perusteella on muodostettavissa algoritmi, joka poistaa tai piilottaa alipuiden kaaret graafista lehtisolmuista aloittaen.

Olkoon G yhtenäinen graafi, jossa vähintään yksi 2-yhtenäinen komponentti. Olkoon $v_1 \in G$ alipuun lehtisolmu, $e_1 = E(v_1)$ ja $v_2 = N(v_1)$. Koska solmu v_1 on lehtisolmu, on olemassa solmu $r \in G$, joka on kyseisen alipuun juurisolmu ja joka kuuluu graafin 2-yhtenäiseen osaan. Tästä seuraa, että $d(r) > 2$ ja $d(v_2) \geq 2$. Nyt poistamalla solmu v_1 ja kaari e_1 , solmun v_2 aste pienenee yhdellä. Jos poiston jälkeen $d(v_2) = 1$, on v_2 nyt lehtisolmu ja poistetaan kuten edellä. Jos solmun aste on poiston jälkeen suurempi kuin yksi, eikä kyseessä ole

alipuun juurisolmu, on solmun v_2 alla olevassa puussa vähintään yksi lehtisolmu v_3 . Poistetaan solmu v_3 kuten edellä, ja jos kaari $e_2 = \{v_2, v_3\} \in E(v_3)$, pienenee solmun v_2 aste yhdellä. Jos solmun v_2 aste on edellisen solmun poistamisen jälkeen yksi, poistetaan se, ja muussa tapauksessa sen alla on edelleen vähintään yksi lehtisolmu.

Poistamalla alipuun lehti ja sen isä jos isästä tulee poiston jälkeen lehtisolmu, saadaan alipuu poistettua lukuunottamatta juurisolmua, jonka aste on aina suurempi kuin yksi.

```
function HideSubTrees(Graph G)
  for all v in G
    while d(v) == 1
      Node parent = N(v)
      HideEdge(E(v))
      HideNode(v)
      v = parent
    end
  end
end
```

Kuva 3.2: Algoritmi alipuiden poistamiseksi graafista G.

Kuvassa 3.2 esitetyn algoritmin yhteydessä on mahdollista tallentaa poistetut kaaret ja solmut toiseen tietorakenteeseen myöhempää käsittelyä varten. Esimerkiksi GTL- ja LEDA-graafikirjastot mahdollistavat solmujen sekä kaarien piilottamisen, jolloin alipuiden piilottamisen jälkeen suoritettava tasotestaus tapahtuu graafin näkyville osille ja tasoesityksen järjestetyt vieruslistat eivät sisällä piilotettuja osia. Kun tasotestauksen jälkeen piilotetut osat palautetaan, tunnistetaan alipuiden juurisolmut siitä että niiden aste on suurempi kuin järjestetyn vieruslistan koko.

3.2. Syvyysuuntainen haku

Kuten luvussa 2 esitettiin, yksittäisen pinnan määrittäminen on mahdollista valitsemalla solmu, siitä lähtevä kaari ja kiertosuunta. Duaaligraafin muodostaminen vaatii jokaisen pinnan etsimistä, mutta pintoja ei voi tai ei kannata etsiä tutkimalla jokaisen solmun jokaista kaarta pitkin alkavaa pintaa. Näin sama pinta löydettäisiin jokaisesta pinnan solmusta alkaen useaan kertaan ja algoritmin tehtävänä olisi poistaa kahteen tai useampaan kertaan löydetty pinnat.

Lineaarisen aikavaatimuksen saavuttamiseksi algoritmi etsii pinnat syvyys-suuntaisen haun yhteydessä, tasoesityksen järjestettyjen vieruslistojen määrämässä järjestyksessä kaaria kiertäen. Erityisen oleellista pintojen tunnistamisen kannalta on, että graafissa on yhtä monta sisäpintaa kuin graafissa on syvyys-suuntaisen haun löytämiä takautuvia kaaria.

Edellä mainittuihin ominaisuuksiin perustuen on mahdollista löytää graafin pinnat syvyys-suuntaisen haun yhteydessä, kun haku suoritetaan kiertäen tasotestauksen määrämien järjestettyjen vieruslistojen mukaisesti myötä- tai vastapäivään. Tasotestaus, järjestettyjen vieruslistojen määrittäminen, on suoritettavissa lineaarisessa ajassa ja lisäksi syvyys-suuntaisen haku pintojen määrittämiseksi voidaan suorittaa lineaarisessa ajassa.

Aluksi piilotetaan graafista alipuut, tämän jälkeen suoritetaan tasotestaus jonka yhteydessä määritetään jäljelle jäävän 2-yhtenäisen aligraafin järjestetyt vieruslistat. Tasotestauksen jälkeen valitaan satunnainen solmu aloitussolmuksi ja satunnainen siitä lähtevä kaari aloituskaareksi syvyys-suuntaista hakua varten. Alustetaan tyhjä duaaligraafi ja lisätään siihen yksi solmu ulkotasoa kuvaamaan. Tämän jälkeen palautetaan piilotetut osat graafiin ja aloitetaan syvyys-suuntaisen haku.

```
function FindDualGraph(graph G) : Dual D
HideSubTrees(G)
Planarity P
if P.IsPlanar(G)
    Node outernode = D.NewNode()
    Node v = RandomNode(G)
    Edge e = random edge from E(v)
    Dual D = new Dual()
    G.Restore()
    int dfi[]
    FindFacesDFS(D, P, outernode, v, e, dfi, 1)
    return D
else
    error, graph not planar
end
```

Kuva 3.3: Pintojen ja duaaligraafin haun alustus.

Kuvan 3.3 haun alustuksessa valittu satunnainen aloitussolmu ja siitä lähtevä satunnainen kaari kuuluvat algoritmin suorituksen aikana graafin ulkotasoon. Kaarien kiertäminen vastakkaisesti suuntiin pinnan haun ja syvyys-suuntaisen haun aikana johtavat siihen, että graafin ulkotaso esitetyssä algoritmissa määräytyy aloitussolmuksi valitun solmun ja siitä lähtevän kaaren määrämänä pintana. Ulkotasoksi jää pinta, joka saadaan kiertämällä samaan suuntaan kuin

syvyysuuntainen haku suoritetaan, tai vastaavasti eri suuntaan kuin pinta etsitään syvyysuuntaisen haun yhteydessä.

3.3. Pintojen haku ja tallentaminen

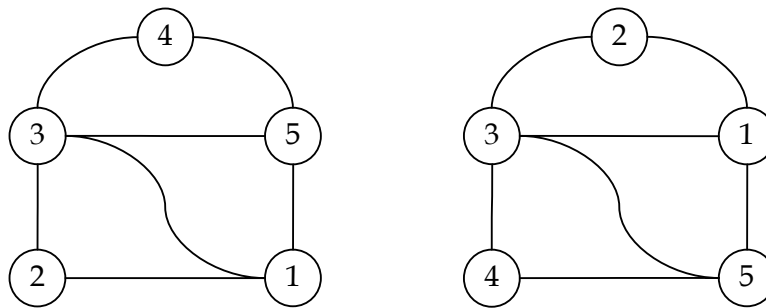
Syvyysuuntainen haku esitetään kiertämällä solmun kaaria toistuvasti myötäpäivään haun edetessä. Jokaisella rekursiotasolla solmulle asetetaan dfi-arvo, joka kuvaa syvyysuuntaisen haun löytämisjärjestystä. Aluksi dfi-taulukko on alustettu nolllaksi jokaiselle solmulle, minkä avulla tunnistetaan läpikäymättömät solmut ja jatketaan rekursiota tällaisen tullessa vastaan. Muut vaihtoehdot ovat, että seuraava solmu on takautuvan tai etenevän kaaren päässä. Näistä olemme kiinnostuneita takautuvien kaarien päässä olevista solmuista, joilla dfi-arvo on pienempi kuin sillä hetkellä käsiteltävänä olevan solmun dfi-arvo. Jos kyseessä on takautuva kaari, etsitään käsiteltävänä olevasta solmusta n lähtevän kaaren `child_e` määräämä taso. Kuvan 3.4 pseudokoodissa takautuva kaari on määritelty ehdolla $dfi < d-1$, joka estää tulokaaren e uudelleen valinnan. Tämä ehto estää samalla kaksinkertaisten kaarien tunnistamisen solmujen välillä.

```
function FindFacesDFS(Dual D, Planarity P, Node n, Edge e,
                    int dfi[], int d)

dfi[n] = d
for all child_e in P(n) in clockwise order
  Node child_n = opposite node to n over child_e
  if dfi[child_n] == 0
    /* Läpikäymätön solmu */
    FindFacesDFS(D, P, child_n, child_e, dfi, d+1)
  else dfi[child_n] < d-1
    /* Takautuva kaari */
    FindFace(D, P, n, child_e)
  end
end
```

Kuva 3.4: Syvyysuuntainen haku pintojen tunnistamiseksi.

Pinnan tunnistaminen ja lisääminen duaaliin suoritetaan kiertämällä kaaria vastakkaiseen suuntaan kuin syvyysuuntainen haku. Tämä ominaisuus on keksitty puolivahingossa, eikä eksplisiittistä todistusta algoritmin oikeellisuudelle ole olemassa. Empiirisillä kokeilla on kuitenkin saatu vahvistusta sille, että pinnat tunnistetaan oikein näin meneteltäessä. Sen sijaan syvyysuuntaisen haun ja pinnan määrittämisen suorittaminen kiertämällä solmujen järjestettyjä vieruslistoja molemmissa samaan suuntaan on osoitettavissa vääräksi vastaesimerkillä.



Kuva 3.5: Virhetilanteet samaan suuntaan pinnan etsimisessä.

Kuvassa 3.5 on esitetty kaksi graafia, joissa solmujen numerointi on sama kuin syvyysuuntaisen haun dfi-arvot.

Vasemmanpuoleisessa graafissa haku ja pinnan etsiminen suoritetaan myötapäivään kiertäen, aloittaen solmusta 1 kaarta $\{1,2\}$ pitkin. Rekursio kiertäisi jokaisessa solmussa myötapäivään kiertäen solmuun 5, jossa kaari $\{5,1\}$ on takautuva kaari. Pinnan haku myötapäivään kiertäen solmusta 5 kaarta $\{5,1\}$ pitkin aloittaen löytää pinnan $\{5,1,2,3,4\}$. Tämän jälkeen solmusta 5 löytyy toinen takautuva kaari $\{5,3\}$, joka määrää pinnan $\{5,3,1\}$. Rekursio palaa solmuun 3, josta löytyy takautuva kaari $\{3,1\}$, joka määrää pinnan $\{3,1,5\}$.

Oikeanpuoleisessa graafissa haku ja pinnan etsiminen suoritetaan vastapäivään kiertäen. Solmusta 1 päädytään rekursiossa solmuun 5, josta löytyy takautuva kaari $\{5,1\}$ ja sen määräämä pinta $\{5,1,2,3,4\}$. Solmusta 5 löytyy lisäksi takautuva kaari $\{5,3\}$ ja sen määräämä pinta $\{5,3,1\}$. Rekursio palaa solmuun 3, josta löytyy takautuva kaari $\{3,1\}$ ja sen määräämä pinta $\{3,1,5\}$.

Molemmissa tapauksissa solmujen 1, 3 ja 5 määräämä pinta löydetään kaksi kertaa. Duaalin etsimisessä tulisi löytää jokainen pinta tasan kerran. Koska aloitusolmu on satunnainen, ei voida olettaa että pinnat löydetään oikein samaan suuntaan kiertäen, vaikka monessa tapauksessa näin tapahtuukin.

Jos edellisissä esimerkeissä pinta haetaan vastakkaiseen suuntaan kuin syvyysuuntainen haku suoritetaan, löydetään molemmissa tapauksissa pinnat $\{5,1,3\}$, $\{5,3,4\}$ ja $\{3,1,2\}$. Tämän huomion perusteella suoritetaan pinnan määrittäminen kiertämällä vastakkaiseen suuntaan kuin syvyysuuntainen haku etenee.

```

function FindFace(Dual D, Planarity P, Node n, Edge e)

Node dual_node = D.NewNode()
Node face_n = n
Edge face_e = e

do
  D.AddFaceNode(dual_node, face_n)
  D.AddFaceEdge(dual_node, face_e)

  if solmun face_n aste != face_n:n järj.vieruslistan koko
    for all tree_edge joka ei kuulu järj.vieruslistaan
      Lisää face_n:stä tree_edge pitkin alkava
      alipuu dual_node:n kaarihin ja solmuihin
    end
  end

  if ensimmäinen kerta kun face_e lisätty D:hen
    D.NewEdge(dual_node, D.OuterNode())

  else if toinen kerta kun face_e lisätty D:hen
    siirrä duaalin kaari ulkotaso solmusta dual_node:en

  else
    error, jokainen kaari lisätään max. 2 kertaa
  end

  face_n = prev_n.Opposite(face_e)
  face_e = P.PrevEdge(face_n, face_e)
while face_n != n

```

Kuva 3.6: Pinnan kaarien kiertämisalgoritmi.

Kuvan 3.6 algoritmi etsii pinnan vastapäivään kiertäen. Tässä vaiheessa on pinta tunnistettu oikein ja duaaliin lisätään tasoa kuvaava duaalin solmu. Tälle duaalin solmulle lisätään tieto sen edustamista alkuperäisen graafin pinnan solmuista ja kaarista kiertämällä löydetty graafin pinta vastapäivään.

Jos pinnan kierämisen aikana törmätään solmuun, jonka järjestetyn vieruslistan koko on erisuuri kuin sen aste, tiedetään että kyseessä on vähintään yhden aiemmin piilotetun alipuun juurisolmu. Etsitään kaikki solmusta lähtevät kaaret jotka eivät kuulu järjestettyyn vieruslistaan, ja lisätään niiden alle jäävät alipuut duaalin solmun pinnan esitykseen.

Lisättävien duaalin solmujen määrä on sama kuin graafin pintojen lukumäärä, mutta vielä tulisi varmistua, että ensimmäisellä kerralla ulkotasoa kuvaavaan solmuun lisätyt kaaret siirretään viereisiä pintoja kuvaaviin duaalin solmuihin aina, kun kaksi pintaa jakavat kaaren. Tähän soveltuu yksinkertainen taulukko, johon kirjataan luodut duaalin kaaret käyttäen alkuperäisen graafin kaarien id-arvoa. Samaan taulukkoon voidaan kirjata myös graafin kaarien käyttökerrat.

```
if map[face_e.Id()].count == 0
    map[face_e.Id()].count = 1
    map[face_e.Id()].dual_e = D.NewEdge(dual_n, outer_n)
else if map[face_e.Id()].count == 1
    map[face_e.Id()].count = 2
    Edge dual_e = map[face_e.Id()].dual_e
    vaihda dual_e:ssä outer_n:n tilalle dual_n
else
    virhe
```

Kuva 3.7: Kaarien käyttökertojen ja duaalin kaarien taulukoiminen.

Kuvassa 3.7 esitetyn taulukoinnin avulla pinnan kaarten yli lisätään duaaliin ensiksi kaari ulkotasoja kuvaavaan duaalin solmuun. Kun kaari kierretään seuraavan pinnan yhteydessä, siirretään sen ylittänyt duaalin kaari ulkotasoja kuvaavasta solmusta nyt löydettyyn uutta pintaa kuvaavaan duaalin solmuun.

4. Yhteenveto

Edellä esitelty algoritmi on toteutettu osana projektityökurssia UTAG-graafikirjastossa (DualGraphAlgorithm). Algoritmi on suunniteltu ainoastaan duaalin avulla suoritettavaan risteämäärvon määrittämiseen. Muodostettavan duaaligraafin ei ole oletettu olevan oikeaoppinen duaali, vaan on keskitytty ainoastaan tarvittavaan duaaliin lyhyimmän polun löytämiseksi siitä, ja duaalin solmuihin liittyvien alkuperäisen graafin pintojen solmu- ja kaarilistojen muodostamiseen. Tätä toteutusta on käytetty risteämäärvon etsimiseen annetulle graafille.

Algoritmi ei toimi kaikissa tapauksissa oikein, esimerkiksi silloin, kun graafissa jakautuu useampaan kuin yhteen 2-yhtenäiseen komponenttiin. Tällaisessa tilanteessa graafi on mahdollista täydentää 2-yhtenäiseksi lineaarisessa ajassa. Tätä lähestymistapaa voisi hyödyntää myös duaalin etsimisessä, mutta tällöin esimerkiksi alipuut tulevat sidotuksi tietyn pinnan sisään, eikä lyhyintä mahdollista olemassa olevaa polkua välttämättä löydetä muodostettavasta duaalista. Toisaalta 2-yhtenäisten komponenttien sitominen tietyn pinnan sisään korjaa algoritmista olevat puutteet tekemällä siitä 2-yhtenäisen.

Jatkotutkimuksena voisi algoritmia pyrkiä parantamaan yleisemmän duaalin määrittämiseksi esimerkiksi yhtenäisillä graafeilla, joissa on useampi kuin yksi 2-yhtenäinen komponentti. Toisaalta ainakin Boyerin ja Myrvoldin [2004] esittämän tasotestauksen yhteydessä muodostuvat solmujen järjestetyt vieruslistat sisäiseen tietorakenteeseen. Algoritmin lopussa on mahdollista kiertää solmusta lähtevät kaaret tasoesityksen määräämässä järjestyksessä muodostetun tietorakenteen avulla, tämän lisäksi on mahdollista kiertää solmusta

lähteivät pinnat saman tietorakenteen avulla. Optimaalinen tapa muodostaa graafin duaali nyky menetelmin saattaisi löytyä, jos duaalin muodostaisi esimerkiksi kyseisen tasotestauksen yhteydessä.

Viitteluettelo

- [Booth and Lueker, 1976] Kellogg S. Booth and George S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and Systems Sciences* **13** (1976), 335–379.
- [Boyer and Myrvold, 1999] John M. Boyer and Wendy J. Myrvold, Stop minding your P's and Q's: A simplified $O(n)$ planar embedding algorithm. In: *Proc. of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, 140–146.
- [Boyer and Myrvold, 2004] John M. Boyer and Wendy J. Myrvold, On the cutting edge: Simplified $O(n)$ planarity by edge addition. *Journal of Graph Algorithms and Applications* **8** (2004), 241–273.
- [Diestel, 2005] Reinhard Diestel, *Graph Theory, Graduate Texts in Mathematics* 173. Springer-Verlag, New York, 2005.
- [Even and Tarjan, 1976] Shimon Even and Robert. E. Tarjan. Computing an st-numbering. *Theoretical Computer Science* **2** (1976), 339–344.
- [Gutwenger *et al.*, 2001] Carsten Gutwenger, Petra Mutzel and René Weiskircher, Inserting a edge into a planar graph. In: *Proc. of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, 246–255.
- [Hopcroft and Tarjan, 1974] John Hopcroft and Robert Tarjan, Efficient planarity testing. *Journal of the ACM* **21** (1974), 549–568.
- [Liebers, 2001] Annegret Liebers, Planarizing graphs – a survey and annotated bibliography. *Journal of Graph Algorithms and Applications* **5** (2001), 1–74.
- [Servatius and Servatius, 1996] Brigitte Servatius and Herman Servatius, Self-dual graphs, *Discrete Mathematics* **149** (1996), 223–232.
- [Shih and Hsu, 1993] Wei-Kuan Shih and Wen-Lian Hsu, A simple test for planar graphs. In: *Proc. of the International Workshop on Discrete Mathematics and Algorithms*, 110–122.
- [Shih and Hsu, 1999] Wei-Kuan Shih and Wen-Lian Hsu, A new planarity test. *Theoretical Computer Science* **233** (1999), 179–191.
- [Tarjan, 1972] Robert E. Tarjan, Depth-first search and linear graph algorithms. *SIAM Journal of Computing* **1** (1972), 146–160.

Ohjelmistopatenteista

Sakari Tamminen

Tiivistelmä

Tämä tutkielma on kirjallisuuskatsaus tietokoneohjelmien patentoimiseen liittyvistä kysymyksistä. Tutkielma sisältää perustietoa patenteista ja patenttijärjestelmästä sekä siihen liittyvistä muista ilmiöistä ja pyrkii tuomaan esille, mitä vaikutuksia ohjelmistopatenteilla voisi olla ja on alan teollisuudelle Euroopassa ja Yhdysvalloissa.

Avainsanat ja -sanonnat: ohjelmistopatentit, immateriaalioikeudet

CR-luokat: D.2, D.m, H.4, K.1, K.4, K.5

1. Johdanto

Nopeasti kasvanut ohjelmistoteollisuus ei ole perinteisesti voinut suojata keksintöjään millään, ja muiden ideoiden kopiointi onkin ollut sääntö eikä poikkeus. Erilaisten immateriaalioikeuksien voimistuessa myös ohjelmistoteollisuus on havahtunut vaatimaan tekijänoikeutta parempaa suojaa tuotteilleen. Tietokoneohjelmien patentointi on useiden tahojen mielestä tarpeellista, toisten väittäessä, että patentit ja tietokoneohjelmat eivät sovi yhteen. Ohjelmistojen patentointi on lisääntynyt voimakkaasti Yhdysvalloissa viimeisen kymmenen vuoden aikana, ja myös Euroopassa on pohdittu niiden käyttöönottoa. Tämän tutkielman tarkoituksena on esitellä perusteluja ohjelmistojen patentoitavuuden puolesta ja sitä vastaan, ja antaa lukijalle kokonaiskuva siitä, mitä kaikkea ohjelmistopatentteihin ja ylipäätään patentointiin liittyy ohjelmistoteollisuudessa.

2. Informaatio tuotteena

Tietämykseen perustuvassa taloudessa informaatiosta on tullut erittäin tärkeä resurssi, joka mahdollistaa sekä yritysten että valtioiden kasvun. Sanonta *”tieto on valtaa”* pitää todellakin paikkansa. Joskus kuulee tätä vielä jatkettavan *”...ja salatieto ylivaltaa”*. Patentit eivät ole salatietoa, mutta ylivallan ne voivat antaa.

Informaatio tarvitsee tärkeytensä vuoksi suojelua. Coriat ja Orsi [2002] selittävät, kuinka informaatio tuotteena aiheuttaa ongelmia sen tuottamiseen käytetyn pääoman suojelussa. Ongelman ydin piilee siinä, että kun informaatio on kerran tuotettu, sitä voidaan monistaa loputtomasti käytännössä ilmaiseksi. Varianin [1998] mukaan informaatiotuotteisiin liittyy vielä kaksi muutakin ongelmallista ominaisuutta: tuote täytyy *”kokea”* ennen kuin voi tietää,

haluaako sen, ja tuote on ”julkista hyvää” eli sen kuluttaminen ei ole keneltäkään pois, eikä kukaan voi omalla kulutuksellaan estää muita käyttämästä sitä.

Tekniset innovaatiot ovat nekin tämän kaltaista informaatiota. Kun se on kerran omaksuttu, sitä ei enää saada takaisin. Jos tämän informaation valmistaminen on maksanut jollekulle paljon, kyseinen taho ei yleensä halua jakaa sitä ilmaiseksi. Ongelman kiertämiseksi esitetään yleensä kahta vaihtoehtoista tapaa: patentointi ja yhteiskunnan tuki. Yhteiskunnan tuella tarkoitetaan sitä, että tutkimus maksetaan verorahoista ja tuloksia jaetaan sitten ilmaiseksi. Näiden kahden tavan välinen ero on siinä kuka informaation tuottamisen lopulta maksaa: patenttien kohdalla kuluttaja ja tuetun tutkimuksen kohdalla veronmaksaja. Normaalisti nämä kaksi järjestelmää toimivat käsi kädessä siten, että julkisesti rahoitettu tutkimus luo pohjaa yksityiselle, yleensä soveltavalle tutkimukselle.

Yhdysvalloissa tilanne alkoi muuttua 80-luvun alussa, kun tiedostettiin, että myös ulkomaalaiset yritykset hyötyvät amerikkalaisten veronmaksajien rahoittamasta tutkimuksesta ja kilpailevat näihin tutkimuksiin perustuvilla tuotteilla Yhdysvaltojen markkinoilla. Ärtymys, jonka tämä ”vapaa-matkustajien” huomaaminen aiheutti, on johtanut immateriaalioikeuksien huomattavasti tehokkaampaan suojelemiseen. [Coriat and Orsi, 2002]

3. Tietokoneohjelmat ja immateriaalioikeudet

Tämä luku esittelee tietokoneohjelmien suhdetta erilaisiin immateriaalioikeuksiin, kuten tekijänoikeuteen ja erityisesti patentteihin. Lisäksi tarkastelun kohteena on muutamia muita patentteihin ja immateriaalioikeuksiin liittyviä ilmiöitä kuten liiketoimintamenetelmäpatentit, strateginen patentointi, ristiinliensointi, patentti-pooli, liikesalaisuudet ja immateriaalioikeudet päätoimialana.

Patentin ja tekijänoikeuden ero on siinä, että tekijänoikeuden piiriin kuuluvat esimerkiksi kirjalliset teokset ja erilaiset nauhoitteet sekä kuvataide, patenttien puolestaan suojellessa keksintöjä, kuten moottoreita ja erilaisia koneita sekä niiden osia. Voidaan sanoa, että patentti suojelee ideaa, tekijänoikeus suojelee idean esittämistä. [Oz, 1998]

3.1. Tekijänoikeus

Tekijänoikeus on yleisesti käytetty ja usein myös ainoa mahdollinen tapa suojella tietokoneohjelmia. Tekijänoikeus ei kuitenkaan kaikkien mielestä ole riittävä suoja ohjelmistoinnovaatioille, koska tekijänoikeutta luotaessa ei ole voitu ottaa huomioon tietokoneohjelmien erityispiirteitä.

Tietokoneohjelmat koostuvat aina kahdesta osasta, sekä ideasta/logiikasta että lähdekoodista, jolla tänä toiminnallisuus on ilmaistu. Koska tekijänoikeutta ei ole tarkoitettu suojelemaan "hyödyllisiä objekteja", tietokoneohjelmat sopivat sen piiriin vain osittain. Tietokoneohjelmat muodostavatkin eräänlaisen lainopillisen paradoksin olemalla luovia teoksia, jotka ovat hyödyllisiä.

Tietokoneohjelmat tarvitsevat tekijänoikeudellisen suojan, koska ne ovat myös kirjallisia teoksia. Tekijänoikeus onkin osoittautunut hyväksi keinoksi esimerkiksi piratismia vastaan [Cook, 2002]. Ozin [1998] mukaan tekijänoikeuden tarvitsee suojella sekä ohjelman lähdekoodia että sen objektikoodia, jotta ohjelma olisi suojattu. Nykyisen käytännön mukaan näin onkin. Tietokoneohjelmien erikoista luonnetta kuvastaa hyvin se, että jos ohjelman lähdekoodi järjestellään uudelleen, kyseessä on uusi teos, mutta toiminnallisuus ei välttämättä ole muuttunut [Randall et al., 1996].

Kaikki tietokoneohjelmat eivät ole yhdenvertaisia toiminnallisuuden suhteen. Randall ja muut [1996] päätyivät kyselytutkimuksessaan erikoiseen lopputulokseen: ohjelmistoyrittäjien ja asianajajien mielestä pelit muodostavat poikkeuksen, eivätkä ne tarvitse yhtä voimakasta suojelua kuin hyötyohjelmat. Tietokonepelit ovat kuitenkin hyvinkin tuottoisa business, joten niistä on jollekulle hyötyä vähintäänkin taloudellisessa mielessä. Tämä jättää auki kysymyksen siitä, pitäisikö myös tietokonepelien kuulua suojattuihin innovaatioihin, ja jos pitäisi, niin millä tasolla.

3.2. Patentti

Suomen Patentti- ja rekisterihallitus määrittelee patentin seuraavalla tavalla: *"Patentti on yhteiskunnan keksijälle myöntämä yksinoikeus, jonka vastineeksi keksijän on sallittava keksintönsä julkaiseminen. Patentin voi saada teollisesti käyttökelpoiseen keksintöön, joka on uusi ja eroaa olennaisesti aikaisemmista"* [PRH, 2005]. Käytännössä patentilla suodaan patentinhaltijalle laillinen monopoli tietyn keksinnön kaupalliseen hyödyntämiseen. Yksityistä käyttöä patenteilla ei voida estää.

Patenttijärjestelmän tarkoituksena on suojella kehitys- ja tutkimustyötä ja täten motivoida yhteiskunnan jäseniä keksimään uusia innovatiivisia tuotteita ja menetelmiä. Patenttijärjestelmä on yhteiskunnalle hyödyllinen, jos patenteista saatava hyöty, uudet keksinnöt, on korkeampi kuin yhteiskunnan niistä maksama hinta, monopoli, joka nostaa kuluttajien maksamaa hintaa ja luo esteitä muille innovaattoreille alalla [Bessen and Hunt, 2003; Henkel, 1992; Pollock et al., 2004]. Patenttien hyödyllisyyden arviointi onkin näiden kolmen vaikutuksen keskinäisten suhteiden arviointia.

Yleisesti Euroopassa patentti voidaan myöntää vain luonteeltaan tekniseen keksintöön [Beresford, 2001]. Suomalainen patenttilaki määrittelee asian näin: *"Keksinnöksi ei katsota pelkästään: 1) löytöä, tieteellistä teoriaa tai matemaattista*

menetelmää; 2) taiteellista luomusta; 3) suunnitelmaa, sääntöä tai menetelmää älyllistä toimintaa, peliä tai liiketoimintaa varten tai tietokoneohjelmaa; eikä 4) tietojen esittämistä.” [PL 1 §] Nämä määritelmät ovat suunnilleen samat koko Euroopan alueella. Myöskään eläin- tai kasvilajeja, ihmisen geenin sekvenssejä, ihmiskehon osia, ihmisen kloonausmenetelmiä tai vastaavia keksintöjä, joiden kaupallinen hyödyntäminen olisi yleisen järjestyksen tai hyvien tapojen vastaista, ei voida patentoida.

3.3. Nykyinen tilanne

Perinteisesti tietokoneohjelmat tai niihin liittyvät innovaatiot eivät ole olleet patentoitavia Euroopassa, mutta erityisesti isot yritykset ovat ajaneet tähän asiaan muutosta. Tietyissä Euroopan maissa ohjelmistopatenttien saaminen on mahdollista, mutta esimerkiksi EU ei vaadi sitä jäsenmailtaan. EU kumosikin ohjelmistojen patentoinnin mahdollistavan lakiesityksen vuoden 2005 kesäkuussa, mutta vastaavanlainen esitys tehtäen muutaman vuoden kuluttua, kun se on uudestaan mahdollista. Esityksen tarpeellisuutta on perusteltu tarpeella harmonisoida eri maiden sekä Euroopan ja Yhdysvaltojen välinen patenttikäytäntö.

Yhdysvaltalainen patenttikäytäntö poikkeaa eurooppalaisesta siten, että sen mukaan myös ohjelmistojen, löytöjen ja liiketoimintamallien patentoiminen on mahdollista [Coriat and Orsi, 2002]. Japanilainen tulkinta ohjelmistojen patentoivuudesta lankeaa Euroopan ja Yhdysvaltojen väliin [McQueen, 2004].

EPO¹ salli pelkkiin tietokoneohjelmiin kohdistuvat patenttivaatimukset vuonna 1999, vaikka ne eivät olekaan lainvoimaisia koko Euroopan alueella [Beresford, 2001]. Ennen vuotta 1999 vaatimuksena oli jonkin fyysisen laitteen liittyminen keksintöön, mutta tietyssä mielessä ohjelmistopatenttien saaminen oli mahdollista [Olsson and McQueen, 2000].

Tähän mennessä Yhdysvalloissa on myönnetty yli 167 000 ohjelmistopatenttia tai ohjelmistoihin liittyvää patenttia [Bessen and Hunt, 2003]. Euroopassa EPO on myöntänyt ainakin 15 000 vastaavaa patenttia, ehkä jopa 60 000 [Beresford, 2001; Chabchoub and Niosi, 2004]. Japanin luvut eivät ilmeisesti ole yhtä isoja, mutta ainakin vuoteen 1999 saakka kasvu on ollut huomattavasti nopeampaa kuin Yhdysvalloissa ja Euroopassa [McQueen, 2004]. Selviä lukuja ohjelmistopatenttien määrästä on vaikea saada selville, koska patenttijärjestelmissä ei ole omaa luokitusta niille ja koska kaikkia ohjelmistopatentteja ei ole kirjoitettu sellaisina, vaan osa ”naamioitu” joksikin muuksi [Chabchoub and Niosi, 2004; Bessen and Hunt, 2003].

¹ European Patent Office

Yleisesti voimakkaat immateriaalioikeudet, varsinkin patentit, ovat pitkälti kehittyneiden maiden käsissä ja näiden oikeuksien kansainvälinen harmonisointi on omiaan lisäämään kehittyneiden maiden valta-asemia. Euroopassa osa maista ja yrityksistä on kokenut harmonisoinnin uhkana, koska yhdysvaltalaiset yritykset voisivat sen turvin siirtää omat patenttinsa myös vanhalle mantereelle, saaden täten etulyöntiaseman. Immateriaalioikeuksia voidaankin käyttää myös muiden maiden kehittymisen estämiseen, vaikka tällainen toiminta on TRIPS²-sopimuksen vastaista. [Ganguli, 2000]

3.4. Patenttien laajuus ja suojaustaso

Patentointiin liittyy oleellisesti kaksi tärkeää muuttujaa, joilla patenttijärjestelmän toimintaa voidaan säätää: 1) uutuusvaatimus eli pienin parannus, joka hyvin tunnettuun tekniikkaan täytyy tehdä saadakseen patentin ja 2) suojaustaso eli kuinka paljon patenttia saa rikkoa ennen kuin teko muuttuu rangaistavaksi. Uutuusvaatimuksen ja suojaustason asettaminen oikealle korkeudelle on olennaista toimivan patenttijärjestelmän luomisessa ja ylläpitämisessä. Jos uutuusvaatimus on liian alhaalla, tekniikan julkistamisesta ei saada hyötyä, mutta kilpailulle luodaan esteitä, ja jos suojaustaso on liian alhaalla, keksijän etu vaarantuu ja kehitys saattaa tyrehtyä. Lopputulosta ei voida tietää ennen käytännön testiä, mutta erilaisia teoreettisia malleja sopivista arvoista on mahdollista rakentaa. Ainakin Thatcher ja muut sekä Bessen ja Maskin [1999] ovat luoneet tällaisia malleja ja soveltaneet niitä ohjelmistopatentteihin.

Thatcher ja muut [2004] tutkivat kansantaloustieteellisen mallin avulla uutuusvaatimuksen ja suojaustason optimaalista tasoa yhteiskunnan kannalta. Tutkimuksessa otettiin erityisesti huomioon ohjelmistopatentit. Kyseinen tutkimus vihjaa, että patenttijärjestelmä tulisi sovittaa jokaiselle teollisuudenhaaralle erikseen, toisin kuin tilanne on nyt. Tutkimuksen mukaan korkea suojaustaso lisää tutkimukseen käytettyjä varoja; uutuusvaatimuksen nostaminen voi joko lisätä tai vähentää tutkimukseen käytettyjä varoja; patenttien pakottaminen lisää innovaattorin tuottoa ja vähentää imitaattoreiden tuottoja sekä kuluttajien ja yhteiskunnan hyötyä ja voi johtaa jopa yleiseen sosiaalisen hyvinvoinnin vähenemiseen. Optimaalinen tilanne saavutettaisiin uutuusvaatimuksen ollessa korkealla ja suojaustason ollessa kohtuullinen.

Bessen ja Maskin [1999] puolestaan esittävät staattisen ja dynaamisen mallin, joilla patenttien hyödyllisyyttä voitaisiin arvioida. Jos staattinen malli pitää paikkansa innovaatio olisi korkeimmillaan, kun markkinoille pääsy on rajattua, ja jos dynaaminen malli pitää paikkansa, innovaatio olisi korkeimmillaan, kun

² Trade-Related Aspects of Intellectual Property Rights. WTO:n jäsenten välinen sopimus, joka määrittelee immateriaalioikeuksien suojaamisen minimitason.

markkinoille on tulossa paljon uusia kilpailijoita. Dynaamisen mallin mukaan korkea suojaustaso ei johda tuotekehityksen lisääntymiseen kuten staattisessa mallissa ja saattaa jopa laskea tätä. Bessenin ja Maskinin [1999] tutkimusten mukaan dynaaminen malli pitää paremmin paikkansa teollisuudenaloilla, jotka ovat riippuvaisia immateriaalioikeuksista.

Bessenin ja Maskinin sekä Thatcherin ja muiden tutkimusten yhdistävä tekijä on se, että molemmat pitävät ohjelmistoinnovaatioiden liian korkeaa suojaustasoa yhteiskunnan kannalta huonona. Toisaalta myöskään se, että suoja puuttuu kokonaan ei tunnu olevan optimaalinen tilanne.

3.5. Liiketoimintamenetelmäpatentit

Kaikki ohjelmistopatentit eivät koske tietokoneohjelmia, vaan osa niistä on itse asiassa liiketoimintamenetelmäpatentteja [Bessen and Hunt, 2003]. Nämä kaksi liittyvät toisiinsa sen takia, että tietyissä tapauksissa näiden kahden erottaminen toisistaan voi olla mahdotonta, koska tietojärjestelmät toimivat usein mahdollistajina uusille tavoille käydä kauppaa [Beresford, 2001].

Ehkäpä kaikkien aikojen kuuluisin ohjelmistopatentiksi naamioitu liiketoimintamenetelmäpatentti on Amazonin one-click –patentti, joka kattaa tuotteen tilaamisen kotiin yhdellä hiiren klikkauksella. Patentti on herättänyt suurta vastustusta, koska sen tekninen osio perustuu kuvaukseen siitä, kuinka eväste-tekniikkaa (cookies) voidaan käyttää lisäämään tilatieto WWW:hen, vaikka tämä nimenomaan oli evästeiden alkuperäinen tarkoitus. Amazonia onkin syytetty siitä, että se patentoi vasaran käytön naulojen hakkaamiseen uutena ideana. [Weiss, 2004] Amazonin patentin arvo ei ole tekniikassa vaan sen mahdollistamassa uudessa liiketoimintamallissa.

3.6. Strateginen patentointi

Strateginen patentointi tarkoittaa patenttien hakemista joko puolustus- tai hyökkäysmielessä. Näiden patenttien tarkoitus ei välttämättä ole suojata kehitystyötä vaan myös luoda esteitä uusien kilpailijoiden tulemiseen markkinoille sekä parantaa yrityksen neuvotteluasemia. [Bessen and Hunt, 2003; Coriat and Orsi, 2002] Strategisesti valittuja patentteja voidaan käyttää kilpailijoiden liiketoiminnan estämiseen myös silloin, kun hakijalla itsellään ei ole aikomustakaan hyödyntää kyseisiä keksintöjä [Olsson and McQueen, 2000]. Yhteiskunnan saavuttama hyöty on tällaisissa tapauksissa vähintäänkin kyseenalainen.

Strategista patentointia kuvastaa hyvin Chabchoubin ja Niosin [2004] tutkimus, jonka mukaan yrityksen omistamaan patenttien määrään vaikuttaa eniten yrityksen omistama patenttien määrä, eli kun yritys on alkanut patentoida se myös jatkaa valitsemallaan linjalla. Suhteettoman suuri osa ohjelmisto-

patenteista kuuluukin yrityksille, joiden päätoimiala ei ole ohjelmistotuotanto [Bessen and Hunt, 2003]. Nämä yritykset ovat selvästi sisäistäneet strategisen patentoinnin ja sen mahdollisuudet.

Esimerkkinä strategisesta patentoinnista voidaan käyttää voimakkaasti patentoivaa IBM:ää. IBM:n uusi toimitusjohtaja pienensi kehitysbudjettia yli miljardilla dollarilla 90-luvun alussa, mutta samaan aikaan patentointia lisättiin huomattavasti. IBM:n filosofian mukaan heidän suuri patenttikokoelmansa sallii heille pääsyn muiden yritysten kehitystyöhön ristiinlisensoimalla. [Bessen and Hunt, 2003]

3.7. Ristiinlisensointi ja patenti-pooli

Ristiinlisensointi (cross-licensing) tarkoittaa tilannetta, jossa kaksi tai useampi samoilla markkinoilla toimivaa yritystä sopii keskenään toistensa patentoitujen keksintöjen hyödyntämisestä, yleensä ilman erillistä korvausta. Ristiinlisensointia voidaan käyttää estämään uusien kilpailijoiden pääsy markkinoille, ostamaan itselle pääsy markkinoille tai pitämään markkinat toimivina tilanteissa, jossa patentinhaltijoita on paljon. Ristiinlisensointi on myös tapa ratkaista patenttirikkomukset oikeuden ulkopuolella: yritys A haastaa yritys B:n oikeuteen patenttirikkomuksesta, yritys B etsii kokoelmastaan patentin, jota A rikkoo ja asia sovitaan päikseen. Tällaisessa tilanteessa patenteilla on strateginen luonne.

Ristiinlisensointi on erityisen tärkeää aloilla, joilla kehitys on kumulatiivista, koska muuten se voi estää myös omien patenttien hyödyntämisen [Olsson and McQueen, 2000]. Ristiinlisensoinnista saatava hyöty kasvaa patenttiportfolion myötä [Mykytyn et al., 2002].

Ristiinlisensointiin liittyy läheisesti ns. patenti-pooli (patent-pool), eli saman alan patenteja omistavien yritysten yhteenliittouma, jossa mukana olevat yritykset saavat hyödyntää ilman erillistä sopimusta toistensa patenteja. Patenti-pooli on tehokas tapa pitää markkinat muutamien avainyritysten käsissä ja estää uusien kilpailijoiden synty. Perchaud [2002] esittää, että tietyissä tapauksissa tämä voi olla kartellilakien vastaista. Kartelli voisi tulla kysymykseen ainakin oligopolisilla markkinoilla, jos suurin osa yrityksistä on mukana tällaisessa järjestelyssä.

Open Sourcen vastavetona GNU:n perustaja Richard Stallman on perustanut ilmaisen patenttipoolin, jonka tarkoituksena on palauttaa patentit julkiseksi omaisuudeksi [Besaha, 2003]. Patenttipoolista on olemassa myös toinen positiivinen esimerkki: Bessen ja Maskin [1999] kuvaavat kuinka puolijohde-markkinoilla oli olemassa patenttipooli 1970-luvulla eikä sitä käytetty yhdenkään yrityksen pääsyn estämiseen markkinoille. Muut yritykset pitivät uusien yritysten tuomaa kehitystä tärkeämpänä kuin hetkellistä taloudellista etua ja

lisensoivat patenttinsa kaikille halukkaille, usein hyvinkin kohtuulliseen hintaan.

3.8. Liikesalaisuudet

Yritys voi käyttää strategiana myös sitä, ettei patentoi, koska näin mahdollisten innovaatioiden toteutustapaa ei tarvitse paljastaa, vaan ne voidaan pitää liikesalaisuuksina. Tämä on myös halvempi tapa toimia, eivätkä liikesalaisuudet vanhene kuten patentit. [Olsson and McQueen, 2000] Liikesalaisuuksien soveltamisesta ohjelmistokeksintöjen suojelemiseen ei tunnu olevan tutkimusta, mutta Randall ja muut [1996] huomauttavat, että jos ohjelman hyödyllisyys selviää ”pinnasta”, se ei kelpaa liikesalaisuudeksi lainopillisessa mielessä.

Tietokoneohjelmien kohdalla myös perinteisten salaisuuksien hyödyllisyyttä voidaan pitää pienenä, koska suurin osa ohjelmista on mahdollista takaisinmallintaa (reverse-engineering) melko pienellä vaivalla ja täten paljastaa niiden toimintalogiikka. Se, onko takaisinmallinnus laillista, riippuu kansallisesta lainsäädännöstä, mutta tällaisten rikkomusten todistaminen voi olla vaikeaa, sillä eräissä tapauksissa halutut asiat voidaan selvittää myös muilla keinoilla. Esimerkiksi verkkoprotokollan toiminnan voi selvittää analysoimalla verkkoliikennettä eikä takaisinmallinnus tässä tapauksessa ole tarpeen. Toisaalta esimerkiksi tehokkaan pakkausalgoritmin toiminnan selvittäminen voi vaatia koodin tutkimista.

3.9. Immateriaalioikeudet päätoimialana

Yhdysvalloissa on muutamia yrityksiä, joiden päätoimialana on immateriaalioikeuksien haaliminen ja niistä saatavien lisenssitulojen kerääminen. Nämä yritykset toimivat patentoimalla erilaisia tekniikoita tai ostamalla patenteja ja tämän jälkeen vaatimalla muilta alan toimijoilta lisenssimaksuja uhkaamalla näitä oikeudenkäynneillä.

Erikoiseksi tämän järjestelyn tekee se, että nämä yritykset eivät itse valmista mitään ja useat niiden omistamista patenteista ovat erittäin laajoja, omaavat hiukan epämääräisen uutuusarvon eikä niitä ole koeteltu oikeudessa. Lisäksi ei ole harvinaista, että näiden yritysten koko henkilökunta koostuu asianajajista ja patenttiasiantuntijoista, joiden tekninen osaaminen ei ole huipputasoa. Esimerkkeinä olkoot yhdysvaltalaiset Ideaflood ja Acacia Technologies. Ainakin Ideafloodia on syytetty pienempiensä ”kiusaamisesta” [Weiss, 2005; EFF, 2005].

4. Keskustelua ohjelmistopatenteista

Tämä luku käsittelee yleisimmin esiteltyjä syitä ohjelmistojen patentoitavuuden puolesta ja sitä vastaan. Tarkasteltaessa esitettyjä syitä tulee ottaa huomioon,

että useat niistä eivät itse asiassa koske tietokoneohjelmia sinällään, vaan patentoitavuutta yleisesti, nykyistä patenttijärjestelmää tai patenttitoimistojen toimintaa. Teoreettisessa keskustelussa näillä ei ole merkitystä, mutta käytännön markkinataloudessa ne voivat viedä pohjan hyvältäkin järjestelmältä. Ongelmat ovat tietyssä määrin eri maiden ja talousalueiden sisäisiä, mutta monet niistä esiintyvät muodossa tai toisessa kaikilla alueilla.

4.1. Tuotekehityksen suojeleminen ja sen hinta

Yleisin perustelu ohjelmistojen patentoitavuuden puolesta on sama, joka koskee kaikkia teknisiä keksintöjä: tarve suojella tuotekehitystyön tuloksia ja täten motivoida eri tahoja käyttämään resursseja tuotekehitykseen [Henkel, 1992]. Yleisin vastaväite tälle on, että ohjelmistoteollisuus on pärjännyt myös ilman patenteja [Oz, 1998]. Bessen ja Maskin [1999] huomauttavatkin, että ohjelmistoteollisuus on ollut hyvin innovatiivinen ja kasvanut lujaa vauhtia patenttien puutteesta huolimatta. Tämä ei tietenkään tarkoita, ettei patenteja voitaisi tarvita tulevaisuudessa. Koska ohjelmistoja koskeva patenttilainsäädäntö on hyvin vanha, ja ala on muuttunut huimaa vauhtia, voidaan perustellusti olettaa, että asia tulisi ottaa uudelleen tarkasteltavaksi [Randall et al., 1996].

Vapaa kilpailu on yleensä kuluttajien edun mukaista, mutta ideoiden nopea kopiointi ja kilpailevien tuotteiden lanseeraus voi johtaa tilanteeseen, jossa alkuperäinen kehittäjä ei pysty saamaan investoimiaan kuluja takaisin [Randall et al., 1996]. Ongelma korostuu ohjelmistoteollisuudessa, joka on tunnettu kyvystään luoda klooneja nopeasti. Nopea kopiointi voi aiheuttaa tilanteen, jossa halukkaita innovaattoreita on vaikea löytää, koska suojauksen puute poistaa pohjan tuotekehitysinvestoinneilta. Vähentynyt kilpailu alalla johtaa siten kehityksen hidastumiseen [Weiss, 2004]. Lisäksi voidaan väittää, että yhden yrityksen innovatiivinen toiminta pakottaa myös muut mukaan kehitystyöhön pysyäkseen mukana kilpailussa ja että patenttien puute sallii kehittäjien vain parantaa olemassa olevia ideoita uusien keksimisen sijaan aiheuttaen täten kehityksen hidastumista. [Chabchoub and Niosi, 2004; Oz, 1998].

Kehityskulujen suuruus ja siten tarvittavan suojauksen taso jakaa mielihetimitä. Tietyissä tilanteissa tietokoneohjelmien kehittäminen voi olla erittäin kallista ja aikaa vievää ja on luonnollista kaivata suojaa näille investoinneille. Esimerkiksi IBM käytti 2.5 miljardia dollaria OS/2 -käyttöjärjestelmänsä luomiseen ja markkinointiin. [Oz, 1998] Toisessa äärelaidassa ovat yhden miehen ohjelmistopajat, jotka tarvitsevat vain halvan tietokoneen ja hiukan aikaa ohjelmien tekemiseksi. [Stallman and Garfinkle, 1992; Perchaud, 2002]

4.2. Pienet yritykset ja patentit

Pienet yritykset voivat patenttien avulla puolustaa henkistä pääomaansa suurten yritysten plagioinnilta. [Oz, 1998; Weiss, 2004] Usein juuri imitaattorit ovat markkinoiden voittajia, eivät alkuperäisen idean keksijät; menestynein on se joka kykenee markkinoimaan omaa versiotaan tehokkaimmin. Tämä markkinointi vaatii paljon rahaa ja antaa siten varakkaille yrityksille etulyöntiaseman. Imitaattoreiden etuna on se, että heidän ei tarvitse maksaa tuotteen alkuperäisiä kehityskustannuksia, vaan he selviävät huomattavasti pienemmillä investoinneilla. [Oz, 1998, Henkel, 1992; Randall et al., 1996] Imitaattorit voivat toki olla myös hyödyllisiä, jos ne kopioinnin lisäksi jatkavat tuotteen kehitystä. Tämä korostuu tilanteissa, joissa alkuperäinen innovaattori lopettaa jatkokehityksen tai eri osapuolet valitsevat eri kehityssuunnan.

Patentit auttavat suojelemaan myös entisten työntekijöiden aiheuttamalta kilpailulta [Chabchoub and Niosi, 2004; Olsson and McQueen, 2000]. Tämän kaltainen kilpailu syntyy kun yrityksen tekniikan kehittäneet tai hyvin tuntevat työntekijät perustavat oman yrityksensä tai loikkaavat kilpailijalle tieto mukanaan. Ilman patenteja heidän ei luonnollisestikaan tarvitse maksaa osaansa yrityksen tuotekehityskustannuksista. Ongelma voidaan yrittää ratkaista myös työsopimuksilla, jotka kieltävät kilpailemisen tietyn ajan sisällä entisen työnantajan kanssa, mutta tällaisilla ehdoilla on vaikea estää epäsuoraa kilpailua, jossa entinen työntekijä vain toimittaa tietoa tekniikasta, mutta ei osallistu toimintaan muilla tavoilla.

Henkel [1992] käyttää esimerkkinä pienen yrityksen patenttien kautta saavuttamasta suojelusta omaa yritystään, joka kehitti tuotteen nimeltä Zoomracks, joka oli ideana uusi ja sisälsi uusia ominaisuuksia, joista osa myös patentoitiin. Zoomracks oli ohjelmoitu Atari ST:lle, jonka käyttäjäkunta oli suhteellisen pieni ja tulisi pian häviämään kokonaan. Idea oli kuitenkin hyvä ja sen kehittämiseen oli käytetty Henkelin mukaan kuusi vuotta aikaa ja huomattava summa rahaa. Tästä syystä Henkel hermostui, kun Apple Computer lanseerasi ilmaisen HyperCard-nimisen tuotteen, joka oli käytännössä kopio Zoomracksista. Henkel koki, että hänen keksintönsä oli varastettu ja haastoi Applen oikeuteen. Asia sovittiin oikeuden ulkopuolella ja Apple lisensoi patentit.

Vaikka patentit voivat olla pienen yrityksen pelastus, pienten yritysten resurssit eivät välttämättä riitä niiden saamiseen. Yhden patentin hankkiminen maksaa tällä hetkellä noin 10 000 euroa ja edellyttää yleensä alan asiantuntijan palkkaamista, koska patenteissa käytetty kieli ei ole tavallisen kuluttajan tai keksijän ymmärrettävissä. Perchaudin [2002] mukaan tärkeämmäksi on nousut kyky kuvata ongelma kuin sen ratkaisu. Hankala kieli vaikeuttaa patentti-

tutkimuksia, joka puolestaan voi johtaa vaikeuksiin löytää tiettyä aihetta koskevia patentteja, vaikka ne yritettäisiinkin tunnistaa [Stallman and Garfinkle, 1992].

Bessenin and Huntin [2003] mukaan todellisuudessa hyvin pieni osa ohjelmistopatenteista kuuluu pienille tai uusille yrityksille. Syynä tähän voidaan pitää pääoman puutetta [Chabchoub and Niosi, 2004]. Coriat ja Orsi [2002] huomauttavat, että tilanne on tietyiltä osin muuttumassa, sillä myös pienet, hyvin erikoistuneet yritykset ovat alkaneet hakea patentteja. Selityksenä tälle tarjotaan sitä, että pienten yritysten on helpompi saada rahoitus, jos niiden tuotteet on suojattu, joten ne ovat valmiita sijoittamaan patenttien hakemiseen [Olsson and McQueen, 2000].

Taloudellisen resurssien lisäksi pieniltä yrityksiltä voi puuttua tieto ja kokemus siitä, kuinka ja miksi patentteja tulisi hakea [Olsson and McQueen, 2000]. Olsson ja McQueen [2000] arvioivat, että kymmenessä Euroopan maassa olevista 4000 ohjelmistoalan yrityksestä noin puolet tarvitsisi patentteja, mutta näistä vain 800 on kykeneviä niitä hankkimaan. Kuitenkin pienet yritykset tuntuvat olevan innovatiivisimpia, joten juuri niiden tulisi kyetä suojaamaan keksintöjään. [Henkel, 1992; Perchaud, 2002] Toisaalta Bessen ja Hunt [2003] tyrmäävät väitteet siitä, että pienemmät patentointikustannukset lisäisivät tuotekehitystä. Heidän mukaansa ne johtavat lähinnä strategisen patentoinnin lisääntymiseen. Tämä puolestaan voi johtaa tilanteeseen, jossa markkinat hajoavat, koska patentinhaltijoita on liikaa [Weiss, 2004].

4.3. Ohjelmistoteollisuuden erityispiirteitä

Pienikin ohjelma voi käyttää kymmeniä eri algoritmeja ja rikkoa kymmeniä patentteja. Jos patentinhaltijoita on liikaa, kaikkien luvan saaminen voi osoittautua liian kalliiksi tai jopa mahdottomaksi pelkästään neuvottelukulujen takia [Coriat and Orsi, 2002; Perchaud, 2002]. Patenttitutkimukset ovat myös työläitä ja kalliita, joten pienet yritykset saattavat jättää leikin sikseen resurssien puutteen takia. [Bessen and Hunt, 2003; Oz, 1998] Toisiin teollisuudenaloihin verrattuna patenttien lisensoinnista aiheutuvat kulut eivät ohjelmistoteollisuudessa ole samassa suhteessa kehityskuluihin; ohjelmassa voi olla yhtä monta komponenttia kuin autossa, mutta kehityskulut ovat murto-osa auton kehityskuluista. Kuitenkin patenttijärjestelmästä aiheutuvat kulut ovat samat, olettaen, että yhtä moni komponentti on patentoitu [Stallman and Garfinkle, 1992].

Tuotekehityksen suojaaminen patenteilla on hyödytöntä, jos patentteja ei voida puolustaa väärinkäytöksiltä. Tämän takia onkin tärkeää ottaa huomioon myös mahdolliset patenttikiistojen aiheuttamat oikeudenkäyntikulut. Patenttikiistojen ratkaiseminen oikeudessa maksaa esimerkiksi Euroopassa vähintään 200 000 euroa ja Yhdysvalloissa hinta voi nousta yli miljoonan dollarin

[Perchaud, 2002; Henkel, 1992] Lisäksi ainakin Yhdysvalloissa ja Englannissa pitää kulkea oikeuden kautta myös silloin, jos haluaa mitätöidä kilpailijan patentin johonkin ilmiselvään "keksintöön". Tämä luo mahdollisuuden kilpailijoiden "kiusaamiseen"; hiukan epämääräisiä strategisesti valittuja patenteja esitellään yrityksille, joilla ei ole varaa lähteä oikeuteen katsomaan, ovatko kyseiset patentit lainvoimaisia. Ne joutuvat joko maksamaan tai vetämään tuotteensa pois markkinoilta [Henkel, 1992]. Tamain [1998] mukaan ongelmaa ei helpota tietokoneohjelmien korkea abstraktiotaso, joka vaikeuttaa patenttirikkomusten arviointia.

Patentit lisäävät riskejä ja kustannuksia ohjelmistoteollisuudessa ja tekevät uusien valmistajien, erityisesti yksityishenkilöiden ja pienten yritysten, tulon alalle vaikeammaksi, koska markkinoille tulo vaatii enemmän pääomaa kuin aikaisemmin [Oz, 1998; Perchaud, 2002]. Lisäksi riski oikeudenkäynneistä johtaa helposti kuluttajien maksaman hinnan nousuun [Bessen and Hunt, 2003]. Patenttien aiheuttama lisääntynyt pääomantarve poistaa niitä hyötyjä, joista ohjelmistoteollisuus on nauttinut, erityisesti pienet tuotantokustannukset perinteisempiin teollisuudenhaaroihin verrattuna. Tämä hyöty johtuu siitä, että prosessit eivät ole fyysisesti yhtä intensiiviä [Randall et al., 1996].

Bessenin ja Huntin [2003] tutkimuksen mukaan ohjelmistotalalla patentit eivät ole johtaneet lisääntyneeseen kehitystyöhön vaan ne ovat korvanneet kehitystyön ja vähentäneet tutkimukseen käytettyjä resursseja. Ilmiön voimakkuus kasvaa yrityksen koon mukana ja syyksi sille tarjotaan strategista patentointia. Toisaalta Bessen ja Hunt [2003] viittaavat Kortumiin ja Lerneriin, jotka selittävät ilmiön pääoman tehokkaammalla käytöllä.

4.4. Patentit ja pysyvä kilpailuetu

Mykytyn ja muut [2002] esittävät, että yritys voi saavuttaa pysyvän kilpailuedun ohjelmistopatenteilla. Nämä patentit voivat olla myös liiketoimintamenetelmäpatenteja, mutta koska tietokone toimii mahdollistajana, ne patentoidaan sen mukaan. Ohjelmistopatentit voivat saavuttaa merkittävän roolin pysyvän kilpailuedun tavoittelussa, koska kilpailuetua haetaan usein tietojärjestelmillä. Ongelmana on nopea kopiointi, joka johtaa hyvin pian kilpailuedun menettämiseen.

Pysyvän kilpailuedun saavuttaminen ohjelmistopatenteilla edellyttää kuitenkin sitä, että patentit kuuluvat yritykselle itselleen tai että se pystyy lisensoimaan ne yksinoikeudella. Tällaisesta strategisesta patentoinnista on olemassa viitteitä, sillä ohjelmistopatenteja on myönnetty suhteettoman paljon yrityksille, jotka eivät ole mukana ohjelmistoteollisuudessa [Bessen and Hunt, 2003]. Varsinaisten ohjelmistoyritysten omistamat ohjelmistopatentit eivät

välttämättä tällaista etua suo. Voidaan olettaa, että jos keksintö todella on hyödyllinen, niin yksinoikeus siihen ei tule halvaksi.

Tässäkin tapauksessa yhden yrityksen etu voi olla yhteiskunnan tappio. Tietokoneohjelmat ovat nykyään käytössä melkein kaikilla tuotannon aloilla ja esimerkiksi tavalliset kodinkoneet ovat alkaneet sisältää ohjelmistokomponentteja [McQueen, 2004]. Ohjelmistopatenttien vaikutukset voivat siirtyä ketjussa eteenpäin, aiheuttaen vahinkoa useilla muillakin aloilla estämällä yritysten pääsyn uudempaan tuotantotekniikkaan [Chabchoub and Niosi, 2004]. Tällaisten kerrannaisvaikutusten arviointi etukäteen on erittäin vaikeaa.

Lisäksi voidaan väittää, että tietokoneohjelmat eivät useinkaan sisällä sinällään uutta ideaa, vaan ovat jonkin jo tunnetun prosessin automatisointia. Patentointi koskee tässä tapauksessa myös alkuperäistä ideaa. [Oz, 1998] Beresfordin [2001] mukaan yleisesti tunnetun prosessin automatisointia ei normaalisti pidetä patentoitavana, mutta tämän automatisoinnin toteutustapa voidaan patentoida.

4.5. Peräkkäinen ja rinnakkainen innovointi

Ohjelmistoteollisuudessa vallitsee kaksi merkittävää kehitysmallia, peräkkäinen ja rinnakkainen. Peräkkäinen malli tarkoittaa sitä, että kehitys tapahtuu edellisten keksintöjen pohjalta, ja rinnakkainen sitä, että useat tahot pyrkivät keksimään saman ratkaisun samaan aikaan. Useat eri tutkijat ovat esittäneet huolestuneisuutensa näiden kehitysmallien ja patentoinnin välisestä ristiriidasta.

Peräkkäisessä mallissa patentointi saattaa haitata jatkokehitystä, koska innovaation hyödyntäminen vaatii lisenssiä tai lupaa kaikkiin edeltäviin keksintöihin. [Oz, 1998; Perchaud, 2002; Randall et al., 1996] Patentinhaltijoiden lisääntyessä ongelma kasvaa ja ainakin Bessenin ja Maskinin [1999] kansantaloustieteellisten mallien mukaan peräkkäinen innovointi saattaa kärsiä huomattavasti patenteista.

Rinnakkaisessa mallissa useat keksinnöt syntyvät jonkin toisen ongelman ratkaisun yhteydessä. Eri tahot keksivät tämän tyyppiset innovaatiot usein uudestaan ja uudestaan. Näiden keksintöjen patentointi haittaa kehitystä tulevaisuudessa, koska usein vain yksi voi saada patentin itselleen. [Oz, 1998] Myös Bessenin ja Maskinin [1999] mallien mukaan patentointi haittaa aloja, joilla innovaatio on rinnakkaista, sillä pienempi määrä yrityksiä lähtee mukaan kehittämään ratkaisua johonkin ongelmaan ja täten myös ratkaisun löytyminen on epätodennäköisempää.

Varian [1998] kuvaa tätä ongelmaa esittämällä, että liian lyhyet kehitysaskeleet johtavat kilpailuun yritysten kesken. Kaikki panostavat seuraavan loogisen askeleen saavuttamiseen enemmän kuin olisi perusteltua vain saadakseen

patentin itselleen. Tämä johtaa kaikkien paitsi patentin saaneen yrityksen kohdalla hukattuihin kehityskuluihin, jotka oltaisiin voitu käyttää myös paremmin. Näyttääkin siltä, että rinnakkaisen innovoinnin kohdalla keksintöjen uutuusarvon täytyy olla hyvin korkea, jotta patenttijärjestelmästä olisi hyötyä.

Rinnakkaisen kehityksen takia myönnettyt patentit koskevat usein jo toisen tunnetun ohjelman ominaisuuksia. Tämän ohjelman käyttäjistä tulee hetkessä patentin rikkojia [Oz, 1998]. Näin voi tapahtua kahdesta syystä: patentin haku- ja myöntämishetken väli on varsin pitkä tai patenttitoimistot eivät löydä edeltävää näyttöä hakemuksen tarkastamisen yhteydessä. Varsinkin Yhdysvalloissa edeltävä näyttö tuntuu jäävän kokonaan tutkimatta ennen patentin myöntämistä [McQueen, 2004]. Koska patentti tulee voimaan hakupäivästä alkaen, mutta julkistetaan myöhemmin, yritys ei voi tietää rikkovansa jonkun patenttia. Tietyt yritykset ovat jopa tahallisesti viivyttäneet patenttiensa julkistamista antaakseen markkinoiden kypsyä ja tulojen täten kasvaa [Varian, 1998] Stallman ja Garfinkle [1992] huomauttavat, että siinä 18 kuukaudessa, joka kuluu ennen kuin patentti julkistetaan, kilpailijat ovat saattaneet kehittää hyvinkin suuria ohjelmia.

Tietokoneohjelmien kehitysajat ovat muutenkin ongelmallisia patentoinnin suhteen. Noin 70% ohjelmista valmistuu alle 12 kuukaudessa ja yli 40% alle kuudessa kuukaudessa. Lisäksi yli 75% ohjelmista korvataan vuoden sisällä käyttöönotosta. Koska patentin saaminen kestää Yhdysvalloissa noin 34 kuukautta ja Euroopassa noin 18–24 kuukautta, keksinnöt ovat jo vanhentuneita ennen kuin patentit on myönnetty. Myös TRIPSin patenteille takaama 20 vuoden elinikä on suhteettoman pitkä ohjelmien elinikään verrattuna. [Perchaud, 2002; EFFI, 2005]

Tunnetuin esimerkki tällaisesta niin sanotusta submarine-patentista on kenties Unisysin patentti LZW-pakkausalgoritmiin. LZW-algoritmiä käytetään muunmuassa suositussa GIF-kuvatiedostoformaattissa. Tapaus tuli ajankohtaiseksi, kun Unisys esitti vuonna 1999 vaativansa lisenssimaksuja web-sivustoilta, jotka käyttävät GIF-kuvia, jotka on tehty ohjelmilla, joiden implementaatiota LZW-algoritmista ei ole lisensoitu. Asian teki ongelmalliseksi se, että kyseinen algoritmi oli julkaistu alan lehdessä vuonna 1984, mutta kenellekään ei kerrottu, että Unisys oli hakenut patenttia tekniikalle vuonna 1983. [Weiss, 2004] Unisysin patenttivaatimusten seurauksena useat ohjelmistovalmistajat poistivat GIF-kuvien tuen ohjelmistaan aiheuttaen päänvaivaa käyttäjilleen, koska tuen poistaminen johti yhteensopivuusongelmiin.

Yhteensopivuusongelmat voivat syntyä myös etukäteen, toisin sanoen yhteensopivuutta ei pystytä luomaan alkujaankaan. Tietokoneohjelmien yhteensopivuuden takaamiseksi tiettyjä osia koodista täytyy kopioida tai

emuloida ja näiden osien patentointi saattaa haitata yhteensopivuutta [Oz, 1998]. Voidaankin väittää, että voidakseen olla todella yhteensopivia, ohjelmien täytyy olla hyvinkin samankaltaisia [Coriat and Orsi, 2002] Sama koskee myös käyttöliittymiä: jos ne patentoidaan, jokaiseen ohjelmaan tulee kehittää omansa ja tämä on esteenä käyttöliittymien oppimiselle [Randall et al., 1996]. Periaatteessa käyttöliittymät kuuluvat ilmaisun eli tekijänoikeuslain piiriin, mutta myös käyttöliittymäpatentteja on myönnetty Yhdysvalloissa ja Euroopassa [Beresford, 2001; Oz, 1998].

Tekstinkäsittely on hyvä esimerkki ohjelmien samankaltaisuudesta: ohjelmat ovat ominaisuuksiltaan melkein yhteneviä, koska niiden kehitys ei ole enää kovinkaan vilkasta. Tavallisen kuluttajan voi olla vaikeaa käsittää miksi usealla eri ohjelmalla voi suorittaa samat operaatiot ja saada saman lopputuloksen, mutta ne eivät ole yhteensopivia.

Yhteensopivuusongelmien takia onkin esitetty, että ohjelmistopatentit saattavat haitata informaation luomista ja jakamista, vaikka tietokoneet on tarkoitettu sen välittämiseen ja ilmaisun tueksi. Toisaalta myös puhelin, radio ja äänilevyt on patentoitu, vaikka niillä on sama käyttötarkoitus. [Oz, 1998]

Ohjelmistokehityspiireissä keskustelu on ollut hyvin avointa. Kilpailu patenteista poistaa keskustelun samaa tekniikkaa kehittävien väliltä. [Oz, 1998] Tämä ongelma on korostunut ainakin Yhdysvalloissa, jossa myös akateemiset tutkijat ja heidän työnantajansa ovat alkaneet hakea patenteja tutkimustuloksilleen.

4.6. Algoritmien patentointi

Tamai [1998] on tutkinut algoritmien patentointia ja esittää, että myös algoritmit syntyvät usein rinnakkaisena innovointina, mutta eri käyttötarkoituksiin. Nämä algoritmit voidaan myöhemmin yhdistää geneerisen muodon luomiseksi tai ne saattavat olla ensin luodun geneerisen muodon sovelluksia.

Tämä muodostaa ongelman patentoinnin kannalta, koska patenttien todellista laajuutta on vaikea ennustaa ainakaan nykyisten järjestelmien ollessa käytössä. Tamai [1998] käyttää esimerkkinä lajittelualgoritmia, jonka määrittelyssä oli kuusi patenttivaadetta ja joka esiteltiin patenttitoimistoille Yhdysvalloissa, Euroopassa ja Japanissa. Patenttitoimistojen päätösten perusteella Tamai toteaa, että algoritmien patentoinnissa on neljä isoa ongelmaa: 1) geneerisen algoritmin patentointi ei onnistu, vaan jokainen käyttötarkoitus tulee eritellä hakemuksessa ja yhdistää johonkin fyysiseen laitteeseen patentin saamiseksi, 2) jos joku myöhemmin yleistää jonkun jo patentoidun algoritmin ja liittää alkuperäisin patentin erityiskohtana mukaan, on erittäin epätodennäköistä, että patentti myönnetään, 3) hyvin tunnetun tai tiettyyn käyttötarkoitukseen patentoidun algoritmin patentointi toiseen

käyttötarkoitukseen ei ole yksiselitteistä ja tulosta on vaikea arvioida etukäteen ja 4) geneerisen patentin uusien käyttötarkoitusten patentoitavuutta on vaikea arvioida etukäteen.

5. Uusia malleja tietokoneohjelmien suojelemiseen

Aika ajoin tietokoneohjelmien suojausta käsittelevässä kirjallisuudessa nousee esille uuden erikoislain luominen tähän tarkoitukseen. Eräiden mielipiteiden mukaan tästä olisi jo kokonaan luovuttu, mutta mielestäni edellä esiteltyjen syiden takia on kuitenkin syytä tarkastella millaisia näkemyksiä ideaalisesta suojauksesta on esitetty.

Oz [1998] kertoo kuinka asianajajien ja ohjelmistokehittäjien keskuudessa tehdyn tutkimuksen mukaan molemmat ryhmät kannattivat, että ohjelmistoja koskevan suojan pitäisi olla voimakkaampi kuin tekijänoikeuslaki, muttei yhtä voimakas kuin patenttilaki. Molemmat ryhmät kannattivat oman erikoislain luomista tähän tarkoitukseen. Henkelin [1992] mukaan tämä aiheuttaisi ongelmia sen suhteen, mikä lasketaan tämän uuden lain piiriin ja mitä rajatapauksille tehdään. Rajatapauksilla Henkel tarkoittaa tietokoneohjelmia, jotka on sidottu tiettyyn fyysiseen keksintöön.

Randal ja muut [1996] ehdottavat uutta mallia, joka suojelisi tietokoneohjelmia. Tämä malli perustuu neljään ominaisuuteen, joita tulisi suojella: lähdekoodi, lähdekoodin tuottama hyödyllinen toiminta, ohjelman rakenne sekä ohjelman metafora (esim. taulukkolaskenta, graafinen käyttöliittymä). He esittävät, että uuden mallin mukaan kaikkien, jotka hyötyvät uusista ideoista, tulisi osallistua niiden kehityskuluihin ja tarjottavan suojan tulisi olla sovitettu ohjelmistojen läpivienti- ja markkina-aikoihin. Oikeaa aikaa on vaikea määrittellä, mutta Randal ja muut [1996] arvelevat, että 2-5 vuotta olisi sopiva aika. Ozin [1998] tutkimuksen mukaan ohjelmistokehittäjien ja asianajajien toivoma aika olisi 14–22 vuotta.

Randal ja muut [1996] ehdottavat myös, että klooneja rajoitettaisiin. Klooneja voitaisiin määrittellä kolmella kriteerillä: 1) kuinka paljon toiminnallisuudesta ja suunnittelusta on huomattavan samanlaista, 2) onko toisen tuotteen kehitys ollut nopeaa ja riippuvaista ensimmäisestä ja 3) ovatko tuotteet melkein samanlaisia ja kilpailevatko ne samoilla markkinoilla. Uusista ideoista ja innovatiivisista tuotteista pidettäisiin elektronista rekisteriä, josta myös oikeuksien ostaminen kävisi helpolla standardoidulla tavalla. Samalla poistuisi kalliiden neuvottelujen tarve. Oikeudenhaltijoiden olisi myös pakko lisensoida keksintönsä kaikille halukkaille täten poistaen mahdollisuuden sulkea kilpailijat pois markkinoilta strategisella patentoinnilla.

6. Yhteenveto

Tietokoneohjelmien paremmalle suojaamiselle on olemassa hyviä perusteluja, mutta patenttijärjestelmä ei välttämättä nykyisessä muodossaan ole oikea tapa toteuttaa tätä suojausta. Ohjelmistopatentit voisivat suojella yrityksiä muiden epärehelliseltä kilpailulta ja suojata tuotekehityskuluja, mutta pienten yritysten resurssit eivät välttämättä riitä patenttijärjestelmän hyödyntämiseen. Tämä jättäisi ne nykyistä huonompaan asemaan isompiin verrattuna. Patentointikulujen laskeminen puolestaan johtaa lähinnä strategisen patentoinnin lisääntymiseen.

Ohjelmistoteollisuudessa vallitsevat kehitysmallit voivat lakata toimimasta patentoinnin seurauksena eivätkä patenttijärjestelmästä aiheutuvat kulut ole samassa suhteessa tuotekehityskuluihin kuin muilla teollisuudenaloilla. Lisäksi ohjelmistojen läpivienti- ja markkinointiajat ovat hyvin lyhyet verrattuna patenttien haku- ja elinaikoihin. Tästä syystä patentit ovat vanhentuneita jo julkistamishetkellään eivätkä siten edistä yhteiskunnan teknistä kehitystä. Paras ratkaisu tietokoneohjelmiin liittyvien innovaatioiden suojaamiseen olisi todennäköisesti oman erikoislain luominen.

Viiteluettelo

- [Beresford, 2001] Keith Beresford, European patents for software, E-commerce and business model inventions. *World Patent Information* **23** (2001) 253-263.
- [Besaha, 2003] Bob Besaha, Bounty hunting in the patent base. *ACM* **46**, 3 (March 2003) 27-29.
- [Bessen and Hunt, 2003] James Bessen and Robert M. Hunt, An Empirical Look at Software Patents. Research on Innovation, Available in electronic form: <http://www.researchineurope.org/documents/swpat.pdf>.
- [Bessen and Maskin, 1999] James Bessen and Eric Maskin, Sequential Innovation, Patents, and Imitation. Working paper, MIT Department of Economics.
- [Chabchoub and Niosi, 2004] Norhène Chabchoub and Jorge Niosi, Explaining the propensity to patent computer software. *Technovation* **25** (2005) 971-978.
- [Cook, 2002] Trevor Cook, EC Draft patent directive. *Computer Law & Security Report* **18**, 3 (2002) 197-200.
- [Coriat and Orsi, 2002] Benjamin Coriat and Fabienne Orsi, Establishing a new intellectual property rights regime in the United States - Origins, content and problems. *Research Policy* **31** (2002) 1491-1507.
- [EFF, 2005] <http://www.eff.org>, 24.11.2005

- [EFFI, 2005] <http://www.ffi.org> , 30.11.2005
- [Ganguli, 2000] Prabuddha Ganguli, Intellectual property rights. Imperatives for the knowledge industry. *World Patent Information* **22** (2000) 167-175.
- [Henkel, 1992] Paul Henkel, Debunking the software patents myth. *ACM* **35**, 6 (June 1992) 121-140.
- [McQueen, 2004] Douglas H. McQueen, Growth of software related patents in different countries. *Technovation* **25** (2005) 657-671.
- [Mykytyn et al., 2002] Kathleen Mykytyn, Peter P. Mykytyn Jr., Bijoy Bordeloi, Vicki McKinney and Kakoli Bandyopadhyay, The role of software patents in sustaining IT-enabled competitive advantage: a call for research. *Journal of Strategic Information Systems* **11** (2002) 59-82.
- [Olsson and McQueen, 2000] Henrik Olsson and Douglas H. McQueen, Factors influencing patenting in small computer software producing companies. *Technovation* **20** (2000) 563-576.
- [Oz, 1998] Effy Oz, Acceptable protection of software intellectual property: a survey of software developers and lawyers. *Information and Management* **34** (1998) 161-173.
- [Perchaud, 2002] Sylvain Perchaud, Software patents and innovation. Available in electronic form: http://www.juergen-ernst.de/download/swpat_studien/studie_perchaud.pdf.
- [PL] Suomen patenttilaki 15.12.1967/550
- [Pollock et al., 2004] Rufus Pollock, Liam Kavanagh and Jonas Maebe, Software Patents and the Current EU Legislation, FFII Software Patents Workgroup. Available in electronic form: http://www.ffi.org.uk/swpat/software_patents_summary.pdf
- [PRH, 2005] Patentti- ja rekisterihallitus. www.prh.fi , 21.3.2005
- [Randall et al., 1996] Randall Davis, Pamela Samuelson, Mitchell Kapor and Jerome Reichman, A new view of intellectual property and software. *ACM* **39**, 5 (March 1996) 21-30.
- [Stallman and Garfinkle, 1992] Richard Stallman and Simson Garfinkle, Viewpoint against software patents. *ACM* **35**, 1 (January 1992) 17-21.
- [Tamai, 1998] Tetsuo Tamai, Abstraction orientated property of software and its relation to patentability. *Information and Software Technology* **40** (1998) 253- 257.
- [Thatcher et al., 2004] Matt E. Thatcher, Taeha Kim and David E. Pingry, Welfare analysis of alternative patent policies for software innovations. To appear in *Decision Support Systems*.
- [Varian, 1998] Hal R. Varian, Markets for information goods, University of California, April 1998, draft.

[Weiss, 2004] Aaron Weiss, Patent Frenzy! *ACM NW* (September 2004) 17-23.

Monikerroksisten neuroverkkojen opetusmenetelmiä

Paavo Toivanen

Tiivistelmä.

Tutkielma käsittelee monikerroksisten neuroverkkojen opetusmenetelmiä. Siinä esitellään monikerroksisen neuroverkkolaskennan arkkitehtuuria, suhteutetaan laskennan olemusta symboliseen tietojenkäsittelyyn ja esitellään Rumelhartin ja McClellandin suosittu vastavirta-algoritmi. Lisäksi esitellään vastavirta-algoritmin puutteita ja siihen kehitettyjä parannuksia sekä uudempia neuroverkkojen opetusmenetelmiä, kuten useiden lähtöjen menetelmät. Lopuksi keskustellaan konnektionismin tieteenfilosofisista implikaatioista.

Avainsanat ja -sanonnat: neuroverkot, konnektionismi, oppiminen, vastavirta-algoritmi.

CR-luokat: A.1, F.1.1, F.2.2, I.2.6

1. Johdanto

Neuroverkkoja (neural networks, artificial neural networks, ANN) on esitetty vaihtoehdoksi tietojenkäsittelyn symboliselle paradigmalle ja myös klassisille tietokoneensuunnitteluperiaatteille, joita Tosig [2004] kuvailee seuraavasti:

”[V]on Neumann computer architecture” - - is based on the following two premises: (i) there is a clear physical as well as logical separation from where the data and programs are stored (memory), and where the computation is executed (processor(s)); and (ii) a processor executes basic instructions (operations) one at a time, i.e., sequentially.

Neuroverkot hyödyntävät idealtaan *paralleelilaskentaa* (parallel processing), eli käsittelevät tietoa yhtäaikaisesti useissa erillisissä toisiinsa kytköksissä olevissa yksiköissä. Neuroverkoille ominaista ja vierasta monille supertietokonearkkitehtuureille, jotka voivat käyttää jopa kymmeniätuhansia yhtäaikaisia prosessoreja, on näiden yksiköiden äärimmäinen yksinkertaisuus [Tosig, 2004].

Kielitieteessä ja muissakin neuroverkkojen sovellustieteissä on suosittu nimitystä *konnektionismi* siitä teoriasta tai teoriaryypästä, joka käyttää neuroverkkojen toimintaperiaatteita teorianmuodostuksessa.¹Konnektionismi pyrkii

¹ Keskustelua konnektionismin olemuksesta: Bechtel and Abrahamsen 2002; Clancey 1997; Harley 2002. Kriittisiä kannanottoja ovat esittäneet esimerkiksi Steven Pinker ja muut, On Language and Connectionism (<http://www.ecs.soton.ac.uk/~harnad/Papers/pinker.conn.html>, lainattu 12.9.2005) ja István S. N. Berkeley, Some myths of connectionism

erityisesti aivojen toiminnan kognitiiviseen selittämiseen. Ajattelutavan juuret ovat 1940-luvulla, jolloin analogia aivoihin esitettiin esikuvaksi tietojenkäsittelylle ja jolloin alettiin myös ensimmäistä kertaa tutkia pienistä elementaarista yksiköistä järjestyneiden verkkojen laskennallisia ominaisuuksia. Aluksi todistettiin, että tällainen verkko saattoi suorittaa loogisia laskutoimituksia. Sopivalla tapaa rakennetut verkot suoriutuivat myös lupaavasti monimutkaisemmista tehtävistä kuten hahmontunnistuksesta. Yhteys aivojen kognitiivisiin kykyihin oli selvä, olivathan varhaiset mallit syntyneet juuri aivojen rakenteen inspiroimina. [Bechtel and Abrahamsen, 2002]

Neuroverkkojen toinen tuleminen tapahtui tietojenkäsittelyn symbolistisen kehitysvaiheen jälkeen 1980-luvulla. Tällöin konnektionismi antoi virikkeitä useille kognitiotieteeseen yhteydessä oleville tieteenaloille. Alan perusteokseksi muodostui Rumelhartin ja McClellandin *Parallel Distributed Processing* (1986), joka havainnollisti kognitiivisia ideoita kieleen ja hahmontunnistukseen liittyvillä malleilla. [Bechtel and Abrahamsen, 2002.] Syntyneen tieteellisen liikkeen vaikutuspiiriin kuuluivat erityisesti tietojenkäsittelytiede, filosofia, kieli-tiede ja psykologia.

Neuroverkkojen suorituskäkyä on sittemmin tutkittu sekä matemaattisesti että empiirisillä kartoittavilla tutkimuksilla. Vastoin varhaisia näkymiä myöhemmin on esitetty, että neuroverkot eivät ole ontologinen vaihtoehto symboliselle tietojenkäsittelylle, vaan pikemminkin eräs lähestymistapa symbolin käsittelyyn ja strukturoitujen systeemien rakentamiseen [Clancey, 1997; Bechtel and Abrahamsen, 2002]. Toisaalta neuroverkkoarkkitehtuurit ovat vallanneet laajan sektorin sovellusaloilla, joilla niitä käytetään erilaisissa tehtävissä ja usein yhteistyössä perinteisten symbolisen paradigman inspiroimien laskennallisten mallien kanssa.

Neuroverkkojen teoria on kehittynyt paljon viime vuosikymmeninä. Erityisen tärkeä neuroverkoille ominainen piirre on tapa, jolla ne yhdistävät tiedon ja oppimisen. Neuroverkkoa voidaan opettaa kumulatiivisesti lisäten sille uusia tehtäviä samalla kun se säilyttää kompetenssia vanhoissa jo opituissa tehtävissä. Tämän lisäksi rakenteeltaan erilaiset neuroverkot voivat käsitellä samaa dataa tai sen eri tavalla hajautettuja variaatioita, mikä mahdollistaa laajan paletin dataan eri tavoilla reagoivia sumeita funktioita. Koska verkon opettaminen vaikuttaa sen käyttäytymiseen, tarjoutuu neuroverkoille runsaasti sovellus-alueita esimerkiksi robotiikan ja hahmontunnistuksen alalla. Rajatuissa tilanteissa voidaan jopa soveltaa säänneltyä itseoppimista, jota tässä tutkielmassa ei käsitellä teorian tasolla.

(<http://www.ucs.louisiana.edu/~isb9112/dept/phil341/myths/myths.html>,
12.9.05).

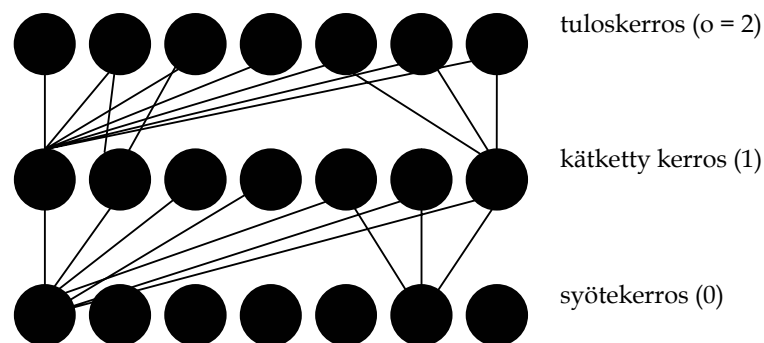
lainattu

Tarkastelen tutkielmassani monikerroksisten neuroverkkojen opetusmenetelmiä. Alkuperäinen mielenkiintoni kohdistui neuroverkkojen sovelluksiin kielitieteen konnektionistisissa malleissa, mutta keskityn verkkojen opettamiseen, eli aiheeseen, joka ei ole sidoksissa pelkästään konnektionismiin. Tutkielmani tarkoitus on kuvailla vastavirta-algoritmia ja siihen rinnastettavia monikerroksisten neuroverkkojen opetusmenetelmiä sekä niihin liittyviä erilaisia hyöty- ja haittanäkökohtia verkkojen sovelluksia ajatellen. Keskustelen myös lyhyesti neuroverkkojen oppimiskäyttötymisen filosofisista ja kielitieteellisistä implikaatiosta.

2. Neuroverkkolaskennan yleisistä periaatteista

2.1. Verkon rakenteesta

Seuraavassa kaavakuvassa on esitetty eräs kolmikerroksinen neuroverkko.



Kuva 1. Kolmikerroksinen neuroverkko kätkeytyllä kerroksella. (Kaikkia solujen välisiä yhteyksiä ei ole piirretty kuvaan.)

Kullakin kuvassa 1 esitetyn verkon kolmesta kerroksesta on joukko *neuroneja*, joista jokainen on yhteydessä seuraavaksi ylemmän ja yhtä alemman kerroksen kaikkiin neuroneihin. Verkko on siis jaettu useihin erillisiin alajoukkoihin [Sarkar, 1995]. Esimerkkimme on siinä mielessä erityistapaus, että verkon eri kerroksissa ei yleensä ottaen tarvitse olla samaa määrää neuroneita. Verkon yhteyksien (tästä eteenpäin *kaarien*) muodostama topologia voi myös vaihdella. Kaariensa puolesta verkko voi olla suuntaamaton tai suunnattu graafi, eli kaaret yksiköiden välillä ovat joko kaksisuuntaisia, jolloin on kyseessä *interaktiivinen verkko* (interactive network), tai vain toiseen suuntaan vaikuttavia, jolloin puhutaan *syöttöverkosta* (feedforward network, FFANN).

Kaarilla on numeeriset painoarvot, ja vastaavasti kullakin neuronilla on arvo. Lisäksi neuroneilla on *aktivaatiofunktio* (activation function), joka määrää

niiden käyttäytymistä, eli sitä, miten neuronit syöttävät tietoa toisiin neuroneihin. Aktivaatiofunktio on ei-lineaarinen.[Duda 2001.]

Kätketyn kerroksen merkitys on olennainen verkon toiminnan kannalta, koska se lisää laskennallista tehoa.1940-luvulla esitetyn analyysin mukaan kaksikerroksinen neuroverkko, joka ei siis sisällä kätkettyä kerrosta, suorittaa loogiset laskutoimitukset AND ja OR oikein, mutta se ei voi oppia XOR-funktiota. Kun kätketty kerros lisätään, XOR-funktion ja edelleen yleisemmän ns. 3-bittisen pariteettiongelman ratkaiseminen on mahdollista neuroverkon avulla. [Bechtel and Abrahamsen, 2002; Duda et al., 2001.] Yleisesti kaksikerroksinen neuroverkon eli *perseptronin* (two-layer neural network, perceptron) toiminta vastaa *lineaarista erottelufunktiota* (linear discriminant function) ja monikerroksisen *ei-lineaarista erottelufunktiota* (non-linear discriminant function) [Duda et al., 2001]. Jos tarkastellaan erottelutehtävää, jossa on jaettava osiin tietty avaruus, kaksikerroksinen verkko kykenee rajaamaan tästä avaruudesta suorien viivojen reunustamia segmenttejä, kun taas monikerroksinen verkko voi rajata viivasegmenttejä monimutkaisempia paraabeli- tai pallosegmenttejä sekä muita ei-lineaarisia muotoja.

Kuvan 1 neuroverkko koostuu *syötekerroksesta* (input layer), *kätketystä kerroksesta* (hidden layer) ja *tuloskerroksesta* (output layer).² Lisäksi neuroverkkoon voi vielä kuulua *vinoutusyksikkö* (bias unit), joka asetetaan syöttämään koko verkkoon tai sen osaan jatkuvaa tasaista syötettä, ja kätkettyjä kerroksia voi olla useampia kuin yksi. [Duda et al., 2001.]

2.2. Aktivaation eteneminen verkossa

Neuroverkoilla on kaksi pääasiallista toimintatapaa, *syöttö* (feedforward) ja *oppiminen*. Syöttö koostuu datan (tästä eteenpäin *kuvio*) ajamisesta verkon läpi niin, että tuloskerroksen yksiköiltä saadaan lopulta jotkin syötekerrokselle syötettyä kuviota vastaavat tulosarvot. Oppiminen puolestaan tarkoittaa verkon parametrien muuttamista jonkin kuvion tai useiden kuvioiden antamien tulosten perusteella niin, että tulokset tuodaan lähemmäs toivottuja opetus- eli *kohdearvoja*.

Opetettava uusi neuroverkko asetetaan alkutilaan, jolloin kaaret saavat yleensä pienet satunnaiset lähtöarvot.³Seuraavaksi syötekerrokselle syötetään tietty kuvio. Syöttömuoto on hajautettu, eli esimerkiksi kuva voidaan antaa syötekerroksen neuroneille pikselidatana tai puheääni äänisignaalina, ja kuvio

²Useampikerroksisen neuroverkon kerroksia indeksoidaan yleensä arvoilla 0,1,...,o.Tässä 0 on syötekerroksen indeksi ja o tuloskerroksen [Sarkar, 1995].

³Jos kunkin kaaren painoarvo nollattaisiin tai jos kaikkien kaarien painoarvo olisi sama, ei verkon tilaa voisi muuttaa esimerkiksi vastavirta-algoritmillä, jota kuvaillaan alempana. [Duda et al., 2001.]

on levitetty kaikille syötekerroksen neuroneille. Syötetyn aktivaation etenemistä mallinnetaan sykleissä.⁴ Aktivaatio etenee syötekerrokselta kätkeytyyn kerrokseen ja siltä edelleen tuloskerrokselle kaarien painoarvojen ja neuronien aktivaatiofunktioiden mukaisesti. Kunkin neuronin kullakin kierroksella saamaan arvoon vaikuttavat sen aikaisempi aktivaatio ja kaikkien siihen yhteydessä olevien neuronien aktivaatiot. Jos kyseessä on interaktiivinen verkko, aktivaatio etenee kaksisuuntaisesti, eli ehdittyään esimerkiksi tuloskerrokselle se etenee seuraavalla kierroksella takaisin päin kohti kätkeytyä kerrosta ja edelleen kohti syötekerrosta kunkin neuronin lähettäviä kaaria pitkin.⁵ [Bechtel and Abrahamsen, 2002.] Verkon tulosarvon aktivaatio voidaan lukea haluttaessa tuloskerrokselta tai odottaa, että se vakiintuu johonkin jakaumaan. Syöttö-toiminto on pääpiirteissään kuvaillun kaltainen.

Neuroverkon toimintaa voi havainnollistaa myös matemaattisella analogialla, joka kuvaa verkkojen yhteyttä symboliseen tietojenkäsittelyyn: neuroverkko antaa jonkin lähtöjoukon (annettu informaatio) aktivaatioon liittyvän tulosjoukon aktivaation (tuloskerroksen informaatio), eli se toimii tässä kuten funktio [Jordan and Bishop, 1996]. Tiedon esitys on kuitenkin hajautettu neuronien arvoihin ja verkon rakenne voi vaihdella. Eräs tyypillinen verkkoarkkitehtuuri kuvaa n -ulotteisen vektorin $\bar{x} \in R_n$ arvolle $y \in R$, siis yhden neuronin muodostamalle tulosjoukolle.

2.3. Verkon opettaminen

Neuroverkkoja on käytetty laajasti sekä luokittelutehtäviin että oikeiden arvojen ennustamiseen erilaisissa ratkaisua vaativissa tilanteissa. Opetusvaiheen yleisenä päämääränä on minimoida virhe, jonka verkko antaa joukolle dataa opetuksen jälkeen [Martí and El-Fallahi, 2004]. Neuroverkolle voidaan siis opettaa vaihtelevankokoisia tunnistustehtävien joukkoja. Toimintansa kannalta valmiiksi opetetun neuroverkon suurin ero symboliseen funktioon nähden on sumeus eli se, että verkko antaa halutun tuloksen likiarvoja ja toimii jossakin määrin myös epätäydellisillä syötteillä. Neuroverkko voi myös suorittaa interpolaatiota eli tehdä arvauksia oppimiensa ennakkotapausten perusteella, vaikka opetuksessa syötetty data onkin yleensä diskreettiä eli koostuu yksittäisistä otoksista. Jordan ja Bishop [1996] esittävätkin, että neuroverkkojen yhteys

⁴ Mallinnus tapahtuu sykleinä tietysti vain, jos verkko on toteutettu perinteisten tietokonearkkitehtuurien varaan eli se toimii lineaarisen tietojenkäsittelyn pohjalta. Todella paralleelin arkkitehtuurin yhteydessä, kuten impulssin eteneminen aivoissa osoittaa, ei voida puhua yhtä selväräjaisesti sykleistä kuin perinteisissä konnektionistisissa mallinnustilanteissa. Kehittyvistä tulevaisuuden vaihtoehdoista Tasic 2004; Clancey 1997.

⁵ Esimerkiksi puheentunnistuksessa käyttökelpoiseksi osoittautunut idea on tehdä silmukka tuloskerrokselta jollekin alemmalle kerrokselle, jolloin verkkoa voidaan käyttää jatkuvan signaalin prosessoimiseen.

tilastollisiin menetelmiin on muodostunut tärkeäksi sekä neuroverkkojen ymmärtämisen kannalta että uusien neuroverkkolaskennan metodien kehittämisessä.

Neuroverkkoarkkitehtuurien haittapuolena on se, että verkon oppimistarkkuus pysähtyy tietylle etäisyydelle ihannearvoista riippuen opetusmenetelmästä ja verkon rajoituksista [Duda et al., 2001]. Haluttaessa tulosarvot saadaan tietysti diskreeteiksi. Toisaalta kahta useampikerroksiset verkot voivat oppia opetettavan datan ”liian hyvin” yksittäistapauksina, jolloin niiden antamien tulosten yleistyvyys kärsii.[Duda et al., 2001.]Duda ja muut esittävät, että kolmikerroksinen neuroverkko voidaan teoreettisesti ottaen rakentaa antamaan oikeita approksimaatioita mille tahansa jatkuvalla funktiolle, mikäli verkko täyttää tietyt ehdot eli siinä on sopiva määrä kätketyn kerroksen neuroneita, sopivat ei-lineaariset erottelufunktiot ja sopivat kaarien painoarvot [Duda et al., 2001].

Esimerkki [Martí and El-Fallahi, 2004]. Tarkastellaan erästä ennustamistehtävää ja siihen liittyvää funktiota $f: \mathbb{R}_n \rightarrow \mathbb{R}$ ja neuroverkko NN. Päämääränä on löytää sellaiset verkon kaarien painoarvot w , joilla verkon tulos $NN(x,w)$ approksimoi arvoa $f(x)$. \square

Vastaavasti luokittelutehtävässä verkolle etsitään toisistaan selvästi erotettavia tulosarvoja vastaamaan syötedatan eri kuvioita (jotka voivat olla esimerkiksi kuvia tai äänisignaaleja), jolloin kuviot voidaan sijoittaa tarpeellisiin kategorioihin. Kielellistä kognitiota mallinnettaessa syöte saattaa olla puhesignaali ja vaadittu tulos tietyllä tapaa esitetty lause, jonka sanoja voidaan verrata mallin toisen osan sisältämän leksikon sanoihin [Harley, 2001].

Tunnetuin neuroverkkojen opetusalgoritmi on matemaattisesti melko yksinkertainen *vastavirta-algoritmi* (back-propagation, error back-propagation, EBP) (ks. luku 3). Vastavirta-algoritmi on yleistys yksikerroksisen neuroverkon oppimisessa käytetystä *delta-säännöstä*, jota kutsutaan tällä nimellä, koska oppimisen määrä on siinä verrannollinen todellisen tulosteen ja opettajan antaman toivotun tulosteen väliseen eroon. [Sarkar, 1995.]Vastavirta-algoritmin peruseriaate on se, että tietylle kuviolle saatua todellista tulosta verrataan haluttuun korjausarvoon ja näiden erotuksen perusteella säädetään kaarien painoja siten, että todellinen tulos siirtyy lähemmäs korjausarvoa.

Muita neuroverkkojen opetusmenetelmiä ovat esimerkiksi liittogradienttimenetelmät, geneettiset algoritmit, tao-voimakas vastavirta-menetelmä ja tabuhaku [Martí and El-Fallahi, 2004; Bayro-Corrochano et al., 2005; Pernía-Espinoza et al., 2005; Duda et al., 2001].

3. Vastavirta-algoritmi Rumelhartin ja McClellandin mukaan

3.1. EBP FFANN-verkossa

Vastavirta-algoritmi eli EBP on Sarkarin [1995] mukaan ollut suosituin syöttöverkkojen (FFANN) opetusmenetelmä. Algoritmin suosio perustuu Rumelhartin ja McClellandin teoksen *Parallel Distributed Processing: Explorations in the Microstructures of Cognition* (1986) menestykseen sekä algoritmin helppoon omaksuttavuuteen. Esittelen seuraavaksi vastavirta-algoritmin sekä siihen kuuluvia käsitteitä [Sarkar, 1995; Duda et al., 2001; Pernía-Espinoza et al., 2005]⁶.

Neuroverkolle tai sen neuroneille paloittain määritellään *aktivaatiofunktio* out_{pi}^l , joka käyttää kullekin neuronille tulevaa kokonaissyötettä sen tulosteen määrittämiseksi. Sigmoidinen funktio

$$out_{pi}^l = f(net_{pi}^l) = \frac{1}{1 + e^{-\beta net_{pi}^l}} \quad (1)$$

on Rumelhartin ja McClellandin [1986] esittelemä aktivaatiofunktion tavallisin muoto. Kaavassa (1) l indeksoi neuroverkon kerrosta, ja *kokonaissyöte*

$$net_{pi}^{l+1} = \sum_{j=1}^{n_l} w_{ij}^l out_j^l + bias_i^{l+1}$$

on neuronille u_i^{l+1} yhteenlaskettu syöte toisista neuroneista u_j^l . Arvo β kaavassa (1) määrittää aktivaatiofunktion jyrkkyyttä, ja sitä on kutsuttu verkon *oppimisherkkyydeksi* tai *herkkyydeksi*.

Energiafunktio (energy function) E määrittää verkon oppimista. Se on funktio, joka on määritelty mittaamaan vastavirta-algoritmin tehokkuutta. Rumelhart ja muut käyttivät alun perin neliövirheen summaa

$$E = \frac{1}{2} \sum_{i=1}^{n_0} (\varepsilon_{pi})^2$$

energiafunktiona. Kaavassa ε_{pi} tarkoittaa tuloskerroksen neuronin $u_{pi}^{l=o}$ virhettä, joka määritellään halutun ja todellisen arvon erotukseksi. Neliövirheen summaa energiafunktiona kutsutaan neuroverkon *opetusvirheeksi* tai luokitteluvirheeksi (training error) J tietyssä tunnistustehtävässä, ja se siis määritellään kaikkien tulosjoukon yksikköjen todellisten tulosten y_k ja korjausarvojen k_k neliöön korotettujen erotusten summaksi eli

$$J(\bar{\mathbf{w}}) \equiv \frac{1}{2} \sum_{k=1}^c (k_k - y_k)^2 = \frac{1}{2} \|\bar{\mathbf{k}} - \bar{\mathbf{y}}\|^2,$$

⁶ Viimeisessä lähteessä esitellään joukko neurolaskennassa melko vakiintuneita notaatioita.

missä $\bar{\mathbf{k}}$ ja $\bar{\mathbf{y}}$ ovat korjausarvovektori ja tulosvektori, jonka pituus on c , ja $\bar{\mathbf{w}}$ kuvaa verkon kaikkien kaarien painoarvoja. Muita käytettyjä energiefunktioita ovat muun muassa *ristientropiafunktio* ja *Minkowskin virhefunktio* [Duda et al., 2001, 318.]

Neuroverkon *oppimisopeus* (learning rate) η on koko verkolle tai sen kaarille paloittain määrätty kerroin, joka määrittää kaarien arvojen säätämistä vastavirta-algoritmissa. Se siis ilmaisee suhteellisen nopeuden, jolla kaarien painoarvoja siirretään kohti opetusvirheen avulla määritettyjä ihannearvoja. Oppimisopeudella on tärkeä rooli, koska sen liian pieni arvo hidastaa oppimista, kun taas liian suuri arvo aiheuttaa heittelemistä estäen tälläkin tapaa verkkoa oppimasta annettua tehtävää [Sarkar, 1995, 525]. Käytännössä sopiva arvo riippuu tehtävästä, jota käsitellään.

Vastavirta-algoritmin keskeinen kaava on

$$w_{ij}^l[s+1] = w_{ij}^l[s] + \eta \delta_{pi}^{l+1} out_{pj}^l[s],$$

jossa s tarkoittaa opetusprosessin iteraatiota eli kierroksen numeroa, w verkon yksittäistä kaarta ja δ eli *porrasgradientti* (energy-weight gradient) kaaren painoarvolle on muotoa

$$\delta_{pi}^{l+1} = -\frac{\partial E_p}{\partial net_{pi}^{l+1}}.$$

Kätketyn ja tuloskerroksen kaarille lasketaan porrasgradientit eri kaavoilla, joista johtamisen jälkeen saadaan

$$\delta_{pi}^0 = out_{pi}^o (1 - out_{pi}^o) \varepsilon_{pi}$$

tuloskerroksen neuronille ja

$$\delta_{pi}^{l+1} = out_{pi}^{l+1} (1 - out_{pi}^{l+1}) \sum_{k=1}^{n_{l+2}} \omega_{ki}^{l+1} \delta_{pk}^{l+2}$$

neuronille i kätketyssä kerroksessa.

EBP on algoritmina seuraavankaltainen. Opetus on iteratiivinen prosessi. Jokainen iteraatio koostuu seuraavista vaiheista [Sarkar, 1995]:

- (1) Valitaan opetettava kuvio.
- (2) Syöttö: esitetään syötekuvio FFANN:n syötekerrokselle ja määritetään verkon tulostekuvio.
- (3) Takaisinsyöttö: lasketaan kunkin neuronin virhesignaali. Laskenta alkaa tuloskerrokselta ja etenee taaksepäin kerroksittain kätketyille kerroksille.
- (4) Kaarien painojen säätäminen. Säädetään painoja saatujen tulosten perusteella.

Sarkar [1995] toteaa, että vastavirta-algoritmille on esitetty muitakin käyttökohteita kuin neuroverkot. Alkuperäinen EBP on kuitenkin aiheuttanut paljon hankaluuksia todellisessa käytössä, mikä johtuu useista eri syistä:(1) se on

äärimmäisen hidas silloinkin, kun se konvergoi kohti haluttua tulosta, (2) se saattaa juuttua kiinni *virhepinnan* (error surface) paikallisiin minimeihin ennen kuin on oppinut kaikki hahmot, (3) se on herkkä alkuehtojen vaikutuksille, (4) se saattaa alkaa heittelehtiä. [Sarkar 1995, 520.] Vastavirta-algoritmin puutteita korjaamaan on kehitetty erilaisia muunnelmia.

3.2. Stokastinen opetus, niputettu opetus ja online-opetus

Käsiteltäessä neuroverkolle opettavien kuvioiden erikokoisia joukkoja turvaudutaan eri tilanteissa vastavirta-algoritmin eri variaatioihin. Duda ja muut [2001] jakavat johtavat opetusmenetelmät kolmeen protokollaan, *online-opetukseen* (online training), *stokastiseen opetukseen* (stochastic training) ja *niputettuun opetukseen* (batch training), joita käsitelen tässä kohdassa, sekä hakuja käyttävään oppimiseen (learning with queries), jossa verkon tuloksia käytetään uusien opettavien kuvioiden valinnassa.

Stokastisessa vastavirta-algoritmissa opettavasta joukosta valitaan satunnaisotannalla opettavia yksiköitä ja verkon painoarvot päivitetään jokaisen kuvion syötön jälkeen. Niputetussa vastavirta-algorimissa kaikki yksiköt syötetään kertaalleen verkon läpi ja niille lasketaan samalla virhesignaali, mutta opetusalgoritmi ajetaan vasta syklin lopuksi yhteenlaskettujen virhesignaalien perusteella. Näitä opetuksen kokonaissyklejä eli *epokkeja* (epoch) voidaan suorittaa useita molemmissa vastavirta-algoritmin muodoissa. Online-opetuksessa kukin datan yksikkö esitetään verkolle vain kerran, joten neuroverkko toimii tällöin "lennosta".⁷ Duda ja muut [2001] esittävät algoritmit mainituille opetusmenetelmille.

Jokaisella johtavalla opetusprotokollalla on etunsa. Online-opetusta käytetään silloin, kun opettavan datan määrä on niin suuri tai muistia tarvitaan niin paljon, että datan säilyttäminen muistissa on liian kallista. Dudan ja muiden [2001] mukaan useimpia todellisia luokittelutehtäviä käsitelläänkin niputetulla tai stokastisella opetuksella.

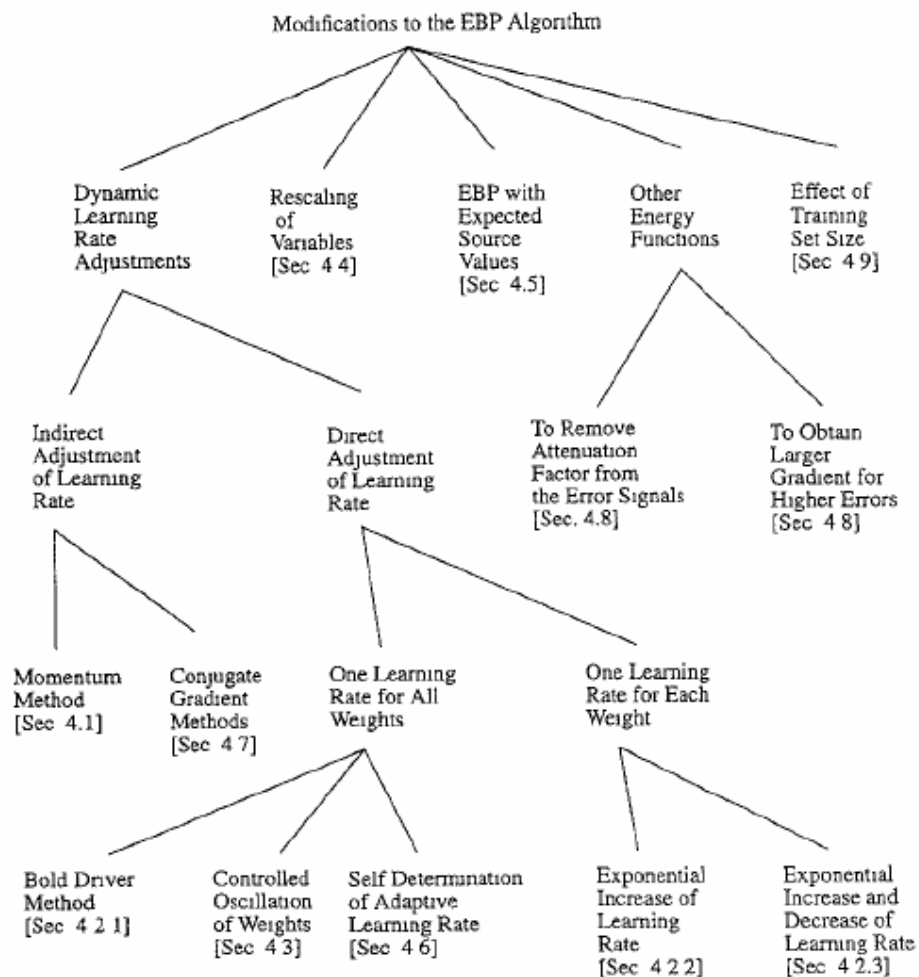
Niputettu opetus on puolestaan tyypillisesti hitaampaa kuin stokastinen; yksi syy tähän on se, että opetuksessa käytetty data on yleensä varsin redundanttia eli se sisältää runsaasti samaa informaatiota. Tästä syystä stokastista opetusta suositetaan monissa tilanteissa. Niputettu opetus tarjoaa kuitenkin joitakin toisen asteen tekniikoita, joita ei voida helposti yhdistää stokastiseen opetukseen. [Duda et al., 2001.]

⁷ Rumelhart ja McClelland esittelevät termit *online backpropagation* ja *batch backpropagation* hieman eri tavalla kuin Duda ja muut [Sarkar, 1995, 523]. Käytän Dudan ja muiden määritelmää.

Martí ja El-Fallahi [2004] käsittelevät neuroverkkojen opetusmenetelmien suorituskykyä sellaisissa aikakriittisissä ennustustehtävissä, joissa pyritään opettamaan verkkoa lyhyessä ajassa. Tällöin esimerkiksi syötearvot antava järjestelmä myös käyttää verkon antamia tuloksia hyväkseen. Näissä tilanteissa turvaudutaan myös online-opetukseen.

3.3. Parannuksia vastavirta-algoritmiin

Sarkar [1995] esittää seuraavan skeeman vastavirta-algoritmiin kehitetyistä muunnelmista (hakasulut kuvassa viittaavat Sarkarin artikkeliin):



Kuva 2. Eräs EBP:n parannusmetodien luokittelu (Sarkarin [1995] mukaan).

Sarkarin luettelemat metodit voidaan jakaa ryhmiin, kuten kuvan puusta käy ilmi. Näitä ryhmiä ovat (1) dynaaminen oppimisnopeuden säätäminen, jossa oppimisnopeuden kertoimia säädetään oppimisen edetessä. Säätäminen voidaan suorittaa suoraan tai epäsuorasti. Esimerkkejä epäsuorasta säätämisestä ovat *liikemäärämetodi* (momentum method) ja *liittogradienttimetodit* (conjugate gradient methods). Nämä menetelmät eivät kuitenkaan anna tyydyttäviä

tuloksia useissa tilanteissa, ja tämän takia on kehitetty metodeja oppimisnopeuden kertoimien suoraan säätämiseen. Nämä voidaan jakaa kahtia yhtäältä niihin metodeihin, jotka käyttävät yhtä oppimisnopeuden kerrointa kaikille kaarien painoille (*bold driver –metodi, painojen säädelty kierto, sopeutuvaan oppimisen itsemääräytyminen*), ja toisaalta niihin, jotka antavat kullekin painolle oman kertoimen (oppimisnopeuden eksponentiaalinen nousu, oppimisnopeuden eksponentiaalinen nousu ja lasku). (2) Muuttujien (virhesignaalien) koon muokkaaminen perustuu havaintoon, että kätketyn kerroksen neuronien virhesignaalit vaimenevat hyvin nopeasti. Vaimeneminen kasvaa eksponentiaalisesti samaan aikaan kun kätkettyjen kerrosten etäisyys tuloskerroksesta kasvaa lineaarisesti. (3) Eräs tapa parantaa verkon tuloksia on käyttää neuronin odotettua tulostetta sen todellisen tulosteen sijaan. (4) Neliövirheen sijasta energiafunktiona voidaan käyttää mitä tahansa virheen pienentyessä laskevaa funktiota. (5) On havaittu, että koulutettavan joukon kokoa voidaan käyttää sopivan oppimisnopeuden määrittämisessä, joten sopivat parametrit valitsemalla voidaan vastavirta-algoritmissa myös nopeuttaa oppimista. [Sarkar, 1995.]

Parannusten tarkemmasta luonteesta, ks. [Sarkar, 1995; Duda et al., 2001; Pernía-Espinoza et al., 2005].

4. Nykyhetken ja tulevaisuuden opetusmalleja

Neuroverkko on eräs teollisuudessa eniten hyödynnetyistä keinoälyn työkaluista, ja monikerroksinen syöttöverkko on näissä tehtävissä käytetyin verkkoarkkitehtuuri, koska sitä voidaan pitää mihin tahansa funktioon soveltuvana approksimaatiovälineenä [Pernía-Espinoza et al., 2005]. Teollisuuden ja sotateollisuuden tarpeet asettavat neuroverkkojen suorituskyvylle suuria tarpeita, joita täyttämään on kehittynyt vastavirta-algoritmia monipuolisempia, nopeampia ja huomattavasti luotettavampia opetusmenetelmiä. Tässä luvussa esitellään joitakin tärkeimmistä opetuksen suuntauksista. Tärkeimpinä lähteinä toimivat Martí and El-Fallahi [2004] ja Oh and Pedrycz [2005].

4.1. Useita aloituksia käyttävät menetelmät

Useita aloituksia käyttävät opetusmenetelmät (multi start methods) ovat tulleet neuroverkkolaskennassa viime aikoina yhä suosittumiksi, ja ne ovat tällä hetkellä yksi tärkeimmistä verkkojen opetusmenetelmistä. Niissä opetus jaetaan kahteen vaiheeseen: ensimmäiseen, jossa generoidaan ratkaisu, ja toiseen, jossa ratkaisua sitten (useimmissa tapauksissa) parannetaan. Tätä prosessia iteroidaan ratkaisuavaruuden eri alueilla, joista jokainen tuottaa *paikallisen ratkaisun*,

joista paras valitaan algoritmin tulosteeksi (ks. kuva 3). [Martí and El-Fallahi, 2004.]

```
i <- 1
while (lopetusehtoa ei täytetty){
  askel 1. (generointi) {
    Generoi ratkaisu  $x_i$ 
  }
  askel 2. (haku) {
    Käytä jotakin hakumetodia  $x_i:n$ 
    parantamiseksi
    Olkoon  $x_i'$  saatu ratkaisu.
  }
  if ( $x_i'$  parempi kuin paras ratkaisu){
    päivitä paras ratkaisu
  }
  i <- i+1
}
```

Kuva 3. Useiden aloitusten metodin pseudokoodi (Martín ja El-Fallahin mukaan)

Tavanomainen useiden lähtöjen metodi generoi tasaisesti jakautuneita ratkaisuja ratkaisuavaruuteen ja aloittaa hakemisen kustakin näistä generoiduista pisteistä. Tällaisen ratkaisun tiedetään lähestyvän globaalia ratkaisua, kun pisteiden määrä lähestyy ääretöntä. Algoritmin tehoa on parannettu kehittämällä suodatinmenetelmiä, joilla generoiduista ratkaisuista valitaan vain parhaat toiseen vaiheeseen. Suodatinmenetelmistä Martí ja El-Fallahi esittelevät *etäisyysuotimen*, joka tarkistaa, ettei generoitu ratkaisu ole liian lähellä jo löydettyä paikallista optimia. *Laatusuodin* puolestaan varmistaa, että generoidut ratkaisut ovat suhteellisen korkealaatuisia estämällä hakua alkamasta pisteistä, joiden arvo on korkeampi kuin tietty kynnyсарvo (QTH), jota säädetään hakujen edetessä pohjautuen jo tehdyistä hauista kerättyyn tilastolliseen tietoon. [Martí and El-Fallahi, 2004.]

4.2. Tabu-haut

Sextonin ja muiden [1998] neuroverkoille kehittämä *tabu-haku* (tabu search, TS) perustuu satunnaiseen hakuun. Tässä haussa *ympäristöt* määritellään tasajakau-
masta satunnaisesti valituille pisteille, ja ympäristö on alue, joka rajataan

$\pm 0.1\%$:iin kullekin painolle senhetkisessä ratkaisussa. Tällä alueella generoidaan viisisataa satunnaista ratkaisua ja paras niistä valitaan.

Ensimmäinen ratkaisu generoidaan automaattisesti välille $[-10,10]$ ja sen ympäristöä tutkitaan. Paras ratkaisu tässä ympäristössä syrjäyttää alkuperäisen ratkaisun ja prosessi toistetaan. Lisäämällä viimeisin ratkaisu listan alkuun ja hylkäämällä vanhin ratkaisu luodaan *tabu-lista*. Uusi ratkaisu kuitenkin hylätään, jos kaikki sen painot ovat $\pm 0.01\%$:n päässä listan minkä tahansa yhden ratkaisun painoista, eli vanhoja jo kuljettuja ratkaisuja vältetään. (Ks. myös: [Martí and El-Fallahi, 2004].)

Laajennettu tabu-haku (extended tabu search) toimii seuraavasti: ensimmäinen ratkaisu x_0 generoidaan jakauman välille $[-10,10]$ ja sen hetkinen paras ratkaisu x_{paras} asetetaan x_0 :aan. Tällä välillä generoidaan ratkaisuja annetun iteraatiomäärän ajan. Kun uusi piste x_{uusi} generoidaan, tarkistetaan tietyt ehdot. Mikäli tämä piste on parempi kuin x_{paras} eli $f(x_{\text{uusi}}) < f(x_{\text{paras}})$, tämä piste hyväksytään ja x_{paras} eli $f(x_{\text{paras}})$ päivitetään. Muussa tapauksessa tabu-ehdot tarkistetaan. Jos tabu-listassa on tietyt läheisyys ehdot täyttävä ratkaisu, koko tarkistus tehdään tälle pisteelle ja pisteelle x_{uusi} , muuten piste hyväksytään. Tätä jatketaan tuhannen hyväksytyt ratkaisun ajan, minkä jälkeen uusi tuhannen iteraation sykli alkaa. Näin jatkuu niin pitkään kuin $f(x_{\text{paras}})$ paranee.

Kun satunnaisten pisteiden valinta loppuu, suoritetaan *vahvistamisvaihe* (intensification phase), jossa suoritetaan haku, joka lähtee liikkeelle parhaasta ratkaisusta. Uudet pisteet luodaan muuttamalla pistettä x_{paras} pienen *askel-arvon* sisään sijoittuvalla satunnaisella muutoksella

$$\text{askel} = ((0.1x_{\text{paras}}) - (0.2x_{\text{paras}}) \times \text{satunnaisluku}) / \text{muutos} .$$

Jokainen vahvistamisvaiheen sykli luo tuhat uutta pistettä. Tämä vaihe kestää maksimissaan kaksikymmentä sykliä, niin pitkään kuin pisteessä x_{paras} tapahtuu yksikin pienennös. Vahvistamisvaiheen jälkeen alkaa *monimuotoistusvaihe*, jonka tarkoituksena on laajentaa hakualuetta. Nyt askel-arvo lasketaan seuraavasti:

$$\text{askel} = ((0.1x_{\text{paras}}) - (0.2x_{\text{paras}}) \times \text{satunnaisluku}) \times \text{muutos}$$

Monimuotoistusvaihe luo uusia pisteitä muuntelemalla pistettä x_{paras} tällä askel-arvolla. Iteraatiiosykliä suoritetaan samoin tuhannen iteraation sarjoissa samoin kuin vahvistamisvaiheessa. Vahvistamis- ja monimuotoistusvaiheita vuorotellaan enintään viiden toiston ajan. Kaavassa *satunnaisluku* annetaan väliltä $[0,1]$ ja *muutos* on vahvistamisvaiheiden syklejä mittaava arvosta 1 lähtevä indeksoija.

Martín ja El-Fallahin [2004] kehittämä tabu-haun variaatio kehittää edelleen laajennettua tabu-hakua. Tämä tabu-haku koostuu kolmesta vaiheesta (alustus, vahvistaminen ja monimuotoistus), mutta toisin kuin laajennettu tabu-haku, se käyttää alustusvaiheessa *simplexin optimoijaa* (Simplex optimizer) suodattamaan

luomiaan satunnaisia pisteitä, ja edelleen vahvistamis-vaiheessa vastavirta-algoritmin sukuista osittaisiin derivaattoihin perustuvaa vahvistusproseduuria pelkän satunnaisen variaation sijasta. Seuraavassa taulukossa Martí ja El-Fallahi dokumentoivat eri tabuhakujen sekä simplex-haun (SS) tuloksia laajalle 45 eri tyyppien funktiota sisältävälle testijoukolle F:

MetaHeuristics over F					
Methods	Train error			Avg. CPU	Test error
	Avg.	Min.	Max.		Avg.
B-TS	142.67	0.00	1528.38	19.64	180.03
E-TS	94.56	0.00	1173.16	287.96	124.27
SS	7.62	0.002	65.39	278.03	57.66
TS	8.65	0.00	47.71	254.00	55.30

Kuva 4. Tabu-hakujen tulosten vertailua tietylle testijoukolle (Martín ja El-Fallahin mukaan)

4.3. Geneettiset algoritmit ja sumeat polynomiset neuroverkot

Oh ja Pedrycz [2005] esittelevät uudentyyppisen variaation neuroverkkoarkkitehtuureihin, joita kutsutaan *sumeiksi polynomisiksi neuroverkoiksi* (fuzzy polynomial neural network, FPNN). Ohn ja Pedryczin verkko hyödyntää geneettisen optimoinnin ja erityisesti geneettisten algoritmien (GA) metodiikkaa toiminnassaan. Sen erityispiirre aikaisempiin monikerroksisiin sumeihin neuroverkkomalleihin nähden on sen rakenteellisesti optimoitu rakenne, joka joustaa enemmän kuin aikaisemmat FPNN-sovellukset. [Oh and Pedrycz, 2005.] Geneettistä optimointia käytetään verkon rakenteen optimoinnissa, kun taas parametrien optimoinnissa hyödynnetään tavallista vastavirta-algoritmin kehittyneempää muotoa (least square method). Vaikka parametrien optimointi onkin tästä syystä tehottomampaa kuin uusissa hakumenetelmissä kuten tabuhaussa ja useiden lähtöjen menetelmissä, väittävät Oh ja Pedrycz, että verkon suoritukset ovat sen optimoidun luonteen takia parempia kuin nykyisissä kilpailevissa menetelmissä [Oh and Pedrycz, 2005]. Utta mallia kutsutaan *GA-pohjaiseksi sumeaksi neuroverkoksi* (GA-based FPNN, gFPNN).

Geneettiset algoritmit on stokastinen tekniikka, joka perustuu evoluution, luonnonvalinnan ja geneettisen rekombinaation periaatteille simuloiden käsiteltävään ongelmaan sopivimpien yksilöiden henkiin jääntä muodostettujen potentiaalisten ratkaisujen populaatiossa. Geneettiset algoritmit pystyvät tutkimaan ratkaisuvastavuuden globaalisti seuraten hedelmällisiä uria, kun samaan aikaan generoituu satunnaisia pisteitä, jotka pienentävät algoritmin paikalliseen optimiin kiinni jäämisen riskiä.[Oh and Pedrycz, 2005] Geneettisten

algoritmien pääpiirteet koskevat yksilöitä (merkki)jonoina käsiteltyinä, populaatio-pohjaista optimointia, joka käyttää genotyyppi-avaruudessa tapahtuvaa hakua, ja stokastista hakumetodia, kuten valintaa ja risteytymistä. [Oh and Pedrycz, 2005.]

Oh ja Pedrycz käyttävät geneettistä algoritmia optimoimaan FPNN-verkon tiettyjä olennaisia ominaisuuksia, kuten syötöneuronien määrää, verkon neuronien polynomisen aktivaatiofunktion astetta ja joukkoa syötemuuttujia, joita käytetään kussakin *sumeassa neuronissa* (fuzzy polynomial neuron, FPN). [Oh and Pedrycz, 2005.] (Ks. myös [Bechtel and Abrahamsen, 2002].)

5. Konnektionismin tieteenfilosofisia implikaatioita

Kuten johdannossa todettiin, konnektionismi on vaikuttanut voimakkaasti useiden tieteenalojen kuten tietojenkäsittelytieteen, filosofian, psykologian, neurotieteiden ja kielitieteen teorianmuodostukseen. Se on näyttänyt tarjoavan biologiaan ja psykologiaan mutta myös laskennallisuuteen kytkeytyvän vaihtoehdon perinteisille tavoille hahmottaa aivojen toimintaa ja kielellisiä ja kulttuurisia kysymyksiä. Rumelhartin ja McClellandin esittelemät monikerroksisten neuroverkkojen ominaisuudet hahmontunnistus- ja luokittelutehtävissä ovat antaneet syytä odottaa paljon konnektionismin panokselta sovellustieteissä. Vastavirta-algoritmilla on ollut tärkeä rooli tässä kehityksessä, sillä se on tarjonnut ymmärrettävän ja intuitiivisesti houkuttelevan metaforan ihmisoppimisen ja ajattelun hahmottamiseen ja tutkimukseen. Konnektionismia yksinkertaisimmassa muodossaan tai muodossa, joka sille helposti ymmärretään, on kuitenkin kritisoitu aiheellisesti useista näkökulmista, joista kuvailen tässä joitakin (laajempi käsittely teoksessa: [Bechtel and Abrahamsen, 2002]; ks. myös tämän tutkielman alaviite 1 sivulla 1).

Ensinnäkin konnektionismin hyödyntämän "paralleelilaskennan" vastavuutta aivojen toimintaan on kritisoitu useilta kohdin. Tämä kritiikki voidaan edelleen jakaa useihin alakohtiin, kuten kritiikki (1) neuroverkkojen suhteellisesta epädynaamisuudesta aivoihin verrattuna, jossa ei ole huomioitu uusien dendriittien muodostumista aivosolujen välille. Geneettisten algoritmien kehitys näyttäisi osittain vastaavan tähän kritiikkiin.(2) hermosolujen emission luonne ei ole diskreettiä ja sykleissä tapahtuvaa, kuten neuronien toiminta perinteisissä neuroverkkomalleissa, vaan koostuu tietyllä frekvenssillä lähetetyistä impulsseista, joiden muoto mukalee jonkin verran normaalijakaumaa. Vaikka neuroverkot ovatkin toimintatavoiltaan epäilemättä lähempänä aivojen tietojenkäsittelyä kuin perinteisen symbolisen metaforan varassa rakennetut mallit aivojen toiminnasta, muotoilee tämä kritiikki erään olennaisen piirteen vaarasta, johon konnektionismi on ollut vaarassa ajautua syntyajoistaan lähtien.

Konnektionismi on kehittynyt tiukassa yhteydessä tietojenkäsittelyn avulla toteutettuihin käytännön malleihin, joista on ollut suurta apua teorianmuodostuksessa. Toisaalta näiden mallien hyödyntäminen on kriitikittömänä aina vaarassa johtaa yksinkertaistavaan ajatteluun muotoa "jos se toimii, malli on oikea".

Toiseksi Rumelhartin ja McClellandin esittämistä tuloksista on löydetty runsaasti aukkoja, ja esimerkiksi Rumelhartin ja McClellandin teoria englannin kielenoppimisesta, vaikka osoittaakin vastavirta-algoritmin ja monikerroksisten neuroverkkojen voiman englannin muotojen pitkän aikavälin opetuksessa, ei edes pyri millään tapaa selittämään sellaisia systemaattisia epäonnistumisia kielenoppimisen yksityistapauksissa, joita tuloksissa esiintyy. Esimerkiksi Clancey [1997] kritisoikin "yksinkertaista konnektionismia" siitä, että se ei ota huomioon oppimisen tilanteista, ympäristöön kytkeytyvää luonnetta tai osuutta, joka aikaisemmillä skeemoilla ja tietorakenteilla oppimisprosessissa on; monikerroksista neuroverkkoa voidaan pitää tietynlaisena toiminnaltaan melko yksinkertaisena oppimiskoneena, joka suoriutuu vain tietyllä tapaa rakennetuista sitä varten suunnitelluista tehtävistä, joissa sille lisäksi syötetään valmiiksi manipuloitua dataa. Uudempi neuroverkkojen teoria ja erityisesti "voimakkaat" oppimismetodit, jotka pyrkivät takaaman verkon oikean oppimisen sellaisilla datajoukoilla, jotka sisältävät satunnaista hälyä, sekä neuro- ja kognitiotutkimuksen yhteydessä kehittyvä käytännön tieto erikoistuneiden hermoverkkojen toiminnasta näyttäisivät osittain vastaavan tähän kritiikkiin.

Kolmanneksi, kuten tässä tutkielmassakin on esitetty, on vastavirta-algoritmi suhteellisen tehoton ja epävarma oppimismenetelmä. Se ei esimerkiksi pysty mitenkään selittämään aivotutkimuksen runsaita ennakkotapauksia, joissa yksi altistus tietylle tapahtumalle riittää voimakkaan oppimiskokemuksen synnyttämiseen. Kehitetyt edistyneemmät neuroverkkojen opetusmenetelmät ja erityisesti online-opetuksen kannoilla kehitetyt metodit, jotka perustuvat rajoitettuun aikaan oppia kukin datan kuvio (jotka lisäksi esitetään neuroverkolle vain kerran) viittaavat kuitenkin siihen, että konnektionistisilla menetelmillä voi olla todellista selitysvoimaa myös tällaisiin aitoihin oppimisongelmiin sovellettuna. Nämä menetelmät hyödyntävät usein sellaisia edistyneitä ongelmanratkaisun menetelmiä, jotka perustuvat arvauksiin, vaihtoehtojen karsintaan ja kehittämiseen sekä valintaan niiden väliltä, kuten useiden aloitusten menetelmät. Näyttääkin siltä, että konnektionistisessa teorianmuodostuksessa joudutaan yhä enemmän luopumaan yksinkertaisista neuroverkkoalgoritmien sovelluksista ja käsittelemään erilaisia oppimistilanteita ja kognition muotoja pitkälle erikoistuneina tietojenkäsittelytehtävinä, joiden rakenteeseen ei ole olemassa universaaleja ratkaisuja. Pikemminkin niitä on mallinnettava hienopiirteisillä

pitkälle erikoistuneilla malleilla. Näiden mallien todelliset yhteydet aivojen tapaan käsitellä tietoa ratkaisevat niiden soveltuvuuden kognitiontutkimuksen lopulliseen teoriaan, ja tämän takia odottelemme vielä kiinnostuneina todella paralleelien tietokone- ja neuroverkkoarkkitehtuurien kehitystä.

6. Yhteenveto

Tämä tutkielma loi katsauksen monikerroksisten neuroverkkojen arkkitehtuuriin ja erityisesti niiden opetusmenetelmiin, sekä keskusteli lyhyesti konnektionismin tieteenfilosofisista implikaatioista. Tärkeitä tuloksia olivat erityisesti huomiot siitä, millä tapaa uudemmat geneettisiin algoritmeihin ja edistyneisiin hakumenetelmiin perustuvat opetusmenetelmät saattavat vastata perinteisiin konnektionismia kohtaan esitettyyn kritiikkiin.

Viiteluettelo

- [Bayro-Corrochano et al., 2005] Eduardo Bayro-Corrochano, Refugio Vallejo and Nancy Arana-Daniel, Geometric preprocessing, geometric feedforward neural networks and Clifford support vector machines for visual learning. *Neurocomputing* **67** (2005), 54-105.
- [Bechtel and Abrahamsen, 2002] William Bechtel and Adele Abrahamsen, *Connectionism and the Mind. Second Edition*. Basil Blackwell, Cambridge, Massachusetts, 2002.
- [Clancey 1997] William J. Clancey, *Situated Cognition*. Cambridge University Press, Cambridge, 1997.
- [Duda et al., 2001] Richard O. Duda, Peter E. Hart and David G. Stork, *Pattern Classification*. John Wiley & Sons, Inc., New York, 2001.
- [Han Kim and Roche, 1995] Jeong Han Kim and James R. Roche, On the optimal capacity of binary neural networks: rigorous combinatorial approaches. In: *Proc. of the Eighth Annual Conference on Computational Learning Theory 1994*, 240-249.
- [Harley, 2001] Trevor Harley, *The Psychology of Language: from Data to Theory*. Psychology Press LTD, Hove, 2001.
- [Jordan and Bishop, 1996] Michael I. Jordan and David I. Bishop, Neural network. *ACM Computing Surveys* **28**, 1 (1996), 73-75.
- [Koiran, 1994] Pascal Koiran, Efficient learning of continuous neural network. In: *Proc. of the Seventh Annual Conference on Computational Learning Theory 1994*, 348-355.
- [Kwon et al., 1994] Kyung-Hoon Kwon, Kukjin Kang and Jong-Hoon Oh, Generalization in partially connected layered neural network. In: *Proc. of*

- the Seventh Annual Conference on Computational Learning Theory 1994*, 356-361.
- [MacIntyre and Sontag, 1993] Angus MacIntyre and Eduardo D. Sontag, Finiteness results for sigmoidal “neural” network. In: *Proc. of the 25th Annual ACM Symposium on Theory of Computing 1993*, 325-334.
- [Martí and El-Fallahi, 2004] Rafael Martí and Abdellah El-Fallahi, Multilayer neural networks: an experimental evaluation of on-line training methods. *Computers & Operations Research* **31** (2004), 1491–1513.
- [Oh and Pedrycz, 2005] Sung-Kwun Oh and Witold Pedrycz, A new approach to self-organizing fuzzy polynomial neural networks guided by genetic optimization. *Physics Letters A* **345** (2005), 88-100.
- [Pernía-Espinoza et al., 2005] Alpha V. Pernía-Espinoza, Joaquín B. Ordieres-Mere, Francisco J. Martínez-de-Pisón and Ana González-Marcos, TAO-robust backpropagation learning algorithm. *Neural Networks* **18** (2005), 191-204.
- [Sarkar, 1995] Dilip Sarkar, Methods to speed up error back-propagation learning algorithm. *ACM Computing Surveys* **27** (1995), 519-542.
- [Sexton et al., 1998] R. S. Sexton, B. Alidaee, R. E. Dorsey and J. D. Johnson, Global optimization for artificial neural networks: a tabu search application. *European Journal of Operational Research* **106** (1998), 570–584.
- [Tosic, 2004] Predrag T. Tosic, A perspective on the future of massively parallel computing. In: *Proc. of the 1st Conference on Computing Frontiers 2004*, 488-502.