

Erkki Mäkinen (toim.)

**Pieniä tietojenkäsittely-
tieteellisiä tutkimuksia**

Talvi 2006



TIETOJENKÄSITTELYTIETEIDEN LAITOS
TAMPEREEN YLIOPISTO

D-2006-2

TAMPERE 2006

TAMPEREEN YLIOPISTO
TIETOJENKÄSITTELYTIETEIDEN LAITOS
JULKAISUSARJA D – VERKKOJULKAISUT
D-2006-2, HELMIKUU 2006

Erkki Mäkinen (toim.)

**Pieniä tietojenkäsittelytieteellisiä
tutkimuksia**

Talvi 2006

TIETOJENKÄSITTELYTIETEIDEN LAITOS
33014 TAMPEREEN YLIOPISTO

ISBN 951-44-6578-4
ISSN 1795-4274

Sisällysluettelo

Toiminnanohjausjärjestelmän käyttöönottoprojekti ja sen suunnittelu.....1 <i>Antti Finni</i>	1
Mainonta WWW-sivuilla - tavat ja tekniikat.....14 <i>Juha Heinonen</i>	14
Aikataulusuunnittelu ja sen haasteet projektityössä.....28 <i>Heli Helminen</i>	28
Ytimet tietokoneissa.....52 <i>Ville Niemi & Tuukka Pasanen</i>	52
Multimodaalisuus käyttöliittymissä.....71 <i>Juuso Näsi</i>	71
Opetuksellinen läsnäolo ja opettajan rooli verkko-opetuksessa.....84 <i>Suvi Peltomäki</i>	84
Web Services -teknologian käyttö tietojärjestelmäintegraatioon toiminnanoh- jausjärjestelmissä.....99 <i>Jari Rajanen</i>	99
Shakkimoottorin pelipuun perusteita.....109 <i>Jon Sahlberg</i>	109

Toiminnanohjausjärjestelmän käyttöönottoprojekti ja sen suunnittelu

Antti Finni

Tiivistelmä

Tutkielma käsittelee toiminnanohjausjärjestelmien käyttöönottoa ja niiden vaikutuksia liiketoimintaprosesseihin, sekä käyttöönotossa ja suunnittelussa huomioitavia asioita. Toiminnanohjausjärjestelmät ovat nykyään yleistyneet nopeasti suurten organisaatioiden lisäksi myös pienissä ja keskisuurissa yrityksissä. Käyttöönotto ja projektin suunnittelu ovat kriittisiä vaiheita onnistuneessa implementaatioissa, koska järjestelmästä saatavat hyödyt riippuvat paljon siitä, miten järjestelmä on otettu huomioon yrityksen strategiassa ja liiketoimintaprosessien suunnittelussa.

Avainsanat ja sanonnat: Toiminnanohjaus, ERP, ES, käyttöönotto, liiketoimintaprosessit.

CR-luokat: D.2.1, H.4.1, H.4.2, K.6.1

1. Johdanto

Toiminnanohjausjärjestelmät juontavat juurensa 1970-luvulle, jolloin ne olivat vielä hyvin toimintokohtaisia, esimerkiksi varastonhallintaan liittyviä ohjelmia. Vähän myöhemmin suurten yritysten tietojärjestelmäprojektit alkoivat sulauttaa näitä eri prosesseihin tarkoitettuja ohjelmia suuremmiksi kokonaisuuksiksi, ja 1980-luvulla oli jo useita kaupallisia toiminnanohjausjärjestelmien toimittajia. 1990-luvulla suurten yritysten ”erppi-buumi” oli kuumimmillaan, ja viimeistään nyt 2000-luvun puolella myös PK-yritykset ovat väistämättä joutuneet miettimään ERP-projekteja. ERP-järjestelmän käyttöönotto ei ole yksiselitteinen tehostusriske yritys toimintaan, vaan siihen sisältyy myös monia riskejä.

1990-luvulta on monien mielessä vielä muutamia epäonnistuneita ERP-implementointeja, joten ERP-projekteilla ei missään nimessä ole pelkästään positiivinen maine, varsinkaan kun ottaa huomioon niihin yleensä liittyvät korkeat kustannukset. Toiminnanohjausjärjestelmien käyttöönotoista löytyy myös

monia menestystarinoita, ja myös järjestelmien toimittajat ovat eri asemassa nykyään kuin esimerkiksi 10-15 vuotta sitten, jolloin kokemuspohja oli heikompi.

Onnistuneiden ja epäonnistuneiden case-tapausten perusteella voidaan sanoa että huolellinen suunnittelu ja yrityksen toimintojen kartoitus on ERP-projek-teissa ensiarvoisen tärkeää, johtuen osaksi tietojärjestelmäprojektien tietyistä eroavaisuuksista tavallisiin projekteihin verrattuna. Tässä tutkielmassa pyritään käymään läpi niitä asioita joita tulisi ottaa huomioon toiminnanohjausjärjes-telmän suunnittelussa ja käyttöönotossa, ja että miten ne vaikuttavat yrityksen liiketoimintaprosesseihin.

2. Mitä on toiminnanohjaus

2.1 Määrittely

Toiminnanohjaus on melko pulmallinen määrittelyn kannalta. On olemassa eri-laisia, erihintaisia, eritasoisia tietojärjestelmiä, joita ko. tuotteen edustajat mielel-lään kutsuvat toiminnanohjausjärjestelmäksi. Aiemmin suuret toiminnanohjaus-järjestelmien toimittajat kuten SAP, Baan, Oracle, ja JD Edwards halusivat varata nimenomaan käyttöönsä, väheksyen pienempiä toimittajia joiden järjestelmät olivat pienempiä, mutta ehkä hyvinkin toimialakohtaisesti räätälöityjä [Mäkipää, 2002]. Toiminnanohjausjärjestelmä, eli ERP (Enterprise Resource Planning) tai nykyisin myös ES (Enterprise Systems), tarkoittaa yrityksen kokonaisvaltaista tietojärjestelmää, joka vastaa useimpiin ellei kaikkiin yrityksen tietotarpeisiin. [Davenport, 2000] Nykyään toiminnanohjausjärjestelmät tunnetaan yleisesti "Erppeinä" (ERP), mutta mielestäni Davenportin kirjassaan käyttämä termi ES kuvaa järjestelmien nykyistä kokonaisvaltaisuutta paremmin. Yhä enemmän verkottuvia ja e-busineeseen painottuvia järjestelmiä on alettu myös kutsua termillä XES (eXtended Enterprice Systems). XES on siten ehkä "seuraavan sukupolven" toiminnanohjausjärjestelmä. Toiminnanohjausjärjestelmien määrit-telemisen yksiselitteisesti on kuitenkin melko mahdotonta. ERP-järjestelmät juontavat alkunsa 1980-luvun MRP-järjestelmistä (Material Resource Planning), jotka silloin koskivat lähinnä materiaalitoimintojen suunnittelua ja hallintaa, ja olivat heikosti integroitua muihin yrityksen järjestelmiin. Nykyiset ERP-järjestelmät kuitenkin ovat paljon muutakin kuin tuotannosuunnittelusysteemi, ne kattavat enenevässä määrin kaikki yrityksen tai organisaation sisällä

tarvittavat tietojenkäsittelytarpeet, samoin kuin ulkoiset yhteydet [Mäkipää, 2002].

2.2 Case-esimerkkejä

ES-järjestelmien implementointi maksaa yritykselle muutamista tuhansista euroista satoihin miljooniin euroihin, ja lisäksi tulee vielä ylläpitokulut päälle. Onko koko ES- tai ERP-järjestelmien hankinnassa ylipäättään mitään järkeä, jos yrityksen toiminnot voidaan pitää yllä edullisellakin taloushallinnon ohjelmistolla ja office-sovelluksilla. Davenport [2003] antaa muutamia esimerkkejä ES-järjestelmien hyödyistä.

Kansainvälisen Ciscon ES-implementaatio maksoi internet-integraatioineen yli 100 miljoonaa dollaria, mutta yritys on laskenut säästävänsä järjestelmän avulla 500 miljoonaa dollaria vuositasolla. Elf Atochem North America -organisaation ES-käyttöönotto tapahtui noin kuudessa vuodessa, ja jo käyttöönoton aikana yhtiö säästi järjestelmän ansiosta yli 10 miljoonaa dollaria vuodessa. Järjestelmä oli maksanut itsensä takaisin jo ennen kuin implementaatio oli saatu valmiiksi. Elf, kuten muutkin isot yritykset saavat huomattavia säästöjä järjestelmien integroimisesta ja reaaliaikaisuudesta, ja juuri Elf laski että se säästi yli 200 miljoonaa dollaria yksistään järjestelmien integroinnilla (siis kokonaisvaltaisella toiminnanohjausjärjestelmällä) ja keskenään epäsovivien ja eri muodossa olevien myyntitilausten, tarjouspyyntöjen ym. dokumenttien eliminoimisella. On tosin olemassa myös lukuisia esimerkkejä, joissa on tehty suuria investointeja saamatta mitään hyötyjä järjestelmästä. Dell Computersin, jonka oli tarkoitus ottaa ES käyttöön maailmanlaajuisesti, Euroopassa, Amerikassa ja Aasiassa, strategiana oli standardoida ja yhdistää prosesseja, sekä tehostaa informaation saatavuutta ja läpimenoaikoja. Vaikka projektin parissa työskenteli täysipäiväisesti yli 300 henkilöä, käyttöönoton kanssa tuli kiire ja testaus jäi viimehetkeen, jolloin huomattiin vakavia puutteita suunnittelussa. Rahaa oli kulunut yli 220 miljoonaa, ja järjestelmästä jäi käyttöön ainoastaan henkilöstöhallinnan moduuli, sillä projekti keskeytettiin [Davenport, 2000]. Vaikka esimerkit olivat kansainvälisistä yrityksistä, suhteessa samankaltaisia hyötyjä on mahdollista saavuttaa myös pienissä ja keskisuurissa yrityksissä.

3. Lähtökohdat

3.1 Miksi mukaan ES-projektiin?

Suurimmat yritykset ovat jo ottaneet käyttöön jonkin toiminnanohjausjärjestelmän. Tällä hetkellä on paljon pieniä ja keskisuuria yrityksiä jotka ovat korvaamassa alkuperäisiä yksittäisten toimittajien järjestelmiä yhtenäisellä toiminnanohjausjärjestelmällä. Alkuperäiset ohjelmistot ovat ehkä tulleet elinkaarensa päähän ja uuden järjestelmän implementointi on toiminnan jatkamisen kannalta pakon sanelemaa. Mutta on myös yrityksiä, jotka lähtevät hakemaan tehokkuutta ja kilpailuetua tarkoitushakuisesti juuri ES-järjestelmistä. Kehitettävälle järjestelmälle voidaan asettaa monenlaisia tavoitteita, joista Ruohonen [2003] listaa mm. seuraavia:

- Tuottavuuden parantaminen
- Hallintokustannusten vähentäminen
- Liiketahtumien kasvaneesta lukumäärästä selviytyminen
- Kuormitushuippujen tasoittaminen
- Tarkkuuden parantaminen esimerkiksi toimitusajoissa
- Asiakaspalvelun parantaminen
- Kassavirran ja varainhallinnan parantaminen
- Johdon raportoinnin ja valvonnan kehittäminen
- Pääomakierron nopeuttaminen
- Nopeamman tiedonvälityksen aikaansaaminen
- Lyhyemmät läpimenoajat
- Toimitusten nopeuttaminen
- Laadun parantaminen.

3.2 Tavoitteet ja kilpailuedut

Kilpailuedun luominen ES-järjestelmällä ei ole vain järjestelmän ominaisuuksien ansiota ja niiden muokkausta, vaan kilpailuetu tulee toimintatavoista, jotka ei välttämättä riipu tietojärjestelmästä. Davenport [2003] esittää muutamia hyviä kysymyksiä mietittäväksi kilpailueduista:

- Mitkä ovat yrityksen / organisaation nykyiset kilpailuedut? Miten tuleva järjestelmä vaikuttaa niihin? Jos esimerkiksi asiakaspalvelu on yrityksen yksi kilpailuvalteista, aiheuttaako ES-järjestelmä siihen muutoksia, ja jos niin huonompaan vai parempaan suuntaan?

- Mahdollistaako ES-järjestelmä uusia strategisia toimintoja jotka voivat olla tarpeellisia tulevaisuudessa? Esimerkiksi, mahdollistaako järjestelmä Lean-tuotannon, ja onko se tarpeen pysyäksemme markkinoilla?
- Miten ES-järjestelmän hinta vaikuttaa nykyiseen tuotteiden ja palvelujen hinnoittelustrategiaan? Jos nyt noudatetaan alhaista kustannusstrategiaa, onko se mahdollista vielä järjestelmän käyttöönoton jälkeen?
- Mitkä muut kilpailijat toimialalla ovat ottamassa käyttöön ES-järjestelmää? Miten se tulee todennäköisesti vaikuttamaan heidän heikkouksiin ja vahvuuksiin? Mitkä tulevat olemaan kilpailuedut jos kaikki ottavat järjestelmät käyttöön? Mitkä toimijat saattavat epäonnistua projekteissa, ja mitkä onnistua?
- Jos muut toimialan yritykset ottavat toiminnanohjauksen käyttöön, onko olemassa integroitumistarvetta heidän järjestelmiinsä tulevaisuudessa? Jos yhteistyötä tehdään jo nyt tuleeko kommunikointi sähköistymään ja entistä automaattisemmaksi?
- Onko olemassa jokin näkökulma liiketoiminnassa, josta on haitallista käyttää toimialan yleisintä tuotetta? Olisiko parempi käyttää jotain muuta? Jos on, niin voiko sen erottaa muusta ES-järjestelmästä?
- Häiritseekö ES-järjestelmä tai sen käyttöönotto yrityksen ydintoimintoja tai estääkö se yrityksen toiminnan?

3.3 Käyttöönottosuunnitelma

Eräs olennainen osa toiminnanohjausjärjestelmän kartoituksessa on sen käyttöönottosuunnitelma. Siinä suunnitellaan käyttöönoton aikataulut ja implementointi tapa. Implementoinnin suorittamiseen on kaksi päätapaa, eli ns. ”kerta-räjäys”, jossa uusi järjestelmä otetaan kerralla käyttöön kokonaisuudessaan ja samalla vanha järjestelmä hylätään, tai sitten vaiheittainen käyttöönotto, jossa vanhaa järjestelmään pidetään yllä uuden rinnalla. Muutamia muita malleja näiden väliltä on prosessikohtainen siirtyminen uuteen järjestelmään, sekä toiminto- tai toimialuekohtainen käyttöönotto. Kaikissa mainituissa implementointityyleissä on puolensa, ja siksi yrityksen tuleekin suunnitella miten minimoida käyttöönotosta aiheutuvat haitat. [Davenport, 2003; Huigang and Yajiong, 2004]

3.4 Vaikutukset

Kuten jo aiemmin mainittiin ES-projekti saattaa olla yritykselle kallis projekti, joillekin PK-yrityksille ehkä kaikkien aikojen suurin projekti. Erityisesti alhaisen

kustannusstrategian yrityksissä voi olla vaikeuksia hyväksyä järjestelmän aiheuttamia kustannuksia, vaikka siitä saatavat hyödyt kompensoisivatkin ne. Vaikka järjestelmän suunnittelussa ja toteutuksessa on paljon ennalta-arvaamattomia sudenkuoppia, useimmiten onnistuneesti implementoitu järjestelmä antaa hyvän pohjan pärjätä kilpailuympäristössä. [Botta-Genoulaz and Millet 2005]

ES-järjestelmä itsessään ei tuota kilpailuetua, vaan kilpailuetu aiheutuu siitä että järjestelmää hyödynnetään paremmin tai erilalla kuin kilpailijat. Periaatteessa kaikilla organisaatioilla on mahdollisuus ottaa käyttöön toiminnanohjausjärjestelmä, eli ohjelmiston muokkaaminen omaan käyttöön on avainasia. [Davenport, 2000]

4. Järjestelmän suunnittelu ja toteutus

4.1 Suunnittelun tärkeys

Toiminnanohjausjärjestelmän onnistunut käyttöönotto edellyttää huolellista suunnitteluprosessia. Suunnitteluprosessin tärkeys korostuu, mitä isommasta organisaatiosta on kyse, ja mitä monimutkaisempi järjestelmä itsessään on. [Huigang and Yajiong, 2004] Suunnittelun alussa määritellään mitä toimintoja järjestelmältä odotetaan, ja mitä osa-alueita toiminnanohjaus tulee kattamaan. Kokonaisuudessaan suunnittelun tehtävänä on kattaa kuvaukset [Ruuhonen ja Salmela, 2003]:

- tuettavista liiketoimintaprosesseista
- järjestelmän suorittamista tiedonkäsittelytehtävistä
- järjestelmään tallennettavista tiedoista sekä
- tiedon esittämistavoista ja raporteista.

Kuvauksiin liittyy myös useita ominaisuuksia liittyen lopputuotteeseen, kuten tietovirtojen kuvauksia, käyttöliittymän suunnittelua ja toteutusta sekä tietokantojen suunnittelua. Kaikissa tuotteissa kaikki edellä mainitut ei välttämättä ole mahdollisia, jos kyse on ns. pakettiratkaisusta, mutta nykyään räätälöinti on vahva edellytys liiketoimintaedun saavuttamiseksi toiminnanohjausjärjestelmän avulla.

Toiminnanohjausjärjestelmän suunnittelu- ja käyttöönottoprojekti eroaa perinteisestä projektityötavasta muutamilta kohtaa. Esimerkiksi työn lähtökohdat ovat usein epämääräiset koska täsmällistä tavoitetta tai "rakennuspiirustuksia"

ei ole. Tarpeet on määriteltävä laajan käyttäjäkunnan kanssa koska ei ole yhtä ainoaa tekijää joka kertoisi mitä tarvitaan, ja kommunikaatio saattaa olla hankalaa johtuen useiden eri alojen henkilöstöryhmien mukanaolosta. Samoin työn sisällön sinetöiminen voi olla hankalaa koska sopimukset ja dokumentit usein väljiä, ja niitä voi tulkita eri tavoin. Lisäksi järjestelmälle voi olla vaikea saada hyväksyntää etukäteen, sillä ratkaisut ymmärretään vasta, kun ne nähdään käytännössä. Samoin muutosten hallinta on vaikeaa, koska työn aikana opitaan uutta, ja siitä johtuen ratkaisut voivat muuttua. [Forsman, 1995]

4.2 Strategia

Tietojärjestelmäprojektin läpiviemisessä on ensiarvoisen tärkeää, että koko yrityksen strategia on selvillä. Yrityksen johdon ja projektin avainhenkilöiden tulee tiedostaa, mikä on yrityksen ydinliiketoiminta-alue, miten asiakkaalle luodaan lisäarvoa ja miten yritys erottuu kilpailijoista [Hyötyläinen ja Kalliokoski, 2001]. Toiminnanohjausjärjestelmän yksi tärkeimmistä tehtävistä on tukea yrityksen strategiaa, eli jos yritys toimii esimerkiksi tukkukaupan alalla, on järjestelmässä mahdollisuus painottaa ja räätälöidä siinä tarvittavia transaktioita, kuten varastohallintaa ja logistiikkaa. [Soh et al., 2003]

Tietohallintostrategia on osa koko yrityksen strategiaa jossa määritellään yrityksen tai organisaation pitkän aikavälin suunnitelma tietoresurssien hyväksikäytöstä ja johtamisesta, sekä toteuttamisesta. [Ruuhonen ja Salmela, 2003] Toiminnanohjausjärjestelmän perustana tulisi olla tietohallintostrategia, jossa on määritelty ainakin seuraavat osa-alueet [Kettunen, 2002]:

- tietohallinnon nykytilanne
- yrityksen tavoitteet ja tulevaisuuden muutokset
- tietohallinnon tarpeet ja visiot
- riskianalyysi
- toimintasuunnitelma tarpeiden kattamisesta
- toimenpiteiden resurssit
- varautuminen poikkeustapauksiin.

Strateginen niveltäminen, Strategic Alignment, tarkoittaa liiketoiminnan strategian, IT-strategian, organisaationaalisen infrastruktuurin, prosessien ja IT-infrastruktuurin samanaikaista huomioon ottamista, eli että ko. toiminnot tukevat toinen toisiaan. [Kataja, 2004] Tämä näkökulma on hyvä perusta tietojärjestelmäprojektille, vaikka siinä on omat heikkoutensa mm. isoissa kansainvälisissä yri-

tyksissä, joissa infrastruktuurit saattavat olla hyvinkin eritasoisia. Myöskään yrityksen strategia ei välttämättä ole hyväksi ES-järjestelmän käyttöönotolle, jos ei siinä ole huomioitu mahdollisuutta järjestelmän aiheuttamalle muutokselle. [Davenport, 2000]

4.3 Metodiset suunnittelumallit

Toiminnanohjausjärjestelmien käyttöönotossa käytetään usein muista tietojärjestelmien suunnittelumenetelmistä tuttuja metodisia lähestymistapoja suunnitteluvaiheen hallitsemiseen. [Botta-Genoulaz and Millet 2005] Tyypillisimpiä metodisia lähestymistapoja ovat vesiputousmalli eli vaihejakomalli, evoluutiomalli, spiraalimalli ja protoilu. Vaihejakomalli etenee kutakuinkin seuraavassa järjestyksessä:

1. Esikartoitus
2. Järjestelmän määrittely
3. Järjestelmän suunnittelu
4. Järjestelmän toteutus
5. Järjestelmän käyttöönotto
6. Ylläpito.

Kyseinen vaihejakomalli auttaa pitämään projektin paremmin hallinnassa ja aikataulussa, ja lisäksi se sisältää selkeät kokonaisuudet, joita on myös helppo arvioida. Mallia on tosin kritisoitu sen jäykkyyden vuoksi. Periaatteessa malli antaa olettaa, että asiakas tietää ja osaa määrittellä järjestelmänsä koskevat vaatimukset heti projektin alussa, näin ollen muutosten tekeminen kesken kehityksen saattaa olla hankalaa ja kallista. [Nykänen, 2005]

Evoluutiomalli muistuttaa vaiheiltaan vesiputousmallia, mutta sen sijaan että yritettäisiin määrittellä ja suunnitella kerralla koko järjestelmä kuntoon, edetään versio kerrallaan. Menettely takaa joustavamman räätälöinnin ohjelmistoon kehitysvaiheen aikana. Protoilu-menetelmä nojaa evoluutiomalliin, mutta tässä menetelmässä ei yritetäkään käydä kaikkia mallin vaiheita läpi, vaan saada toiminnallisuudesta käytännön kuva mahdollisimman aikaisessa vaiheessa. Tämä on varsinkin loppukäyttäjän kannalta hyvä, koska tuotteen pääsee näkemään ajoissa, jolloin muutokset ovat kaikkein joustavimmin tehtävissä ja säästytään turhalta työltä, jos jokin asia vaatiikin muokkaamista.

Spiraalimalli etenee nimensä mukaisesti spiraalimaisesti tiettyjä suunnittelu tai toteutusvaiheita toistaen kohti valmista tuotetta. Spiraalimallin vaiheet voivat olla esimerkiksi arviointi, suunnittelu, riskianalyysi ja toteuttaminen, josta taas

siirrytään arviointiin. Käytännössä harvoin sovelletaan mitään tiettyä mallia puhtasoppisesti, vaan kehitys saattaa noudattaa esimerkiksi sekä vaihejakomallia että protoilua [Nykänen, 2005].

4.4 Liiketoimintaprosessien kuvaaminen

Usein ES-järjestelmien suunnittelu aloitetaan liiketoimintaprosessien kuvaamisella. Kuvauksessa käsitellään tietyn transaktion, esimerkiksi tarjouspyynnön, suorittamisessa vaadittavat vaiheet prosessikaaviona. Tässä vaiheessa asiakkaalla on mahdollisuus vaikuttaa, kuinka prosessi kuvataan ja kuinka se toimii itse järjestelmässä. Vielä muutamia vuosia sitten oli yleinen käytäntö, että liiketoimintaprosessien uudelleen suunnittelu (BPR) toteutettiin hankittavan ES-järjestelmän ehdoilla. Tällaisella toteutustavalla ei tietenkään pystytä saavuttamaan, ainakaan pitkäaikaista liiketoimintaetua, koska prosessit toimivat tällöin kaikilla lähes samalla tavalla. Nykyään BPR pyritään toteuttamaan asiakkaan tavoitteiden ja toimintamallien mukaan, jolloin asiakas ei ole pakotettu tietyn tyyppisiin perusratkaisuihin. [Huigang and Yajiong, 2004]

4.5 Tietovirtojen kuvaaminen

Liiketoimintaprosessien määrittämisen jälkeen kuvataan itse ERP-systeemin toimintaa. Toiminta kuvataan yleensä tietovirtojen avulla, eli tietovirtakaaviolla, ne esittävät kuinka järjestelmän käyttäjiltä, tietokannoista tai ulkoisista lähteistä tuleva tieto prosessoidaan, tallennetaan ja yhdistellään. Tietovirtakaaviot antavat siis hyvän kokonaiskuvan koko systeemin toiminnasta, ja auttaa sekä asiakasta että sovelluskehittäjää ymmärtämään järjestelmän toimintaa. Samoin kuin prosessikaaviot, voivat tietovirtakaaviot olla hierarkkisia, eli toiminta voidaan kuvata ensin karkeasti, ja sitä voidaan tarkentaa syvemmillä tasoilla. [Ruohonen ja Salmela, 2003]

4.6 Tietokantojen suunnittelu

Tietokantojen suunnittelu on osa perinteistä tietojärjestelmäsuunnittelua, mutta useat, varsinkin raskaammat, toiminnanohjausjärjestelmät käyttävät pitkälle vakioituja tietokantarakenteita. Pienemmissä ja alusta asti rakennetuissa järjestelmissä tietokantojen suunnittelu on tärkeä osa kokonaisuutta, mutta en puutu siihen tässä tutkielmassa, koska relaatioiden suunnittelu toiminnanohjausjärjestelmän kaltaisessa on iso ja monimutkainen projekti. Lisäksi monet kaupalliset tuotteet ovat tietokantojen osalta jo niin suunniteltuja, ettei niihin kovin

helposti puututa. Niissä räätälöinti tietokantojen osalta tarkoittaa lähinnä tietokannan hallintajärjestelmän optimointia ja muokkaamista. Isoissa valmiissa järjestelmissä tietokantojen muuttaminen toisi todennäköisesti enemmän ongelmia ja lisätyötä kuin hyötyjä. Monissa järjestelmissä kantojen rakenteen muuttaminen onkin estetty. [Ruohonen ja Salmela, 2003]

5. Toiminnanohjausjärjestelmän käyttöönotto

5.1 Käyttöönotto

Toiminnanohjausjärjestelmän käyttöönotto fyysisessä ja teknillisessä mielessä ei välttämättä ole mikään yrityksen suurin ponnistus, vaikkakin IT-järjestelmän infrastruktuurin tulee olla järjestelmälle suunnitellulla tasolla, ja hyvin asennettu. Myös vanhojen tietokantojen ja tiedostojen konvertointi uuden järjestelmän vaatimaan muotoon voi olla työläs prosessi, joka itsessään vaatii huolellista suunnittelua ja toteutusta. Teknistä henkilökuntaakin siis toki tarvitaan, ja heidän panoksensa järjestelmän oikeassa konfiguroinnissa on tietenkin korvaamatonta, mutta suurimmat haasteet alkavat usein vasta sitten, kun järjestelmä on installoitu. [Davenport, 2003]

5.2 Koulutus ja asenteet

Ei ole harvinaista että ES-järjestelmän käyttöönotto aiheuttaa yrityksessä muutostarainta. Työntekijät eivät välttämättä ymmärrä järjestelmän tuomia hyötyjä, eivät halua kouluttautua tai eivät ole motivoituneita käyttämään sitä. Osa saattaa epäillä omaa kompetenssiaan käyttää uutta järjestelmää ja jotkut saattavat pelätä järjestelmän aiheuttavan irtisanomisia työvoiman tarpeen vähentyessä. Järjestelmästä pitää puhua yrityksessä avoimesti jo suunnitteluvaiheessa, ja kertoa sen hyödyistä ja haitoista.

Koulutus on vielä nykyäänkin yksi suurimmista ongelmakohdista järjestelmän käyttöönotossa. Itse projekti saattaa olla hyvin suunniteltu paperilla, mutta kova kiire projektin loppuvaiheessa on usein pois koulutustunneista, ja jos asiakkaalle tulee vastustamaton halu säästää järjestelmän kustannuksissa, niin leikkaukset osuvat usein koulutukseen.

Avainryhmien koulutuksesta Davenport [2003] on koonnut seuraavat huomiot:

- IT-tuen tulee oppia järjestelmän luonne, infrastruktuuri, ylläpito ja yleensä tekniseentukeen liittyvät asiat.
- Projektien/prosessien suunnittelijoiden tulee oppia minkälaisia prosesseja järjestelmä pystyy tukemaan, ja miten järjestelmä on muokattavissa tulevaisuudessa tukemaan ko. prosesseja.
- Jokaisen käyttäjän tulee osata hyvin jokapäiväinen järjestelmän käyttö siltä osin kuin se on käyttäjän kohdalla tarpeellista. Myös omien transaktioiden vaikutukset muiden työhön järjestelmän kautta on ymmärrettävä yleisellä tasolla
- Ylemmän tason johtajien on ymmärrettävä vaikutukset yhtiön strategiaan, organisaatioon, ja prosesseihin, sekä kuinka ES-järjestelmällä luodaan kilpailuetua.

6. Tietojärjestelmähankkeen onnistuminen

6.1 Miten mitataan ES-hankkeiden onnistumista

Tietojärjestelmähankkeen onnistuminen on monitahoinen käsite, josta eri tahoilla on erilaisia käsityksiä. Tietojärjestelmähankkeen onnistumista voidaan kuvata ja mitata useista eri näkökulmista. Seuraavassa on esitetty muutamia niistä: Tietojärjestelmän tekninen laatu. Tekninen laatu kattaa järjestelmän teknilliseen puoleen liittyvät asiat, kuten tietokantakyselyjen vasteajat, käyttöliittymien ominaisuudet, ohjelmiston toimivuus ja ohjelmistovirheiden määrän. Tietojärjestelmän tuottaman informaation laatu sisältää informaation luotettavuuteen, oikeellisuuteen, reaaliaikaisuuteen ja helppokäyttöisyyteen liittyviä asioita. Vaikutukset käyttäjän työhön ja päätöksentekoon on myös tapa mitata tietojärjestelmän onnistumista. Siihen kuuluvat mm. tehtäviin käytetty aika, työn laatu ja työn kustannukset. Tietojärjestelmän vaikutukset liiketoimintaprosesseihin kattaa prosessin läpimenoajan, laadun ja kustannukset. Tietojärjestelmän vaikutukset yrityksen kilpailukykyyn, eli toiminnan tehostumisen tuoma kilpailuetu, tai uuden liiketoimintamallin luominen. [Ruohonen ja Salmela, 2003]

6.2 Projektin päättäminen ja arviointi

ES-projekti ei saa jäädä loputtomaan käyttöönottovaiheeseen, vaan se tulee päättää suunnitellusti ja aikataulun mukaisesti. Järjestelmää voidaan kyllä jatkossakin kehittää, ja sen tuleekin olla suunnitellulla tasolla, mutta järjestelmän

käyttöönotto konkretisoituu paremmin henkilökunnallekin kun se otetaan virallisesti käyttöön. Projektin päättämiseen liittyy usein myös sopimusteknisiä asioita. [Hyötyläinen ja Kalliokoski, 2001]

Edellisessä kappaleessa mainitun järjestelmän arvioinnin asiakasyritys tekee pitkällä aikavälillä omien resurssien voimilla, ja varmasti yrityksen johto ja taloushallinto ovat hyvinkin kiinnostuneita niin tuottavuudesta ja tehokkuudesta kuin kustannussäästöistäkin, mutta yleensä järjestelmän toimivuus tarkastetaan myös toimittajan kanssa. Ylläpitosopimukset on yleensä tehty jo samassa vaiheessa kun päätökset itse järjestelmästäkin, ja toimittaja jatkaa yhteistyötä asiakkaan kanssa ylläpidossa, jos niin on sovittu. [Davenport, 2000]

7. Yhteenveto

Toiminnanohjausjärjestelmien rooli yrityksen strategiassa tulee varmasti kasvamaan tulevaisuudessa kaikilla toimialoilla sitä mukaa kun prosessit ja ES-järjestelmät kehittyvät. 1990-luvun ”ERP-hypetys”, ja siitä seuranneet monet epäonnistumiset pilasivat järjestelmien mainetta yritysjohtajien silmissä, mutta tämän voidaan katsoa tuoneen tervettä kritiikkiäkin järjestelmien suunnitteluun. Onnistuneesti implementoidun ES-järjestelmän edut ovat kiistattomat, ja projektien onnistumisprosentti on valoisampi kuin vuosikymmen sitten. Käyttöönottoprojekteissa on kuitenkin vielä paljon opittavaa, ja myös toimittajilta vaaditaan kokoajan parempaa osaamista koska järjestelmät alkavat aidosti olla toimialakohtaisesti räätälöityjä. Toimittajien rooli siten kasvanut myös liikkeenjohdon konsultoinnissa.

Tätä tutkielmaa tehtäessä päällimmäisiksi asioiksi onnistuneissa ES-projekteissa nousivat yhä uudelleen hyvän esikartoituksen teko, yrityksen strategian ja prosessien huomioiminen, huolellinen järjestelmän suunnittelu ja koulutus. Järjestelmien toimittajia on monia ja monenkokoisia, mutta osaavat konsultit saavat pienilläkin resursseilla enemmän hyvää aikaa kuin monta keskitasoista. Vaikka toiminnanohjausjärjestelmän brändi tai toimittajakohtainen valinta ovatkin eräs tapa valita järjestelmä, ei niiden merkitystä voida nostaa edellä mainittujen asioiden tasolle. Toiminnanohjausjärjestelmän implementointi eroaa paljon yritysten menneiden vuosien IT-hankinnoista sen vaatiman suunnitelmallisuuden vuoksi. Enää ei voida vain ostaa työasemia perus toimisto- ja prosessikohtaisilla ohjelmilla varustettuna. Suunnittelu täytyy viedä pitemmälle, koko yrityksen strategiasta yksittäisiin transaktioihin. ES-projektit ovat mielenkiintoi-

nen katsaus koko yrityksen toimintoihin, ja vaikka muuttujia on monia, lopputulos on hallittavissa.

Viiteluettelo

- [Botta-Genoulaz and Millet 2005] Valérie Botta-Genoulaz and Pierre-Alain Millet, A classification for better use of ERP systems, *Computers in Industry* **56**, 6 (Aug. 2005) 573-587.
- [Davenport, 2000] Thomas Davenport, *Mission Critical: Realizing The Promise Of Enterprise Systems*. Harvard Business School Press, 2000.
- [Forsman, 1995] Lauri Forsman, *ATK-projektin läpivienti*, Suomen Atk-kustannus, Gummerus, Jyväskylä 1995.
- [Huigang and Yajiong, 2004] Liang Huigang and Xue Yajiong, Coping with ERP-related contextual issues in SMEs: a vendor's perspective, *Journal of Strategic Information Systems* **13**, 4, (Dec. 2004) 399-415.
- [Hyötyläinen ja Kalliokoski, 2001], R. Hyötyläinen ja P. Kalliokoski, Tietojärjestelmien käyttöönottoprosessi, Julkaistu Teoksessa Kettunen, J. ja Simons, M. (Toim.), *Toiminnanohjausjärjestelmän käyttöönotto pk-yrityksessä*, VTT Automaatio, Espoo 2001.
- [Kataja, 2004] Kari Kataja, *Tietojärjestelmäprojektiin valmistautuminen yrityksen ollessa osa konsernia*, Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, Raportti **B-2004-2**.
- [Kettunen 2002] Sami Kettunen, *Tietojärjestelmän ostaminen – käytännön opas yrityksille*. WS Bookwell Oy, Porvoo, 2002.
- [Mäkipää, 2002] Marko Mäkipää, *Toiminnanohjausjärjestelmän käyttöönotto – teoreettinen metodi ja empiirinen koettelu kahdessa case-yrityksessä*, Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, Pro gradu -tutkielma, 2002.
- [Nykänen, 2005] Pirkko Nykänen, *Tietojärjestelmien suunnittelumenetelmät*, Tampereen yliopisto, Tietojenkäsittelytieteiden laitoksen kurssi ja luentomateriaali (9. 2005).
- [Ruuhonen ja Salmela, 2003] Mikko Ruuhonen ja Hannu Salmela, *Yrityksen tietohallinto*, Edita Prima Oy, Helsinki, 2003.
- [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**, 1 (2003) 81-100.

Mainonta WWW-sivuilla - tavat ja tekniikat

Juha Heinonen

Tiivistelmä.

Internet-mainonta on kasvanut käsi kädessä Internetin kasvun kanssa. Www-sivustot täyttyvät yhä useammin mitä erilaisimmista mainoksista. Valta-asemassa ovat kuitenkin ns. bannerit, jotka ovat tietyn kokoisia kuvia, usein miten GIF-, JPG- tai Flash-formaatissa. Yhä on kuitenkin nähtävissä myös pelkkää tekstiä olevia hyperlinkki-mainoksia. Itse asiassa niiden määrä on kasvanut viime aikoina, sillä yhä useammat käyttäjät ovat kyllästyneet vilkkuviin ja välkkyviin sekä pahimmissa tapauksissa soiviin mainoksiin ja ovatkin estäneet niiden näkymisen kokonaan. Käyttäjälle onkin tarjolla useita erilaisia keinoja estää mainokset. Niin kaupallisia ohjelmistoja kuin ilmaisia vinkkejä kuinka muuttaa selaimen asetuksia.

Avainsanat ja -sanonnat: Internet-mainonta, banneri, toteutus, estäminen, kohde-ryhmät

CR-luokat: J.4, J.7, K.1, K.2, K4.0, K.6.5, K.8.1

1. Johdanto

Internetissä on ollut mainoksia sen alusta asti ja niiden määrä vain kasvaa. Www-sivuilla on hyperlinkkien ja bannerien lisäksi erilaisia appletteja, sekä muita mainoksia. Tässä tutkielmassa kartoitan sitä, kuinka tämän päivän tilanteeseen on tultu, sekä sitä minkälaisessa mainosmaailmassa nykyään elämme. Olen tarkkaillut kolmea suurta suomalaista Internet-sivustoa ja niiden mainoksia.

Internet-mainonta on tekniseltä toteutukseltaan nykyään todella monipuolista. Yhä on käytössä alkuaikojen hyperlinkit. Niiden lisäksi on käytössä jo pitkään olleet GIF- ja JPG-kuvaformaatit sekä viime vuosina kovan nousun tehnyt Flash. Flashin suosion kasvuun liittyy myös Internet-yhteyksien nopeuksien kasvu, sillä Flash, etenkin huonosti optimoituna, vie hieman enemmän kaistaa. Luonnollisesti yhä uusia tekniikoita tulee ja ne valtaavatkin alaa jo olemassa olevilta. Näistä tämän hetken, ehkä kuumin trendi on DHTML-mainokset jotka pystyvät ”kellumaan” sivun päällä.

Mainoksien estäminen on myös oma maailmansa, ja siihen onkin tarjolla useita ei kaupallisia ohjelmistoja sekä aivan ilmaisia monilla useilla www-sivuilla. Sitä mukaa kun keksitään uusia tapoja estää mainosten näkyminen, myös mainostajat keksivät uusia tekniikoita kiertää nämä esteet.

Tulevaisuus on vielä auki, mutta on varmaa etteivät mainokset katoa www-sivuilta. Myös muut mainosmuodot Internetissä, kuten sähköpostimainonta ja etenkin laitton

roskaposti, tulevat lisääntymään. Onnistuuko roskaposti tukkimaan koko Internetin, jää nähtäväksi.

2. Internet mainonnan ympäristönä

Internet kehittyi teknisesti vain vähän vuosien 1989 ja 1995 välillä, mutta sen suosio kasvoi räjähdysmäisesti ja käyttäjämäärien kasvu on jatkunut hurjana. Nykymuotoisen Internetin ja Internet-mainonnan voidaan laskea alkavan vuodesta 1995. Siitä vuodesta tähän päivään on Internetin käyttäjämäärä yli 60 kertaistunut [1]. Internet-mainontaan käytetty raha yksistään USA:ssa kasvoi vuosien 1999-2003 välillä yli kaksinkertaiseksi ja kasvun on odotettu vain kiihtyvän [Janoschka 2004]. Käyttäjämäärän kasvua on kuvattu taulukossa 2.1. ja rahan käyttöä Internet-mainontaan USA:ssa taulukossa 2.2.

Taulukko 2.1. Internetin käyttäjämäärät vuosina 1995 - 2005

Vuosi	käyttäjien määrä	lähde
1995	16 milj.	IDC ⁽¹⁾
1996	36 milj.	IDC
1997	70 milj.	IDC
1998	147 milj.	C.I. Almanac ⁽²⁾
1999	248 milj.	Nua Ltd. ⁽³⁾
2000	451 milj.	Nua Ltd.
2001	513 milj.	Nua Ltd.
2002	587 milj.	Internet World Stats ⁽⁴⁾
2003	719 milj.	Internet World Stats
2004	817 milj.	Internet World Stats
2005 (marraskuu)	972 milj.	Internet World Stats

¹⁾ <http://www.idc.com>

²⁾ <http://www.c-i-a.com/index.htm>

³⁾ http://www.nua.com/surveys/how_many_online/world.html

⁴⁾ <http://www.internetworldstats.com/pr/edi007.htm>

Taulukko 2.2. Internet-mainontaan käytetty raha USA:ssa vuosina 1999-2003.

VUOSI	miljoonaa dollaria
1999	4,600
2000	6,000
2001	7,500
2002	8,906
2003	10,688

Ensimmäiset mainokset Internetissä olivat staattisia tekstejä. Ne eivät sisältäneet edes hyperlinkkiä vaan olivat samoin luettavissa kuin esimerkiksi sanomalehtimainokset.

Pikku hiljaa tekstiin tuli mukaan hyperlinkki-ominaisuus. Yhteyksien nopeutumisen myötä alkoi sivustoille ilmestyä myös kuvia. Alun bittikarttojen jälkeen siirryttiin nopeasti GIF-formaattiin, joka on laajalti käytössä tänäkin päivänä. 2000-luvun aikana on Flash lyönyt itsensä läpi, ja yhä useampi mainos onkin toteutettu sillä. Flash on oikein tehtynä kevyttä ja hyvän näköistä, ja sillä on mahdollista toteuttaa mitä erilaisimpia interaktiivisia toimintoja.

Interaktiivisuus onkin Internetin suurin erikoisuus muihin mainontaympäristöihin verrattuna. Haroing Li [2] toteaaakin Internetin olevan poikkeuksellinen ympäristö kolmesta eri syystä. Ensimmäisenä Internet ei ole vain kommunikointikanava, vaan se toimii myös ostotapahtuma- ja jakelukanavana. Asiakkaat voivat saada tietoa haluumistaan tuotteista kaikkialta Internetistä. Mikään muu ympäristö ei pysty yhdistämään näitä toimintoja ilman, että muut toiminnon muuttuvat. Toiseksi, Internetin ominaispiirre on interaktiivisuus. Tuotteiden hakeminen, selaaminen, tutustuminen ja ostaminen vaativat hyperlinkkien seuraamista ja valintojen tekemistä. Näin ollen Internet toimii jatkuvasti kaksisuuntaisena kanavana. Kolmantena on mahdollisuus multimedia-sisältöön. Näiden myötä Internetistä onkin tullut osa niin kutsuttua Media-Mixiä.

Www-sivustot eivät ole Internetin ainoa mainoskanava, vaan mainoksia on liitetty kaikkeen mahdolliseen. Kaikille on varmasti tuttua sähköpostimainonta ja etenkin sen laitton muoto eli roskaposti. Lain mukaan mainospostia ei saa lähettää ilman vastaanottajan suostumusta. Yhtälailla mainoksia löytyy myös uutisryhmistä, chateista eli keskusteluryhmistä, näytönsäästäjistä, verkkopeleistä ja yhä useammin niihin törmää myös e-sponsoroinnin muodossa. Tässä tutkielmassa kuitenkin keskitytään www-sivuilla tapahtuvaan mainontaan.

3. Mainosten tuottaminen

Ennen yrittäjät säästivät tekemällä itse mainoksensa. Nykyisin valtaosa Internet-mainoksista on ammattilaisten suunnittelemaa ja toteuttamaa. On opittu myös työn teettäjän ja tekijän välisen yhteistyön tärkeys. Sweeney [2004] kertoo, että 1990-luvun puolella mainoksen tuotanto tapahtui jotakuinkin seuraavasti. Tilaaja tuo www-suunnittelijalle nipun mainoksia, esitteitä, tilastoja ja muuta markkinointi materiaalia ja odottaa www-suunnittelijan koostavan näistä sivuston. Useinhan www-suunnittelijalla ei ole tuotteiden ja niitä myyvän yrityksen tuntemusta, eikä hän voi tietää mitä asioita yritys haluaa tuoda esille. Tästä johtuen tuloksena oli usein kooste paperimuodossa olevasta mainosmateriaalista.

Nykyään yritykset ovat oppineet kartoittamaan asiakkaidensa tarpeet ja sivustoja luodaan näiden tarpeiden pohjalta. Selvitetään sivustot, joilla omaa tuotetta kannattaa mainostaa, mitä tietoa sivustolle kannattaa laittaa esille ja minkälaiset sivujen tulisi käytettävyydeltään ja ulkoasultaan olla. Myös markkinoiden odotuksia tutkitaan yhä tarkemmin erilaisilla markkinointianalyysillä.

4. Mainosten esto

Vaikka Internet nykyään pursuaa mitä mielikuvituksellisimpia interaktiivista toimintaa sisältäviä mainoksia, niin edelleen on myös nähtävissä alkuajoilta tuttuja hyperlinkki tekstejä. Puhtaiden tekstimainosten on jopa sanottu lisääntyvän, sillä useita käyttäjiä ärsyttävät suuret, ruudulle hyppivät mainokset ja näitä onkin alettu estää erilaisilla ohjelmistoilla kuten Normanin AdWare, joka on yksi tunnetuimpia mainostenesto-ohjelmistoja [3]. Myös useat virustorjuntaohjelmat sisältävät oheistuotteina erilaisia mainostenestojärjestelmiä, usein niin kutsuttuja pop-up killereitä. Pop-up killerit, eli ponnahdusikkunoiden tappajat nimensä mukaisesti estävät ponnahdusikkunoiden aukeamisen ilman käyttäjän hyväksyntää.

Kaikissa nykyaikaisissa selaimissa on mahdollisuus estää kuvien näkyminen. Näin käyttäjä voi halutessaan estää kuvaformaateissa olevien mainoksien näkymisen. Selaimet itse eivät pysty erottelemaan mainoksia ja muita sisältöön kuuluvia kuvia, vaan kaikki kuvat katoavat näkyvistä. Toiminnosta on hyötyä myös hitailla yhteyksillä ja erilaisilla mobiililaitteilla Internetiä selatessa. Myös pelkkien mainoskuvien estoon on kehitetty tekniikoita, joista toimivimpia on userContent.css-tiedoston muokkaus [4]. Kyseinen tekniikka toimii ainakin Firefox, Mozilla-, Camino-, Netscape- ja Safari-selaimilla. Käytännössä ideana on määritellä userContent.css-tyylitiedostoon kohta, jossa estetään näyttämästä kuvia jotka toimivat linkkinä osoitteeseen joka sisältää "ad" eli mainos-sanana. Tietysti kieltolistaan on mahdollista lisätä myös muita sanoja. Samaa tekniikkaa käyttäen voidaan piilottaa myös omiin kehyksiin aukeavat sivut, jotka sisältävät mainoksia.

Sivustoja yhä enemmän valloittavat Flash-tekniikalla toteutetut mainokset ovat myös melko helposti käyttäjän itse estettävissä. John Haller on www-sivustollaan esitellyt kolme tapaa estää Flash-mainokset [5]: Ensimmäisenä Haller mainitsee Flashin poistamisen koneelta kokonaan. Ongelmana on tällöin myös kaiken hyödyllisen Flash-sisällön menettäminen. Toinen vaihtoehto on Ad-block eli mainostenesto-ohjelmat. Ohjelmia on useita erilaisia, ja ne toimivat useimpien selainten kanssa. Ongelmana on, että yhä useammat mainokset osaavat ohittaa esto-ohjelmat. Kolmas tapa on Gecko-pohjaisten selainten FlashBlock, eli Flash-esto. Tämä toiminto muuttaa Flash-mainokset napeiksi, joita painamalla mainokset saa esiin. Toiminto on käytettävissä vain Geckoydintä käyttävillä selaimilla. Flash-eston voi toteuttaa esimerkiksi editoimalla userContent.css-tyylitiedostoa [6]. Neljäs tapa on mahdollinen Internet Explorerin käyttäjille. Siinä editoidaan Windowsin rekisteritiedostoja. Valmiita muokkauksia on saatavilla Internetistä useita, joten suurta tietotaitoa ei käyttäjältä vaadita. Tarjolla on myös kaupallisia ja ilmaisia ohjelmistoja, jotka estävät mainosten näkymisen. [7]

Mikään mainituista tavoista ei siis ole kaikille soveltuva ja varmasti toimiva. Mainostajat tuntuvat jatkuvasti olevan askeleen edellä esto-ohjelmien ja -käytäntöjen suunnittelijoita.

5. Mainosten formaatit

Valtaosa mainoksista on toteutettu tekstinä (hyperlinkki), kuvana (JPG- tai GIF-formaatti) tai Flashilla. Näiden lisäksi on hyvin vähän mm. JavaScriptillä tuotettuja mainoksia. Voidaan kuitenkin sanoa, että kolme ensiksi mainittua tapaa ovat ylivertaisessa asemassa.

Alkuaikojen hyperlinkkimainokset ovat jälleen nostaneet päätään ja esimerkiksi Google ei salli sivustollaan muita kuin tekstipohjaisia mainoksia. Google myös välittää mainoksia muille sivustoille. Voitkin siis törmätä "Ads by Gooooogle" hyperlinkkimainoksiin myös muualla kuin Googlen omalla sivustolla.

Hyperlinkkimainoksissa on siis kyse aivan tavallisesta tekstistä, joka toimii linkkinä halutulle sivustolle. Mainostettava asia on saatava ilmaistua tehokkaasti ja tiiviisti sanoin. Lähes alusta asti on Internetissä käytetty CompuServen kehittämiä GIF-formaatissa olevia kuvia. GIF-formaatin erikoisominaisuuksina ovat läpikuultavuus halutulle värille ja mahdollisuus tehdä animoituja kuvia. Animaatioissa on useita kuvatiedostoja peräkkäin tallennettuna yhteen tiedostoon, jolloin katsojalle jää vaikutelma liikkuvasta kuvasta. Samalla kun Internet-yhteydet nopeutuivat, lisääntyi myös kuvien määrä Internetissä. Kokonaisia sivustoja alettiin rakentaa lähes pelkästään kuvista. Tästäkin trendistä on onneksi päästy jo ohi ja rakenteessa on palattu enemmän tekstipohjaiseen. Digitaalikuvauksen yleistyttyä on myös valokuvien määrä verkossa lisääntynyt hurjasti.

Valokuvat ja muut tarkkuutta vaativat kuvat tallennetaan yleensä JPG-formaattiin. Tämä formaatti sopii erityisen hyvin paljon värejä vaativaan tarkkaan tallennukseen. JPG- ja GIF-formaattien lisäksi Internetissä on myös jonkun verran PNG formaatissa olevia kuvia. PNG:n vahvuuksia ovat Alpha-kanavan läpinäkyvyys, gamma-korjaus ja pieni hävikki tiedoston pakkauksessa [8]. Useimmat www-selaimet tukevat PNG-tiedostoja.

Vuonna 1995 Jonathan Gay [2001] päätti kehittää yhdessä ystäviensä kanssa heidän SmartSketch-piirto-ohjelmansa Internetiin soveltuvaksi animointiohjelmaksi. Ongelmaksi muodostui se, etteivät silloiset www-selaimet tukeneet plug-in:jä eli laajennuksia. Ainoa tapa olikin toteuttaa ohjelma Javan avulla, minkä vuoksi se osoittautui todella hitaaksi. Myöhemmin samana vuonna Netscape julkaisi heidän plug-in API:n, jolloin laajennusten tekeminen tuli mahdolliseksi. Tuotteen nimi muuttui matkalla FutureSplash Animator:ksi ja sen toimitus tapahtui kesällä 1996.

FutureSplash sai hyviä asiakkaita, sillä Microsoft ja Disney päättivät käyttää sitä. Disney teki yhteistyötä Macromedian kanssa, joka hieman myöhemmin päätti ostaa FutureSplashia tehneen FutureWave-yhtiön ja ohjelma sai uuden nimen, Macromedia Flash 1.0.

Flash on vektoripohjalla toimiva animointityökalu. Sen etuihin kuuluu täydellinen skaalautuvuus käyttäjän resoluution mukaan, helppous tehdä liikkuvaa kuvaa ja toiminta kaikissa selaimissa, joissa on Flash plug-in. Samainen plug-in on myös toisaalta Flashin pullonkaula, sillä ilman sitä, Flash ei toimi. Flashin avulla mainoksiin pystytään toteuttamaan myös paljon interaktiivisuutta, kuten alusvetovalikkoja tai pelejä.

DHTML:n avulla voidaan sivustoille saada huomioarvoltaan suuri mainos. DHTML-mainos tulee sivun päälle kellumaan ennalta määräytyksi ajaksi, jonka jälkeen se katoaa itsestään. Usein mainoksesta löytyy myös erillinen "sulje"-nappula. Ylläpitäjä pystyy määrittämään DHTML-mainokselle myös frekvenssin, joka kertoo montako kertaa kyseinen mainos yhdelle käyttäjälle näytetään. DHTML-mainokset ovat viime aikoina lisääntyneet huomattavasti ja ne ovatkin erittäin tehokas tapa kiinnittää käyttäjän huomio. Suurin DHTML:n ongelma on ettei se ole täysin yhteensopiva kaikkien selainten kanssa. Esimerkki paikallaan kelluvasta DHTML-mainoksesta on liitteessä 2 ja liikkuvasta DHTML-mainoksesta liitteessä 3.

6. Mainosbannerit

Bannerimainosten perusidea on pysynyt samana siitä asti, kun niitä alettiin käyttää. Banneri on kuva, joka pyrkii kiinnittämään katsojan huomion. Lähes kaikki bannerit toimivat myös linkkinä tuotetta tai palvelua myyvän tahon www-sivuille. Ajan kuluessa on bannerien kokoihin muodostunut standardit, joihin on kuitenkin tullut poikkeamia jatkuvasti. Perinteinen vaakatasossa oleva standardibanneri on 468 x 60 pikseliä. Tämän lisäksi IAB, eli Interactive Advertising Bureau, on listannut taulukossa 6.1 olevat koot standardeiksi [9].

Taulukko 6.1. Bannerien standardikoot IAB:n mukaan.

468 x 60 – banneri
234 x 60 – puolikas banneri
88 x 31 – mikrobanneri
120 x 90 – painike 1
120 x 60 – painike 2
120 x 240 – pystybanneri
125 x 125 – neliö painike
728 x 90 – jättibanneri

Esimerkiksi Suomen suurimpiin kuuluva verkkomarkkinoinnin välittämiseen keskittynyt yritys White Rabbit välittää kolmea eri kokoa: banneria, jättibanneria sekä 140x350 pikselin kokoista suurtaulua. Näin ollen heidän valikoimansa on rajattu vain muutamaaan vaihtoehtoon. White Rabbit on aito kohderyhmämedia ja se tavoittaa lähes kaikki 15-24-vuotiaat Internetin käyttäjät [10]. Yleisimmät bannerien sijoituspaikat näkyvät liitteestä 4.

Bannerimainokset on toteutettu lähes aina JPG-, GIF-, tai Flash-formaatissa. JPG:tä käytetään tarkkoihin still-kuviin kun taas GIF-formaattia animoituun tuotokseen.

Flashilla sen sijaan on toteutettu yleensä enemmän animaatiota, interaktiivisuutta tai muuten erilaisempia elementtejä sisältävät bannerit.

Sweeney [2004] on listannut hyvän bannerimainoksen ominaisuuksia seuraavasti:

- bannerin tulee olla nopeasti latautuva, kooltaan mielellään alle 5k.
- yksinkertaista, ei liikaa tekstiä, värejä, animaatiota
- helposti luettava fontti ja koko
- käytä ALT-tagia, jotta myös ne jotka surffaavat ilman kuvia tietävät mistä on kyse
- linkitys oikein ja oikealle sivulle
- animaatiossa 2-4 freimiä
- mainoksen pitäisi sisältää "click here / klikkaa tästä" tai vastaava vihjeteksti
- testaa toimivuus eri selaimilla
- jos et tiedä mitään bannerimainonnasta, älä yritä tehdä sitä itse. Jos teet itse niin pyydä mielipide joltakulta toiselta.

Nykyisin bannerimainoksen voi toteuttaa myös muuten kuin neliskulmaisena laatikkona sivun reunassa. Yksi malli on laajentuvat bannerit. Banneri siis laajenee kun käyttäjä klikkaa sitä, tai vie hiiren sen päälle. Suurempaan banneriin voi laittaa esimerkiksi lisätietoja mainostetusta tuotteesta tai palvelusta. Käyttäjä ohjataan eteenpäin vasta, kun hän klikkaa laajennettua banneria tai haluttua kohtaa siinä. Eri muotoiset bannerit ovat yleensä todellisuudessa neliskulmaisia, joista on määrätty läpinäkyvyys tietyille värille. Näin käyttäjälle jää mielikuva esimerkiksi soikeasta kuvasta.

Banneriin voidaan sisällyttää myös videokuvaa ja ääntä. Esimerkiksi Soneralla on ollut marras-joulukuussa 2005 mainos, jossa käyttäjä on ensin klikannut itsensä katsomaan videota. Pienen klipin jälkeen käyttäjälle tarjotaan mahdollisuutta osallistua kilpailuun Soneran sivuilla. Näin on saatu interaktiivinen mainoskokemus, joka sisältää kuvaa ja ääntä. Pelkän äänen käytön mallina on esimerkiksi Saunalahden puheaikalajakortin mainosbanneri, joka soi kun hiiren osoitin viedään sen päälle. Musiikkina on teeman mukaisesti joululauluja.

Banneriin voidaan myös sisällyttää alasetoalikko tai muu vastaava, jolla käyttäjä voi itse kohdistaa mainontaa tarkemmin. Kyseessä voi olla vaikkapa matkatoimiston mainos, jossa on alasetoalikkossa eri matkakohteita. Käyttäjä valitsee häntä kiinnostavan kohteen, jonka jälkeen hänet ohjataan kyseistä kohdetta esittelevälle sivulle.

7. Havainnot suomalaisilla sivustoilla

Olen tutkinut kolmea suurta suomalaista Internet-sivustoa. Näillä kaikilla on aktiivinen käyttäjäkunta ja säännöllisesti mainostajia. Seuraavassa on esitetty sivustojen esittelyt ja yhteenveto sivustojen sisältämisestä mainoksista. Tarkempi selvitys liitteessä 1.

Klubitus.org – Elektronisen musiikin harrastajien tiedotus ja keskustelusivusto

- rekisteröityneitä käyttäjiä lähes 8500
- viikon nettokävijämäärä yli 24 000
- keski-ikä yli 22v.
- 3-5 yhtäaikaista mainosta, joista 1-2 joka latauskerralla vaihtuvia.

- mainostajat elektronisen musiikin tapahtumat, elokuvat, puhelinoperaattorit, nuorisomatkatuimistot ja matkapuhelinvalmistajat
- Lähes kaikki mainokset sisälsivät animaatiota, muttei mikään interaktiivista toimintaa
- selvästi 15-25 vuotiaille suunnattuja tuotteita, tapahtumia ja palveluita
- käytetyt tekniikat: GIF, JPG ja Flash.

MTV3.fi – Suomen suurin Internet-media ja käytetyin Internet-sivusto [11]

- viikon nettokävijämäärä 1 352 000
- yhteensä lähes 6 miljoonaa käyntiä viikossa
- 3-5 yhtäaikaista mainosta + MTV3:n omat mainokset. Näitä ovat mm. MTV3 Store, soittoäänet ja Se Oikea – deittipalvelu.
- mainostajat: Pankit, Suomen Posti, automerkit, kodintavaratalot, kartta- ja numeropalvelut, vaateliikkeet, kodintekniikka
- Yhtä poikkeusta lukuun ottamatta kaikki mainokset sisälsivät animaatiota, pari mainosta sisälsi myös interaktiivisuutta. Tosin käyttäjä ei päässyt kohdentamaan mainosta tarkemmin vaan interaktiivisuus oli lähinnä ”kiva lisä” .
- Mainokset selvästi suunnattu yli 25 vuotiaille työssäkäyville. Tätä kertoo mm. kodintavaratalojen ja automaisten näkyminen.
- käytetyt tekniikat: GIF, JPG ja Flash

irc-galleria.net – Suomen suurin Internet-yhteisö

- yli 250 000 rekisteröitynyttä käyttäjää
- viikon nettokävijämäärä 510 000
- yli miljardi sivulatausta kuukaudessa
- suurin käyttäjäryhmä 15-17 vuotiaat nuoret, miehiä hieman enemmän kuin naisia
- mainospaikkoja 5, joista tutkimuksen aikana 3-4 käytössä
- mainostajat: puhelinoperaattorit, soittoäänet, Pelastakaan Lapset ry., virvoitusjuomat ja matkapuhelinvalmistajat
- Yhtä mainosta lukuun ottamatta kaikki sisälsivät animaatiota, yhdessä mainoksessa interaktiivista toimintaa jossa käyttäjä pääsi esikuuntelemaan soittoääniä. Ei laajennettuja, eikä DHTML-mainoksia vaikka ylläpitäjä niihin mahdollisuuden tarjoaakin.
- Mainokset selvästi suunnattua alle 20-vuotiaille. Tavarat ja palvelut pikkurahalla ostettavissa tai vanhempien lapsilleen ostamia
- käytetyt tekniikat: GIF, JPG ja Flash.

Kun tarkastellaan, millaisia tuotteita kullakin sivustolla mainostetaan, niin huomataan, että kyseessä on selvä kohderyhmämarkkinointi. Aikuisille mainostetaan autoja, kun taas nuorille soittoääniä. Elektronisen musiikin kuuntelijoille alan tapahtumia ja irc-galleriassa Pelastakaan Lapset ry. muistuttaa, ettei Internet ole täysin riskitön ympäristö. Nämä kolme sivustoa ovat hyvä läpileikkaus suomalaisiin Internet-portaaleihin ja niillä tapahtuvaan mainontaan. On kuitenkin todettava, että sivustot olivat kaikki hyvin

samankaltaisia mainospaikkojen ja toteutusten suhteen. Esimerkiksi yhtään DHTML-mainosta en tutkimuksen aikaan niiltä löytänyt, vaikka sellaisia useilta isoilta sivustoilta jo onkin.

8. Internet-mainonnan tulevaisuus

Internet-yhteyksien nopeutumisen myötä, yhä useampia elementtejä sisältävät mainokset ovat mahdollisia. On mahdollista törmätä pop-up ikkunaan, jossa aukeaa lähes tv-mainoksen kaltainen mainos. Internet-mainonta tulee sisältämään yhä enemmän ääntä ja korkeatasoisempaa liikkuvaa kuvaa. Interaktiivisuus tulee lisääntymään erilaisten huomion herättäjien muodossa.

Samalla kun mainostulva Internetissä kasvaa, myös niitä estävät ohjelmistot ja www-selaimet yleistyvät. Monet verkon käyttävät ovat kyllästyneet pop-upeihin ja soiviin Flasheihin. Hyvän tavan mukainen Flash ei päästä ääntä ennen kuin käyttäjä on antanut siihen luvan. Monilla sivustoilla on huomattu, ettei tilaa mainoksille ole enempää ilman, että varsinainen sisältö kärsii. Tämä onkin johtanut siihen, että on alettu puhumaan banneri-sokeudesta. Asian toivat esille ensimmäistä kertaa Jan Panero Benway ja David M. Lane [1998]. Heidän mukaan käyttäjä ei huomaa bannereissa olevaa tietoa, vaikka se olisi juuri sitä, mitä hän on etsinyt. Kyseessä oli aika, jolloin mainonta ei ollut lähellekään yhtä yleistä kuin nykyään. Siksi aiheesta kirjoittelu onkin ollut vilkasta ja useat asiantuntijat arvelevatkin, että bannerimainonta nykyisessä muodossaan on ohimenevä ilmiö ja tulevaisuudessa on käyttäjän huomio kiinnitettävä toisin tavoin.

Ratkaisuksi sekä tilan puutteeseen että bannerisokeuteen ovat tulleet DHTML-mainokset, joiden huomioarvo on erittäin suuri. Käyttäjä näkee mainoksen aivan varmasti, koska se peittää osan tai jopa lähes koko sivun. Näin ollen hän joutuu vähintäänkin sulkemaan mainoksen tai odottamaan sen sulkeutumista, jotta pääsee käsiksi sivuston varsinaiseen asiasisältöön.

Muista mainosmuodoista kasvussa on ainakin e-sponsorointi. Tässä on kyse esimerkiksi siitä, että joku yhtiö sponsoroi esimerkiksi verkkopeliä. Tuotemerkki siis näkyy verkkopelissä tavalla tai toisella.

Lisäksi mainossähköpostin määrän oletetaan edelleen lisääntyvän. Etenkin luvattoman roskapostin määrän pelätään jopa tukkivan Internetin. Roskapostin lähettäminen on todella helppoa ja pienikin onnistumisprosentti tuo tuloja. Roskapostin sanotaan jo tänä päivänä kattavan useita kymmeniä, ehkä yli 50 % Internetin liikenteestä.

9. Yhteenveto

Internet ja www-sivut ovat täynnä mainoksia ja näin näyttäisi olevan myös tulevaisuudessa. Käyttäjät ovat kuitenkin osaltaan tulleet sokeiksi nykyiselle bannerimerelle, joten uusia vaihtoehtoisia tapoja ja keinoja on keksittävä. Vaikka uusia tekniikoita onkin tullut, niin silti käytetään edelleen jo alkuajoilta tuttuja hyperlinkkimainoksia.

Oman osansa tähän mainosviidakkoon tuovat tahot, jotka yrittävät estää mainosten näkymisen. Mainosten näkymistä yritetään estää sekä ilmaisilla että kaupallisilla ohjel-

milla. Internetissä on myös paljon ilmaisia vinkkejä selainten asetusten ja tyylietodotusten muokkaamiseen.

Tulevaisuutta tulee luultavasti hallitsemaan Flash, sillä niin ylivoimaisessa asemassa se nykyään on. DHTML on tuonut uuden tavan sijoitella mainoksia bannerisokeiden ihmisten kiusaksi. Sen avulla pystytään pakottamaan käyttäjän katsomaan mainosta ainakin pienen hetken. Kauanko menee siihen, että mainokset pystytään kokonaan estämään, jää nähtäväksi. Aina kuitenkin suurin osa ihmisistä ei välitä näkyvistä mainoksista.

Tutkimani Klubitus.org, MTV3.fi ja irc-galleria.net saavat kaikki tuloja säännöllisestä bannerimainonnasta. Kaikilla kolmella sivustolla on lähes samanlaiset bannerien sijoituspaikat ja niiden esittämät bannerit olivat tehty samoilla tekniikoilla. Kyseisillä sivustoilla esitetyt mainokset olivat selvästi kohdistettu sivuston kävijöille. Uutta DHTML-tekniikkaa ei kyseisillä sivustoilla tutkimuksen aikana ollut käytössä.

Viiteluettelo

[Benway ja Lane, 1998] Jan Panero Benway, David M. Lane, *Banner Blindness: Web Researcher Often Miss "Obvious" Links*, Rice University, 1998.

[Gay, 2001] Jonathan Gay, *The History of Flash*, Adobe Systems Incorporated, 2001.

[Janoschka 2004] Anja Janoschka, *Web Advertising. New forms of communication on the internet*, John Benjamins Publishing Company, 2004.

[Sweeney, 2004] Susan Sweeney, *101 Ways to Promote Your Web Site : Filled with Proven Internet Marketing Tips, Tools, Techniques, and Resources to Increase Your Web Site Traffic (5th Edition)*, Maximum Press., 2004

[1] <http://www.internetworldstats.com/emarketing.htm> - viitattu 14.12.2005

[2] <http://www.admedia.org> - viitattu 18.12.2005

[3] http://www.conseils.fi/ADWPro_fi.html - viitattu 18.12.2005

[4] <http://www.floppymoose.com/> - viitattu 19.12.2005

[5] http://johnhaller.com/jh/useful_stuff/disable_flash.asp - viitattu 19.12.2005

[6] <http://www.squarefree.com/userstyles/xbl.html> - viitattu 19.12.2005

[7] http://malektips.com/spywareblaster_0009.html - viitattu 19.12.2005

[8] <http://www.mit.jyu.fi/luk/toteutettuja/PNG/> - viitattu 17.1.2006

[9] <http://www.iab.net/standards/adunits.asp#> - viitattu 19.12.2005

[10] <http://www.whiterabbit.fi/index.php?page=yritys> - viitattu 19.12.2005

[11] <http://spotti.mtv3.fi/mediatiedot.shtml/1258?a03b02> - viitattu 17.1.2006

Liite 1. tarkasteltujen sivustojen mainokset aikavälillä 14.11-19.12.2005

sivusto / portaali	mainospaikkojen määrä ja sijainti	esiintyneet mainokset	käytetyt formaatit	Huom! (DHTML, Java, Shockwave, video, ääni)
Klubitus.org	6, joista käytössä 3-5. Etusivulla yläreuna, 2 kpl oikea reuna, 1kpl etusivun keskiosa, 2 kpl alareuna.	Elektronisen musiikin tapahtumia, matkapuhelinoperaattorit, matkapuhelinvalmistajat, elokuvat, musiikki levyt, nuorisomatkatöimistöt, dj:t	GIF, JPG, Flash	Oikean reunan ja alareunan bannereissa karuselli.
MTV3.fi	5, joista käytössä 3-5. 1kpl yläreuna, 1-3 kpl oikeareuna, 1 kpl sivun keskiosa + MTV3:n omat mainokset kuten soittoäänet ja MTV3-store	kodintavaratalot, vaatekaupat, autokauppa, pankit, Suomen Posti, laajakaistaoperaattorit, matkapuhelinoperaattorit	GIF, JPG, Flash	2 mainosta sisälsi interaktiivisen osuuden, joissa käyttäjältä ei kuitenkaan vaadittu kuin hiiren luukuttaminen kohteen päälle.
Irc- galleria.net	5, joista 3-4 käytössä. 1 kpl yläreuna, 2 kpl oikea reuna, 1 kpl etusivun keskiosa, 1 kpl alareuna.	soittoäänet, matkapuhelinoperaattorit, Pelastakaa Lapset ry., virvoitusjuoman joulupeli, Internet-selain	GIF, JPG, Flash	Soittoäänimainoksessa pystyi esikuuntelemaan äänen pitämällä hiirtä kohteen päällä hetken.

Liite 2. DHTL-mainos, paikallaan pysyvä.

The image shows a screenshot of the Suomi24 website in a Mozilla Firefox browser window. The browser's address bar displays the URL http://www.suomi24.fi/mediasettely/esimerkit/dhtol_staattinen/. The website's header includes the logo "SUOMI24.FI" and navigation links for "Keltaiset Sivut", "Henkilöt", "Kartat", "Kuvat", and "Hakemisto". A search bar with the text "Hae" and "Laaja haku" is visible, along with the "eniro" logo. The main content area features a large red circular overlay with the text "Klikkaa minua!" and a button labeled "X Sulje". Behind the overlay, a news article is partially visible with the headline "Pääministerin ero vaivasta yllätti". The article text includes: "Pääministerin ero vaivasta yllätti", "Vanhasen ero tuli täytänään", "senä. Vakuuttaako ero mielestäsi", "sen uskottavuuteen mahdollisena", "nttiehdokkaana? [Keskustele](#)", "Sinnemäki ehdolla", "puheenjohtajaksi", "Anni Sinnemäki on ilmoittanut lähtevänsä mukaan Vihreiden puheenjohtajaksi.", and a small portrait photo of Anni Sinnemäki. To the right of the article, there is a sidebar with various service links under the heading "Sähköposti", including "Gmail - Etias", "Hotmail", "iCloud", "Outlook", "Yahoo Mail", "Sähköposti", "Uutiset, sää", "Helsingin Sanomat", "Ilta-Sanomat", "Kaupalehti", "MTV3 Uutiset", "Helsingin Uutiset", "MT3 Uutiset", "Tieskö", "Yle Uutiset", and "Valitse lehti". Below the sidebar, there is a section for "Hakupalvelut" including "AltaVista", "Google", "Bing", "Koukka.com", and "Makemate". The browser's status bar at the bottom shows "Valmis" and various system icons.

Liite 3. DHTML-mainos, liikkuva



Liite 4. Yleisimmät bannerien paikat

The screenshot shows the IRC-Galleria website interface. The page is dark-themed with a blue header and a light blue sidebar on the left. The main content area is white. Four yellow rectangular boxes with black numbers (1, 2, 3, 4) are overlaid on the page to indicate common banner positions:

- 1**: Located at the top of the main content area, above the "Tervetuloa IRC-Galleriaan" section.
- 2**: Located on the right side of the page, in a vertical column.
- 3**: Located in the middle of the main content area, below the "Tervetuloa IRC-Galleriaan" section and above the "Viimeksi katsutut julkaisut" section.
- 4**: Located at the bottom of the main content area, below the "Viimeksi katsutut julkaisut" section.

The website content includes a search bar, a navigation menu, a sidebar with a profile picture and statistics, and a main content area with a welcome message and a list of recent publications.

Aikataulusuunnittelu ja sen haasteet projektityössä

Heli Helminen

Tiivistelmä.

Tutkielma käsittelee aikataulusuunnittelua ja -hallintaa yhtenä keskeisimmistä projektinhallinnallisista osa-alueista ja ongelmista. Aikataulusuunnittelu on hyvin monimuotoinen tapahtuma. Realistisen aikataulutuksen rakentaminen on haastavaa sen rakentuessa useista osatekijöistä. Suunnittelun tueksi on olemassa lukuisia arviointimenetelmiä. Tutkielma perustuu pääosin kirjallisuuteen. Lisäksi kirjoittajan omakohtainen projektityökokemus vahvistaa useita jo todettuja tutkimustuloksia.

Avainsanat ja -sanonnat: Aikataulusuunnittelu, työmääräarviointi, aikalaskenta, riippuvuussuhteet.

CR-luokat: K.6.1

1. Johdanto

Tässä tutkielmassa tarkastellaan projektityön aikataulutukseen liittyviä asioita, kuten huomioon otettavia osatekijöitä, haasteita, ongelmia ja suunnittelumenetelmiä. Projektisuunnitelman ja erityisesti aikataulusuunnitelman merkitys on huomattava projektin onnistumisen kannalta. Projekti aikataulun tarkoituksena on auttaa allokoimaan resurssit eri tehtäville optimoimalla suorituskyky. Suunnitelma toimii myös suunnittelun pohjana ulkoisille aktiviteeteille, kuten materiaalin tuotto, ylläpidon valmistelu ja tilausten toimittaminen sisäisille ja ulkoisille asiakkaille. Perusaikataulut toimivat kaiken kommunikoinnin ja koordinoinnin lähtökohtana projektin sisäisten ja ulkoisten liittymien kanssa. Niiden mukaan tehdään sopimukset ja sovitaan määräajat projektin eri vaiheille.

Tutkielman tarkoituksena on antaa joitakin malleja projektityön aikataulujen suunnitteluun ja hallintaan. Tutkielmassa käsitellään joitakin aikataulujen arvioinnin avuksi olemassa olevia laskentamalleja eri arviointimenetelmissä.

2. Projekti

Projektilla tarkoitetaan yleensä sellaista asiaa, jota varten on tilapäisesti koottu ryhmä henkilöitä suorittamaan jotakin tehtävää yhteisen tavoitteen saavuttamiseksi määrättyssä ajassa. Projektilla tulee olla selkeä tavoite, jonka täyttyminen päättää projektin. Projekti perustetaan yleensä kohtuullisen suuria hankkeita varten, pienemmät kokonaisuudet hoidetaan normaalin työn merkeissä.

Projekti jaetaan eri vaiheisiin, puhutaan projektin elinkaaresta. Elinkaaren eri vaiheet vaikuttavat toisiinsa. Muutoksia voi ilmaantua missä tahansa projektin vaiheessa ja asiat yleensä tarkentuvat projektin edetessä.

Projekti on oma kokonaisuutensa, jossa voi olla useita eri osapuolia, mutta toteutus ja vastuu on yleensä keskitetty. Projektia varten perustetaan usein tilapäinen projektiorganisaatio, joka delegoi ja suorittaa projektin eri tehtäviä. Projektiorganisaation koko voi vaihdella projektin eri vaiheissa. Monissa projekteissa työskennellään ryhmissä ja vastuut sekä valtuudet tulee olla hyvin määritelty.

Kokonaisuus muodostuu monista osa-alueista ja muuttujista, kaikki vaikuttaa kaikkeen. Kokonaisuuden pitää olla toimiva. Jokaiseen projektiin liittyy useita erilaisia riskitekijöitä ja epävarmuutta. Hyvä suunnittelu, projektin seuranta ja varsinkin riskinhallinta ovat ensisijaisen tärkeitä projektin onnistumisen kannalta. Hallinta perustuu hyvin tehtyyn projektisuunnitelmaan. Projekti-suunnitelma määrittelee mitä tehdään, missä aikataulussa, millä resursseilla ja millä pelinsäännöillä. Onnistuneen projektin lopputuloksena on aikaansaatu jokin tuote tai asia, joka on alussa asetettu tavoitteeksi ja täyttää tilaajan sille asettamat vaatimukset. [Ruuska, 2001; Pelin, 2002]

2.1. Projektien luokittelu

Projekteja voidaan luokitella useilla eri tavoilla riippuen niiden luonteesta. Tässä tutkielmassa käsitellään käytettävissä olevan ajan mukaista luokittelua. Projektityypin valinta tulee tehdä tietoisesti ja kaikkien osallisten tulee olla siitä selvillä. Ruuska [2001] jakaa projektin vaiheet aikataulullisesti normaaleihin, pikaprojekteihin ja katastrofiprojekteihin.

Normaaleissa projekteissa toteutukselle on varattu riittävästi aikaa. Suunnittelu on tehty hyvin ja niissä on huomioitu käytettävissä olevat resurssit, sekä projektin selkeä tavoite ja sen laatu. Normaali projekti luo paremmat edellytykset onnistumiselle, sillä kokonaisuus on hallittavampi. Tämä on ideaalinen tilanne ja varmasti useimmiten myös lähtötilanne. Kaikessa projektisuunnittelussa pyritään normaaliprojektin toteutukseen. Toisinaan ajaudutaan aikataulullisiin kriiseihin ja toimintatapa voi muuttua pika- tai katastrofiprojektin omaiseksi. Näitä projektityömalleja ei kannata ottaa lähtökohdaksi, mikäli valinnan varaa on. [Ruuska, 2001]

Pikaprojektissa toteutus pyritään tekemään nopeammin. Tähän päästään lisäämällä projektiin sijoitettavaa pääomaa, jolla saadaan panostettua mm. työntekijöiden lukumäärään. Tarvittaessa voidaan jopa tinkiä laatuavoitteista. Pikaprojektin toteutus sisältää jo huomattavasti enemmän riskejä. Tässä voi iskeä ns. vauhtisokeus, jolloin kontrolloitu toiminta unohdetaan ja lopputuloksena voi olla melkoinen kaaos. Lopputuote ei välttämättä ole sitä mitä odotet-

tiin. Tähän projektimalliin saatetaan ajautua, kun alkuperäisesti normaali-projektiksi suunniteltu ei etenekään hallitusti ja joudutaan tekemään hätä-ratkaisuja. Toisinaan myös ympäristön äkilliset muutokset, kuten esimerkiksi organisaatio- tai lakimuutokset voivat aiheuttaa pikaprojektin tarpeen, jossa asiat syystä tai toisesta on saatava nopeasti tehtyä. [Ruuska, 2001]

Katastrofiprojektissa melkein kaikki on sallittua. Aikatauluista ollaan myöhässä, kaiken piti jo olla valmiina. Aikaa pyritään säästämään tekemällä ylitöitä ja hyväksymällä laadullisia puutteita, ylitöistä johtuen pääomakustannukset nousevat jyrkästi. Aikataulua yritetään pelastaa kaikilla keinoilla. Katastrofi-projektin riskit ovat suuret, onnistumisesta ei ole mitään takeita. Voidaanko tällaisessa projektissa onnistumisesta enää oikein puhuakaan. Tähän ratkaisuun ei toivottavasti päädytä kuin hätätapauksissa, ikään kuin pakon sanelemana, kun mitään muuta ei enää ole tehtävissä. [Ruuska, 2001]

2.2. Projektin rajaus

Projektia valmistellessa tehdään projektin rajaus eli päätetään mitä tehdään. Sovitaan mitä lopputuotteita projektin tuloksena syntyy ja mitä niiltä vaaditaan. Rajauksen tekeminen ei aina ole helppoa ja usein se tehdäänkin liian karkeasti. Rajaus on syytä tehdä huolellisesti, sillä virheellisellä rajauksella on vaikutusta koko projektiin. Aikataulut ja työmääräarviot eivät voi pitää paikkaansa ja laatu kärsii. Ruuskan [2001] mukaan väärin tai puutteellisesti tehty rajaus on yksi yleisimmistä syistä projektin epäonnistumiselle.

Asioita on syytä pohtia laajalti ja eri näkökulmista. Päätetään, mitä toteutetaan ja mitä ei. Kumpikin osa-alue dokumentoidaan huolella. Projektin aikana alkuperäistä rajausta ei saa unohtaa, vaikka joskus siihen joudutaankin tekemään tarkennuksia muuttuvien tilanteiden vuoksi.

Projektin aikana toteutettava kokonaisuus on syytä pitää kohtuuden rajoissa. Kohtuudella tarkoitetaan lähinnä toteutettavan kokonaisuuden kokoa suhteessa käytettävissä oleviin resursseihin. Täydellisyyteen ei kannata pyrkiä mutta laatuun satsataan. Ruuska [2001] pitää peukalosääntönä, että projekti ei saisi kestää yli yhtä kalenterivuotta, mutta pidempiäkin projekteja tehdään. [Ruuska, 2001]

2.3. Projektin vaiheet

Projektilla on aina selkeä alkamis- ja päättymisajankohta. Projektin elinkaari sisältää useita eri vaiheita. Kukin vaihe on erilainen toimintamalleineen ja ongelmineen. Projektin vaiheistuksesta on olemassa useita erilaisia malleja, mutta niissä kaikissa on samat peruselementit, perustamisvaihe, suunnittelu, toteutus ja päättäminen. Vaiheet useimmiten limittyvät toistensa kanssa. Eri työvaiheisiin liittyy useita erilaisia tehtäviä. Toisinaan myöhemmissä työvai-

heissa joudutaan palaamaan aikaisempaan vaiheeseen, kun esimerkiksi suunnittelu on tehty puutteellisesti ja sitä joudutaan tarkentamaan. Organisaatiossa voi olla meneillään myös useita eri projekteja samanaikaisesti. [Ruuska, 2001]

Perustamisvaiheessa tehdään niin sanottu esitutkimus esille nousseen idean tai tarpeen perusteella. Esitutkimuksessa selvitetään, onko projektille olemassa todellisia edellytyksiä. Lisäksi kartoitetaan projektin tavoitteet, reunaehdot, sekä keskeiset ongelma-alueet. Esitutkimuksen perusteella projektin asettaja saa riittävästi tietoa päättääkseen projektin toteutuksesta. Toteutus päätöksen jälkeen laaditaan projektisuunnitelma. Projektisuunnitelmassa määritellään aikataulut, varataan resurssit ja sovitaan työtavoista. [Ruuska, 2001]

Suunnitteluvaiheessa määritellään, mitä tulevalta järjestelmältä odotetaan ja mitä sen tulee tehdä. Määrittelyn perusteella tehdään tekninen suunnittelu, jossa määritellään miten järjestelmä toteutetaan. Suunnitteluvaihe on yleensä melko pitkä projektin koko elinkaareen nähden. Tämä vaihe kannattaa tehdä erityisen huolella. Suunnitteluun käytetty aika maksaa itsensä varmasti takaisin myöhemmissä vaiheissa. [Ruuska, 2001]

Toteutusvaiheessa toteutetaan suunnitelmien mukainen järjestelmä. Lisäksi laaditaan tarvittavat dokumentit toteutuksesta ylläpitoa ja käyttäjiä varten.

Varsinaisen järjestelmätoteutuksen jälkeen suoritetaan kattava testaus. Testaus jaetaan usein kokonaisuuden laajuudesta riippuen järjestelmä- ja systeemitestaukseen. Järjestelmätestauksessa testataan järjestelmän yksittäisiä osia sekä yksittäistä järjestelmää omana kokonaisuutenaan. Systeemitestauksessa testataan kyseisen järjestelmän yhteentoimivuutta muiden järjestelmien ja niiden liittymien kanssa. Testauksella varmistetaan järjestelmän laadusta, korjataan mahdolliset virheet sekä puutteet.

Toteutuksen ja testauksen jälkeen järjestelmä otetaan käyttöön. Tässä vaiheessa järjestetään myös mahdolliset koulutukset asiantuntijoille ja muille käyttäjille. [Ruuska, 2001]

Päätämisen vaiheessa hyväksytään toteutettu järjestelmä. Sovitaan mahdollisten jäljelle jääneiden virheiden korjauksesta ja järjestelmän ylläpidosta. Lopuksi projektiorganisaatio puretaan ja projekti päätetään. [Ruuska, 2001]

3. Projektisuunnittelun yleisiä ongelmia ja heikkouksia

Projektin onnistuminen on kiinni useasta eri osatekijästä. Projektisuunnitteluun pätee vähintäänkin sanonta ”hyvin suunniteltu on puoliksi tehty”. Koko projektin toteutus perustuu tehdyille suunnitelmille. Puutteellisella suunnittelulla voi näin ollen olla hyvinkin katastrofaalisia vaikutuksia koko projektin onnistumiselle. Suurimmassa osassa projekteja ongelmat ovat hyvin saman-

tyyppisiä. Yleensä ne johtuvat puutteellisesta suunnittelusta tai huonosta projektinhallinnasta.

Yleisimpiä suunnitteluvirheitä tehdään arvioitaessa työmääriä ja aikatauluja. Näissä ollaan usein liian optimistisia. Tehokas työaika ja kalenteriaika ovat eri asia. Henkilöresurssit saatetaan yliarvioida. Samat henkilöt hoitavat useita projektin eri tehtäviä ja voivat olla muissa projekteissa samaan aikaan. Heillä on usein myös muita työtehtäviä, työkokemus ja ammattitaito vaihtelevat. Resursseja ei ole merkitty aikatauluihin, jolloin ei tunneta varsinaista resurssitarvetta. Pahimmassa tapauksessa tarvittava resurssi ei olekaan käytettävissä sitä tarvittaessa. Näillä asioilla on suora vaikutus tehtävien suoritusnopeuteen ja laatuun. Asioiden riippuvuussuhteet on syytä pitää mielessä suunnitelmia tehtäessä. [Ruuska, 2001]

Projektisuunnittelun keskeisimpiä asioita on aikataulusuunnittelu ja sen perustana oleva työmääräarviointi. Aikataulu määrittää projektin kulun ja sen seurannan. Yleisimpiä aikataulujen heikkouksia ovat liian karkeat tehtäväerittelyt ja ylläpidon puute muutosten ilmaantuessa. Aikataulut joudutaan usein rakentamaan ainakin jossain määrin arvioiden varaan. Kokematon ja liian ylimalkainen aikataulusuunnittelu on selkeä riskitekijä, lopputuloksena voi olla hyvinkin epärealistinen aikataulu. Samoin on vaarassa käydä, mikäli aikataulut sanellaan ”ylhäältä”.

Osa tehtävistä voi puuttua suunnitelmista. Varsinaista pelivaraa ei kannata sisällyttää arvioihin, mutta niihin on hyvä varautua. Pelivaran käyttöä ei kannata ottaa tavaksi, se on sallittua vain poikkeustilanteissa. Kriittisiin kohtiin on hyvä varata hieman pelivaraa, ei varsinaisiin tehtäviin. Aikataulusta poikkeaminen voi tapahtua melko huomaamatta. Yksi asia johtaa toiseen ja pienistä lipsahduksista kasaantuukin kriittinen riski projektin onnistumiselle.

Tehtävien väliset riippuvuudet puuttuvat usein. Arvioita ei kannata tehdä liian suurina kokonaisuuksina. Hyvän perustan aikataulusuunnittelulle antaa projektiositus (WBS). [Pelin, 2002; Ruuska, 2001]

4. Työmääräarviointi

Aikataulun suunnittelu perustuu projektille asetettuihin reunaehtoihin, työvaiheisiin ja niiden työmääräarviointiin. Tehtävien laajuutta ja kestoja on yleensä melko vaikea arvioida etukäteen. Arvioinnin avuksi on olemassa monia erilaisia tekniikoita, mutta niitä tehdään usein myös sattumanvaraisesti. Täsmällistä laskentamallia ei ole olemassa. Vaikutusta on myös projektin aikana ilmaantuvilla muutoksilla ja yllätyksillä. Muutoksenhallinta ja yllätyksiin varautuminen (riskinhallinta) ovatkin hyvän projektisuunnittelun ja -hallinnan ominaisuuksia.

Työmääriä voidaan arvioida edeltävien kokemusten perusteella samantyyppisistä tehtävistä ja projekteista. Aikatauluja tarkennetaan ja päivitetään projektin edetessä asioiden ja siten myös tehtävien ja niiden työmäärien tarkentuessa. Kuten yleensä projektisuunnitteluun, myös työmäärien arviointiin kannattaa panostaa. Arviot ovat sitä tarkempia, mitä enemmän niihin nähdään vaivaa. [Pelin, 2002; Ruuska, 2001]

Työmäärien arviointi edellyttää projektin rajaamista. Toteutusvaiheen työmääräarvio on realistisempi, kun projektin suunnitteluvaihe on tehty. Työmääriä arvioidessa täytyy tietää projektin tavoite ja prosessiin liittyvät tehtävät. Tehtävien kartoittamisessa projektin osittaminen on erittäin hyvä menetelmä. Lisäksi työmääräarvioihin pitää sisällyttää projektin hallinta, palaverit, erilainen yhteydenpito, dokumentointi ja koulutus valmisteluineen. Yleensä asioita arvioidaan liian optimistisesti. Realistisuus on hyvin tärkeää projektisuunnitelmia ja arvioita tehtäessä. Toisaalta voitaisiin sanoa, että pieni pessimistisyyskään ei olisi pahitteeksi. [Ruuska, 2001]

4.1. Työkokonaisuuksien osittaminen

Projekti voidaan osittaa esimerkiksi rakenteen, toteutusvaiheiden, järjestelmien tai toiminnallisuuksien perusteella pienemmiksi toteutuskokonaisuuksiksi. Projektiosittelua nimitetään WBS:ksi (Work Breakdown Structure, kuva 1). Työvaiheiden ositus jakaa projektin organisatorisesti omiin osaprojekteihin ja vastuualueisiin.

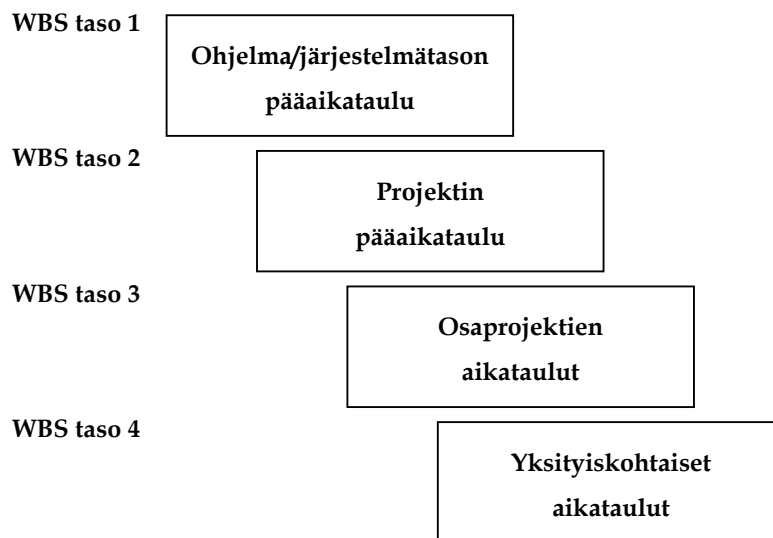
Kunkin eri vaiheen sisällä projekti jaetaan rinnakkaisiin osaprojekteihin, jotka puolestaan jaetaan osaprojekteihin kokonaisuuden laajuudesta riippuen. Lopulta osina ovat yksittäiset tehtävät. Kustakin vaiheesta syntyy jokin mitattavissa oleva tulos. Vaiheen lopussa saatetaan pitää ns. vaihekatselmus, jossa tuotokset ja tulokset tarkistetaan. Projektia on helpompi hallita pienempinä kokonaisuuksina. Mitä yksityiskohtaisemman osittelun voi tehdä, sitä tarkempiin työmääräarvioihinkin päästään. Osittelua voidaan tarkentaa projektin edetessä, kun asioista saadaan tarkempaa tietoa. Osittelutekniikkaa voidaan myös muuttaa vastaamaan paremmin kutakin kokonaisuutta. [Ruuska, 2001; Pelin, 2002]

Projektinhallintaohjelmissa on mahdollista osittaa koko projekti ja toteuttaa ns. työn hajautusrakenne. Osittelun tuloksena syntyvät kokonaisuudet (WBS-rakenteet) voi muotoilla ja numeroida yksilöllisesti kunkin projektin vaatimusten mukaisesti ja varmistaa samalla tietojen yhdenmukaisuus. Microsoftin MS-Project-ohjelmassa, joka on kehitetty projektinhallinnan apuvälineeksi, projektin työtehtävien osituksen tasot saadaan hyvin kuvattua ja raportoitua. Projekti jaetaan osakokonaisuuksiksi kaikkine alatehtävineen ja riippuvuuksineen.

Tehtävät jäsennellään ja niille asetetaan etapit ja resurssivaraukset. [Huotari et al., 1995]

Hyvin tehty osittelu on erinomainen perusta aikataulusuunnittelulle. Riippumatta siitä, mitä aikataulujen arviointitekniikkaa käytetään, osittelu on aina hyvä tehdä mahdollisimman tarkalla tasolla. Pelin [2002] kutsuu ositusta projektin selkärangaksi. Sen perusteella työ jäsennellään hierarkkisesti, ajallinen ja taloudellinen suunnittelu ja ohjaus integroidaan sen avulla. Tämä menetelytapa on melko työläs, mutta suunnittelussa voidaan päästä melko tarkkoihin arvioihin. [Pelin, 2002; Ruuska, 2001]

Projektin osituksen ylimmällä tasolla on koko osa-alueen (esim. yhden järjestelmän) pääaika-aulu. Tämä jaetaan edelleen järjestelmäprojekteihin, joiden aikataulut ovat kukin osa järjestelmätason pääaika-aulua. Jokainen järjestelmäprojekti jaetaan edelleen osaprojektien aikatauluihin, jotka ovat kukin jonkin projektin aikataulua. Osaprojekti jaetaan työtehtävätason aikatauluihin. Jako voi olla hyvinkin yksityiskohtainen. Käytännössä aikataulutus toimii harvoin näin hyvin. Usein on ongelmia kokonaisuuden esittelytarkkuuden, toimivuuden ja riippuvuuksien hallinnassa. [Pelin, 2002]



Kuva 1. Aikataulujen liittymät toisiinsa projektin osittelussa (WBS-rakenne).

[Pelin, 2002]

4.2. Sopivan henkilön valinta

Tehtäviin kannattaa aina valita sopivin henkilö. Tehtävän työmäärä ja kesto on riippuvainen suorittavan henkilön sopivuudesta. Aikataulusuunnitelma on hyvä tarkistaa aina sen jälkeen, kun kunkin tehtävän suorittaja on tiedossa. Työn

suoritukseen vaikuttavat tehtävän vaikeusasteen ja laajuuden lisäksi suorittavan henkilön kokemus, osaamisalueet, työtapa sekä mahdolliset muut tehtävät.

Sopivuudeksi voidaan varmaan laskea myös projektiorganisaation yhteen hiileen puhaltaminen ja henkilökemioiden toimivuus. Turha kitka henkilöstön välillä hankaloittaa tehtävien suorittamista. Huono ilmapiiri ei myöskään ole kovin hyvä motivaatiotekijä. Motivoitunut ja positiivinen henkilö on varmasti tehokkaampi ja työn jälki laadukkaampaa. [Ruuska, 2001]

4.3. Arviointimenetelmiä

Työmäärien ja niiden suorittamiseen kuluvan ajan sekä kalenteriajan arviointiin on olemassa erilaisia menetelmiä. Arvioiden tukena kannattaa käyttää aiempaa kokemusta vastaavanlaisista projekteista ja tehtävistä. Näiden kahden tekijän summalla päästään mahdollisimman pitävään arvioon. Projektin työmäärien arvioimiseksi pitää tehtävät määrittellä mahdollisimman tarkalla tasolla. Erinomaisena apuna tehtävien määrittelylle on esimerkiksi aiemmin kerrottu projektin osittelu.

4.3.1. Matemaattisia arviointimenetelmiä

Työmäärien arvioissa täytyy huomioida tehtävään, suorittajaan ja ympäristöön liittyvät asiat. Tehtävän laajuus, suorittajan ammattitaito, työvälineet ja tilat vaikuttavat luonnollisesti tehtävän suoritustehokkuuteen. Työmäärä saadaan kertomalla kukin tehtävä tehtäväkohtaisella muuttujalla ja laskemalla kaikki yhteen. Tehtäväkohtainen muuttuja saadaan tehtävän, sen suorittajan ja siihen vaikuttavan ympäristön yhdistelmästä (keskeisimpiä muuttujia). Kullekin muuttujalle määritellään arvo ja ne kerrotaan keskenään (kuva 2.). Joillekin muuttujille on olemassa ohjeellisia viitearvoja, esimerkiksi työn suorittajan kokemuksen mukainen työmääräkerroin. Mikäli tehtävän suorittaja ei ole tiedossa arviointitilanteessa, käytetään keskiarvoja. Työn suorittajan vaikutus työmäärään huomioidaan kertomalla keskimääräinen työmäärä ns. kokemuskertoimella (kuva 2.). [Pelin, 2002; Ruuska, 2001]

Tehtävä $x = a_n$	
Tehtäväkohtainen muuttuja (tehtävä, suorittaja, ympäristö) $x = x_n$	
Työmäärä = $a_1 + a_2x_1 + a_3x_2 + \dots + a_nx_n$	
Kokemus	Työmääräkerroin
Harjoittelija	3,5 – 4,0
Jonkin verran kokemusta	2,0 – 3,0
Ammattitaitoinen	1,0 – 1,5
Ekspertti	0,6 – 0,8

Kuva 2. Työmäärän laskentamalli. [Pelin, 2002]

Työmäärien arvioinnin jälkeen lasketaan tehtävien kestoaika. Kesto saadaan laskettua jakamalla työmäärä tehtävään käytettävissä olevilla henkilöresursseilla (kuva 3.).

Kesto = työmäärä : resurssimäärä

Kuva 3. Työn keston laskentamalli [Pelin, 2002]

Laskennallinen kesto aika muutetaan kalenteriaikaiseksi kestoksi huomioimalla todellinen työaika. Mikäli työllä on yksi kokopäiväinen tekijä, kesto on työmäärä. Osa-aikaisen tekijän ollessa kyseessä, kesto aika luonnollisesti pitenee samassa suhteessa. Resurssin käytettävyyden on syytä merkitä suunnitelmiin.

Kestossa on huomioitava lisäksi viikonloppujen, juhlapyhien, lomien, matkapäivien, koulutusten yms. tilaisuuksien vaikutukset. Aika, jolloin henkilö on estynyt hoitamaan kyseistä tehtävää, vähennetään tehokkaasta työajasta. Normaali työviikko on 5 henkilötyöpäivää (htpv). Kalenteriaikaa laskettaessa tästä on vähennettävä poissaolot, jolloin varsinainen kalenteriaika tehtävän suorittamiseen pitenee. Ruuskan [2001] mukaan henkilön realistinen maksimikäytettävyyden on normaaliolosuhteissa (ilman tausta- ja työtilannetekijöitä) 70 – 80%, eli 3 - 4 päivää/viikko. Usein henkilön tehokas työaika lasketaan liian suureksi.

Työmäärä saadaan siirrettyä kalenteriajaksi laskemalla käytettävissä olevat työpäivät, jolla jaetaan työpäivien teoreettinen maksimimäärä. Tästä saadaan kerroin, jolla kerrotaan työmääräarvio, lopputuloksena työhön kuuluva kalenteriaika (kuva 4.). [Ruuska, 2001]

Kerroin =

Työpäivien teoreettinen maksimi: Käytettävissä olevien työpäivien maksimi

Kalenteriaika = Työmääräarvio (htpv) x kerroin

Kuva 4. Työmäärän muuttaminen kalenteriajaksi, laskentamalli.

[Ruuska, 2001]

Usein laskennallinen projektiaikataulu on sille asetettujen reunaehtojen suhteen todellisena kalenteriaikana liian pitkä. Kalenteriaikaa voidaan lyhentää limittämällä kriittisiä tehtäviä, mikäli se on mahdollista. Kriittinen tehtävä on sellainen tehtävä, jossa kokonaispelivaraa ei ole, eli tehtävä ei voi siirtyä vaarantamatta projektin valmistumista. Projektin lyhin mahdollinen suoritamisaika määräytyy kriittisen polun mukaan. Kriittinen polku on kriittisten tehtävien muodostama ketju toimintaverkon alusta loppuun. Tehtävien yksityiskohtaisesta kartoituksesta projektin alusta loppuun selviää kriittinen tehtäväketju, eli kriittinen polku ja pelivarat.

Kestoa ei ole mahdollista lyhentää, kun yhtäkään kriittisen polun tehtävää ei voida enää kohtuullisin kustannuksin lyhentää. Optimaalinen tilanne on saavutettu, kun kustannus-hyöty-suhde on paras mahdollinen. Mikäli kalenteriaikaa ei voida lyhentää, on projektiaikataulun tavoitteita muutettava tai projektin tavoitteita karsittava. [Ruuska, 2001]

Optimoinnin lisäksi voidaan hankkia lisää henkilöresursseja tai jopa karsia tehtäviä. Toisaalta voidaan harkita, onko mahdollista hankkia kokeneempi ja ammattitaitoisempi tekijä jollekin tehtävälle, tai muulla tavoin tehostaa sen suorittamista. Resurssien lisääminen on yleensä paras lääke aikataulun parantamiseen. Muissa menetelmissä on melkoiset riskit projektin onnistumisen kannalta. Henkilöresurssien kasvattaminen lisää kulujen lisäksi projektinhallinnallisia haasteita mm. kommunikoinnin ja seurannan suhteen. [Ruuska, 2001]

Työn keston suhteen on kehitetty melko monimutkaisiakin laskentakaavoja. Näillä laskentamalleilla voidaan aikataulusuunnittelun lisäksi tyydyttää erilaisia projektinhallinnallisia tavoitteita, kuten esimerkiksi kustannusten minimointina. [Ke and Liu, 2005]

4.3.2. PERT-menetelmä

Kriittisen polun metodissa toimintojen odotetaan kestävän täsmällisen ajan ilman vaihteluita. Näin ei kuitenkaan useimmiten ole, vaan eri tehtäville voidaan arvioida kestoajan todennäköisyys. Tällaisessa työmäärien arvioinnissa voidaan käyttää apuna PERT-menetelmää (program evaluation and review technique), jonka avulla voidaan laskea todennäköinen odotettavissa oleva työmäärä.

Todennäköinen työmäärä lasketaan jokaiselle tehtävälle käyttämällä suunnittelijan aiempaa kokemusta ja muuten hankittua tietoa eri toimenpiteiden aika-arvioista, sekä selvittämällä kriittinen polku. Todennäköisen työmäärän odotusarvon keskiarvo lasketaan optimistisimman, pessimistisimmän ja todennäköisen työmäärä kerrottuna neljällä summalla, joka edelleen jaetaan kuudella (kuva 5). PERT-kaavio (toimintoverkko) on graafinen esitys tehtävien riippuvuuksista. [Joensuun yliopisto, 2006; Ruuska, 2001, Huotari *et al.*, 1995]

Tässä menetelmässä oletetaan, että tehtävien kestoajat ovat toisistaan tilastollisesti riippumattomia. Tällöin voidaan laskea kriittisten toimintojen odotusajat ja varianssit yhteen, jolloin saadaan myös koko hankkeen arvioitu kesto aika eli estimaatti (μ) ja varianssi (σ^2). Keskihajonta kuvaa arvojen keskimääräistä etäisyyttä keskiarvosta ja on tärkein hajonnan mitta. Varianssi on keskihajonnan neliö ja sillä kuvataan muuttujan vaihtelun mittaa. Keskeisraja-arvolauseen perusteella oletetaan, että toimintojen määrä on riittävän suuri ja ne ovat tilastollisesti riippumattomia. Tällöin myös kokonaisaika ja varianssi ovat normaalisti jakautuneita. Pienissä hankkeissa, joissa on vähän toimintoja, oletamus ei aina pidä paikkaansa. Tämä tulee muistaa tuloksia tulkittaessa. Juuri nämä perusolettamukset mahdollistavat hankkeiden kestoajojen todennäköisyyslaskelmat. Kunkin havainnon standardipoikkeamien määrän etäisyys keskiarvon estimaatista voidaan erikseen laskea. Tämän avulla voidaan edelleen määrittää normaalijakauman (Gaussin käyrä) taulukosta havainnon esiintymistodennäköisyys. [Joensuun yliopisto, 2006; Tilastokeskus, 2006]

<p>Todennäköisen työmäärän keskiarvon odotusarvo</p> <p>a = optimistisin työmääräarvio</p> <p>b = pessimistisin työmääräarvio</p> <p>c = todennäköisin työmääräarvio</p> <p>t = todennäköinen odotettavissa oleva työmäärä, keskiarvon odotusarvo</p> <p>$t = (a + 4c + b)/6$</p> <p>Varianssi (v)</p> <p>$v = ((b-a)/6)^2$</p> <p>Standardipoikkeamian määrä</p> <p>x = havainto</p> <p>μ = hankkeen arvioitu kesto aika</p> <p>σ = varianssien summa ^{1/2}</p> <p>$Z = (x-\mu)/\sigma$</p> <p>Havainnon esiintymistodennäköisyys</p> <p>P(t), % = standardipoikkeaman avulla laskettu hankkeen kestoajan todennäköisyys</p> <p>Φ kuvaa normaalijakauman kertymäfunktiota</p> <p>$P(t), \% = \Phi(-Z)$</p>

Kuva 5. PERT-menetelmän laskentamalleja [Joensuun yliopisto, 2006; Ruuska, 2001]

4.3.3. Muita arviointitekniikoita

Arviointitekniikoita on useita. Edellä mainittujen lisäksi on olemassa myös melko epätieteellisiä menetelmiä, kuten Stetson, 'takaperoinen ajoitus', Parkinsonin menetelmä, Mutu, Rahi ja Metu sekä erilaisia nyrkkisääntöjä.

Stetson-menetelmässä pohditaan tehtäväkokonaisuutta ja käytettävissä olevia resursseja. Kokonaistyömäärästä ja projektin kestosta tehdään karkeat arviot. Tämän jälkeen pyydetään joltakin toiselta henkilöltä pikainen arvio samasta asiasta. Arvioita vertaillaan keskenään lopullisten työsuunnitelmien ja toteutumien kanssa. Näiden erot analysoidaan ja niistä yritetään ottaa oppia seuraavaa kertaa varten. Tätä menetelmää kutsutaan myös hatusta vetämiseksi. Näin ollen menetelmä ei luotettavuudeltaan vastaa kuin parasta arviota mutu-peri-

aatteella (mutu eli "musta tuntuu"). Mikäli kokemus vastaavanlaisista hankkeista on kattavaa, voidaan tälläkin metodilla tietysti päästä kohtuullisen lähelle karkean tason realistisia arvioita. Kokonaisen projektin aikatauluttaminen tällä periaatteella on mielestäni melkoinen riski. [Ruuska, 2001] Takaperoinen ajoitus perustuu siihen, että projektille asetetaan ensin valmistumispäivä. Sen jälkeen käytettävissä oleva aika jaetaan projektin tehtäville. Tämä menetelmä suosii tiukkoja ja epärealistisia aikatauluja ja on projektiorganisaatiolle, varsinkin suorittavalle tasolle melko raskasta ja stressaavaa. Usein takaraja on annettu vasten parempaa tietoa kyseisen kokonaisuuden realistisemmasta suoritusajasta. Lopputuloksen laatu kärsii kohtuuttoman aikataulun johdosta. Itse olen toiminut useammankin kerran tämän tapaisessa projektissa. Silloin joudutaan tekemään useita kompromisseja, jotta päästään mahdollisimman lähelle tavoiteaikataulua. Tämä menettelytapa ei ole suositeltavaa, ellei se ole hyvin perusteltua kuten esimerkiksi vuosituhannen vaihteen tai eurojen käyttöön siirtymisen yhteydessä. [Pelin, 2002]

Parkinsonin menetelmässä sovelletaan ns. Parkinsonin lakia, jossa työ vie sille varatun ajan. Varattu aika saadaan kertomalla projektin kesto käytettävissä olevilla henkilöresursseilla. Tätä menetelmää käyttäen saadaan työhön kuluvan ajan alaraja, mutta todellisuudessa projektiin voi kuulua työtä paljon enemmän. [Pelin, 2002]

Mutu, Rahi ja Mehu ovat intuitiivisia arviointimenetelmiä. Mutu perustuu omakohtaiseen arvioon, musta tuntuu. Rahi kuulostaa jo humoristisemmalta, ravistetaan hihasta. Tätäkin joudutaan joskus tekemään, niin surkuhupaisalta kuin se kuulostaakin. Metussa arvio tehdään ryhmässä, meistä tuntuu. Näitä arviointimenetelmiä voivat käyttää kohtuullisen hyvin kokeneet tekijät, joilla on kokemusta vastaavanlaisista projekteista ja asioista. Näihin päädytään varmaan useimmin silloin, kun konkreettista tietoa ei ole tarpeeksi tai tarkempaa arviointitekniikkaa ei ole käytettävissä. Nämä menetelmät voivat toimia hyvänä tukena ja "varmistuskeinona" arvioitaessa eri laskentamenetelmillä saatujen arvioiden realistisuutta. [Ruuska, 2002]

4.4. Suunnitelmien vertailu

Kun työmääristä on tehty tarkemmat suunnitelmat, niitä voidaan verrata viitearvoihin. Ruuskan [2001] mukaan projektin työvaiheet jakautuvat seuraavasti:

- määrittely ja suunnittelu 30 %
- toteutus 40 %
- testaus 30 %.

Projektin luonne vaikuttaa näiden työvaiheiden jakautumiseen. Uutta tehtäessä määrittelyyn ja suunnitteluun kuluu yleensä huomattavasti enemmän aikaa kuin olemassa olevan tietojärjestelmän pienkehityksen suunnitteluun tai

ylläpitotyöhön. Lisäksi eri työvaiheiden kuormittavuus vaihtelee mm. yhteydenpidon vaadittavan määrän mukaan, joka esimerkiksi määrittely- ja suunnitteluvaiheessa on suurempaa (palaverit yms.). Projektin vaiheista on olemassa erilaisia kokemukseen perustuvia prosenttijakoja tarkemminkin tasolla. Tämä vertailutekniikka on vain suuntaa-antava. Viitearvojen avulla voidaan arvioida ollaanko arvioissa realistisella tasolla vai ei. Vertailu on hyvä tarkistus suunnitelmien pitävyydestä ainakin karkealla tasolla. Täytyy kuitenkin muistaa, että jokainen projekti on erilainen. [Pelin, 2002, Ruuska, 2001]

4.5. Aikalaskenta

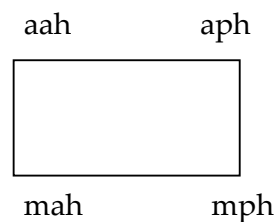
Aikalaskennalla saadaan laskettua kunkin tehtävän aikaisimmat ja myöhäisimmät mahdolliset alkamis- ja päättymisajat ja pelivarat (kuva 6 ja 7). Aikalaskenta tehdään käyttämällä aikaisimman- ja myöhäisimmän alkamis- ja päättymisajan kohdan merkintöjä lohkoverkossa. Lohkoverkko on toimintaverkko, jossa tehtävät esitetään lohkoilla, suljetuilla kuvioilla (esim. suorakaide) ja riippuvuudet nuolilla. Aikaisimman alkamis- ja päättymishetken avulla nähdään projektin pisin tehtäväketju eli kriittinen polku ja kokonaiskesto.

aah = tehtävän aikaisin mahdollinen alkamishetki

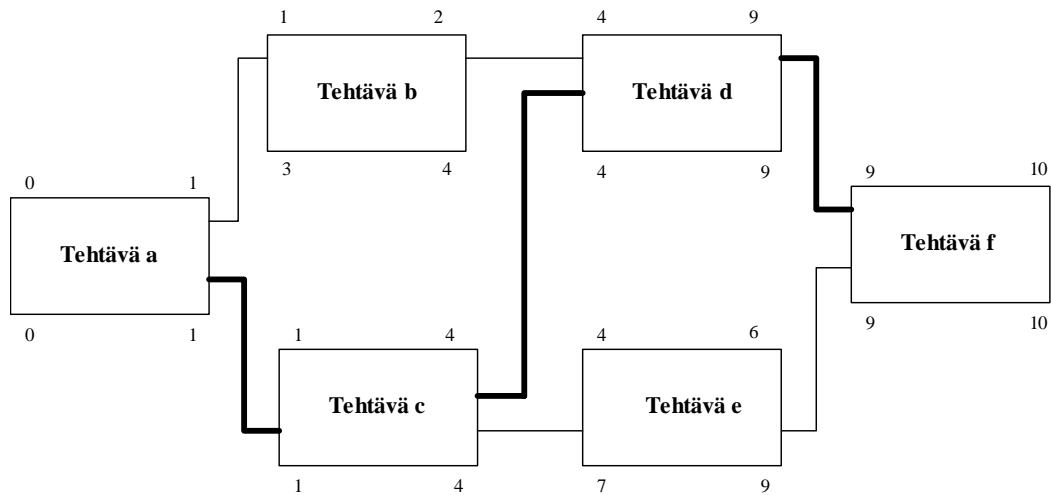
aph = tehtävän aikaisin mahdollinen päättymishetki

mah = tehtävän myöhäisin mahdollinen alkamishetki

mph = tehtävän myöhäisin mahdollinen päättymishetki



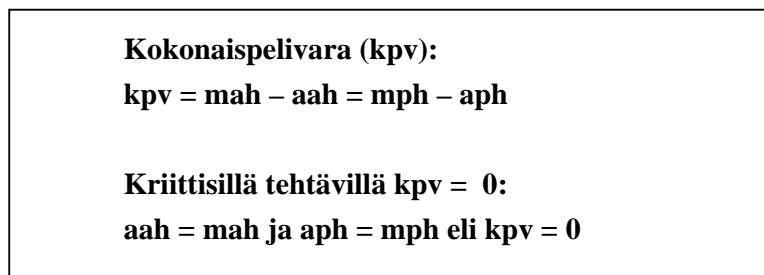
Kuva 6. Lohkoverkon yksittäisen lohkon aikalaskenta. [Ruuska, 2001]



Kuva 7. Esimerkki täydennetystä lohkoverkosta, kriittinen polku paksummalla.

[Ruuska, 2001]

Pelivara on aikaväli, jonka sisällä tehtävän aloittamista tai päättämistä voidaan siirtää myöhemmäksi. Kokonaispelivara on aikaväli, jonka verran tehtävän aloittamista voidaan tarvittaessa siirtää siirtämättä projektin valmistumista. Tehtävän kokonaispelivara voidaan laskea myöhäisimmän alkamis- ja päättämishetken avulla (kuva 8). Kriittisillä tehtävillä ei ole kokonaispelivaraa. Vapaa pelivara on aikaväli, jonka sisällä tehtävän aloittamista voidaan tarvittaessa siirtää ilman, että minkään muun tehtävän aloittaminen siirtyy. [Pelin, 2002, Ruuska, 2001]



Kuva 8. Kokonaispelivara ja kriittisten tehtävien toteaminen.

[Ruuska, 2001]

5. Aikataulusuunnittelu

Projektin aikataulun pitävyys on erittäin tärkeää useastakin eri syystä. Alussa kerrottiin suunnitelman merkityksestä koko projektin kannalta. Projektin toteu-

tuksen kestolla on vaikutuksia mm. investointien tuottoon ja yrityksen kilpailukykyyn. Aikataulun merkitys voidaan usein arvioida myös rahassa. Hyvin toteutunut projekti on yksi projektin onnistumisen mittareista ja vaikuttaa tietenkin myös sen kokonaiskustannuksiin. Lisäksi on merkitystä mm. mahdollisilla myöhästymissakoilla, projektiin investoidun pääoman korkomenoilla, välillisillä kustannuksilla (esim. tilajärjestelyt, laiteresurssit), projektin tuloksen tuotosta (esim. järjestelmä saadaan käyttöön aikaisemmin) sekä markkinaedulla, kun tuote saadaan käyttöön/markkinoille ennen kilpailijoita. Toimittajan kannalta on ehdottoman tärkeää että aikataulun pitävyyteen voi luottaa. Laadukas tulos on parasta mainosta, työ puhuu puolestaan. [Pelin, 2002]

Hyvä aikataulun tekeminen on haasteellista ja aikaa vievää. Suunnittelu- menetelmiä on paljon, mutta osaltaan aikataulujen suunnittelu on hyvin pitkälle arvioperusteista. Suunnittelua varten hankitaan mahdollisimman paljon taustatietoa. Kun yhdistetään kattava taustatieto ja tilanteeseen sopiva arviointiteknikka, päästään mahdollisimman realistiseen lopputulokseen. [Pelin, 2002, Ruuska, 2001]

Miten hyvin arvioita yleensä pystyy tekemään, riippuu paljon aikaisemmasta projektityökokemuksesta, kokemuksesta vastaavanlaisista tehtävistä sekä kyseisen liiketoiminta-alueen ja tekniikoiden sekä arviointimenetelmien tunteuksesta. Erittäin hyvä tapa kasvattaa tätä tietämystä on haastatella vastaavanlaisissa projekteissa toimineita henkilöitä ja todeta miten kauan erilaisten asioiden tekemiseen todellisuudessa kuluu. Suunnittelijan täytyy tietää, mitä ollaan tekemässä ja miten se aiotaan toteuttaa. Mitä pienemmällä suunnittelulla aikataulut rakennetaan, sitä stressaavampi projekti on yleensä tiedossa. Suunniteltu aikataulu on keskimäärin vain noin 80% toteutuneesta. Kuten aiemmin todettiin, aikataulujen toteutumiseen vaikuttaa hyvin moni seikka, täsmälliseen arvioon on vaikea päästä. [Chapman, 1997]

Useimmiten aikataulujen epäonnistuminen johtuu epärealistisista aikatavoiteista, liian passiivisesta projektin toteutuksesta tai suunnitelmien puutteellisuudesta. Epävarma osaaminen, laitteet ja teknologia ovat myös omiaan pidentämään projektiin kuluvaan aikaa. Suunnitelmissa on hyvä varautua ns. kohtuullisiin varmuusmarginaaleihin. Mikäli kaikki aikataulumuuttajat on kiristetty äärimmilleen, on hyvin todennäköistä että projekti ei pysy aikataulussa. [Chapman, 1997; Ruuska, 2001]

5.1. Aikataulun laadinta

Aikataulu rakennetaan vaiheittain. Kun projekti on ositettu, laaditaan yksityiskohtaisempi tehtäväluettelo. Tehtävät kartoitetaan yleensä ylhäältä alaspäin projektin alusta loppuun asti. Kunkin alueen vastuuhenkilöt ja asiantuntijat

ovat parhaita henkilöitä suunnittelemaan ja arvioimaan oman osa-alueensa aikatauluja. Tehtävien ositustarkkuus aikataulun eri vaiheissa riippuu myös projektin kestosta. Pitkän projektin (yli vuosi) loppuvaiheen tehtäviä tarkennetaan usein myöhemminkin.

Tehtävien työmäärät ja kesto arvioidaan. Tämän jälkeen selvitetään tehtävien suoritusjärjestys ja niiden riippuvuudet. Tehtäville tehdään resurssien allokointi. Arvioita kannattaa tehdä useammalla menetelmällä, tulokset tukevat toisiaan. Suunnittelu tehdään ns. todennäköisyysajattelulla, jossa arviointivirheet kumoavat toisensa. Usein varsinainen aikataulu tehdään jollakin työkalulla, esimerkiksi projektinhallintaohjelmistoilla kuten Microsoftin MS-Projectilla. Aikataulu kuvataan usein aikajanoilla, joissa eri työtehtävät on sijoitettu aikajanoiksi kalenterille. Toimintaverkkomenetelmät (esim. PERT ja GANT) tukevat janakaavioita, koska niissä on todettavissa paremmin myös tehtävien väliset riippuvuudet ja kriittisyydet.

Henry Gant kehitti janakaavion jo 1900-luvun vaihteessa, ns. Gantin kaavio. Kaaviolla kuvataan yleensä koko projektin aikataulu. Aikataulu sisältää projektin tärkeimmät vaiheet työtehtävineen. Kukin tehtävä kuvataan aikajanalla erikseen. Ensiksi suoritettavat tehtävät alkavat vasemmasta reunasta. Kunkin tehtävän janan alkupää kuvaa tehtävän aloitusajankohtaa ja loppupää tehtävän arvioitua päättymisaikaa. Janan pituus kuvaa tehtävän kesto. Tehtävien riippuvuussuhteet kuvataan (esim. nuolilla). Kaavion avulla on mahdollista löytää myös ns. kriittinen polku.

Kun aikataulu on rakennettu, sitä analysoidaan. Lopuksi aikataulu hyväksytään ja siihen sitoutuu koko projektiorganisaatio sekä johto. Muutoksia tulee usein projektin aikana, on hyvin tärkeää että aikataulua ylläpidetään. Projektin joka vaiheessa on tiedettävä missä ollaan menossa ja mitä jatkossa tapahtuu.

Yleisaikataulussa on mukana yleensä koko projekti. Se sisältää päätehtävät eri osa-alueilla, sopimuspisteet ja keskinäiset liittymät. Päivätarkkuudella tehty aikataulu tehdään usein noin kolme kuukautta eteenpäin. Tätä aikataulua tarkennetaan ja jatketaan noin kerran kuukaudessa. Yksityiskohtainen viikko-aikataulu ja tehtäväluettelo tehdään usein pariin viikoksi kerrallaan. Tämä aikataulu tarkistetaan ja käsitellään viikoittain. Kullakin tehtävällä on tehtäväkuvaus ja vastuuhenkilö. Pelin [2002] mainitsee suunnittelutehtävien erittelyn nyrkkisäännöksi että ne tehdään 1 – 6 viikon tarkkuudella. Tehtävän suorittajan vaihtuessa tehtävä eritellään aina.

Aikataulussa otetaan huomioon myös erilaiset laite- ja materiaalitoimitukset, varsinkin kriittiset. On hyvä huomioida myös muut ajallisesti vaikuttavat tehtävät, kuten luvat, päätökset ja sopimukset, hyväksynyt, toimitusajat, kuljetukset ja pakolliset seisonta-ajat. Aikataulut ja

tehtävälueketlot voidaan jakaa hierarkkiseksi kokonaisuudeksi. Kaikkea ei välttämättä kannata laittaa samaan aikatauluun. [Pelin, 2002]

5.2. Riippuvuudet ja kriittiset polut

Useita projektisuunnittelumenetelmiä kutsutaan verkkoanalyyseiksi. Tällaisia ovat mm. aiemmin käsitelty PERT-menetelmä sekä CPM-menetelmä. CPM (critical path method) on kriittisen polun menetelmä. Nämä menetelmät eroavat toisistaan lähinnä siinä, että CPM perustuu ennalta määrättyihin aikoihin ja PERT:ssä käytetään apuna satunnaislukujakaumia. Verkkoanalyysissa kuvataan projektin tehtävät, suoritusjärjestys sekä tehtävien väliset riippuvuudet nuoli- ja lohkodeigrammein. Näiden menetelmien tarkoituksena on mm. toimia hälytyskelloina kun toimenpiteitä vaativia aikatauluongelmia ilmaantuu. [Järvinen, 2004]

Tehtävien riippuvuus on tehtävien tai tapahtumien välisen järjestyksen määräävä valittu tai ehdoton rajoitus. Ehdoton riippuvuus tarkoittaa tehtäviä, jotka on mahdollista suorittaa vain yhdellä tavalla. Ehdollinen riippuvuus antaa valinnan mahdollisuutta, mutta kahden tehtävän välillä on kuitenkin riippuvuutta. Looginen riippuvuus määrää tehtävien johdonmukaisen suoritusjärjestyksen. Resurssiriippuvuus määrää tehtävien suoritusjärjestyksen joka johtuu annetuista resurssirajoituksista. Mikäli tehtävän suoritus riippuu toisen tehtävän vaiheesta, kyseessä on limitsriippuvuus. Tehtävien resurssiriippuvuus syntyy kun tekijöinä ovat samat henkilöt. Mikäli tehtävän alkamis- ja päättymisajankohta on sidottu johonkin ajankohtaan, kyseessä on kalenteririippuvuus. Kun seuraava tehtävä voi alkaa vasta tietyn ajan kuluttua edellisen loppumisesta, puhutaan viiveriippuvuudesta. Toisistaan täysin irrallisilla tehtävillä ei ole riippuvuussuhdetta.

Toimintaverkko on kuvausmenetelmä, jolla esitetään tapahtumat, tehtävät ja niiden riippuvuudet. Sen avulla voidaan mm. miettiä, mikä on edullisin mahdollinen suoritusjärjestys. Kriittinen polku on selkeästi nähtävissä tässä menetelmässä. [Ruuska, 2001; Pelin, 2002]

5.3. Epävarmuustekijät

Suurin osa projektiaikataulutuksen tutkimuksesta ja metodeista olettaa, että kaikki tarvittava tietämys suunnitelmien ja aikataulujen rakentamiseksi on olemassa. Näin ei kuitenkaan useimmiten todellisuudessa ole. Ulkoisten tekijöiden vaikutus on hyvä pitää mielessä. Erilaisten asioiden keskinäinen riippuvuus on myös merkittävä tekijä. Epävarmuus voi syntyä useista eri syistä. Tehtävien suorittamiseen voi mennä enemmän tai vähemmän aikaa kuin on suunniteltu, asiakas voi muuttaa vaatimusmäärittelyjä, kaikki resurssit eivät olekaan saata-

villa, materiaalitoimitukset voivat myöhästellä, määräpäiviä joudutaan muuttamaan, uusia tehtäviä voi ilmaantua ja toisia poistua jne.

Yllätystekijöiden esiintymistä ja vaikutusta on tutkittu paljon. Tähän aiheeseen liittyvä tieto ja menetelmät ovat vielä melko kesken, vaikka yllätysten oletetaan kuluvan projektityön luonteeseen lähes automaattisesti. Herroelen ja Leus [2003] esittelevät viisi yleisintä lähestymistapaa epävarmuustekijöiden hallinnasta aikataulusuunnittelussa. Näitä menetelmiä on tutkittu lähinnä koneellisen aikataulusuunnittelun yhteydessä. [Herroelen and Leus, 2003]

5.3.1. Reaktiivinen aikataulusuunnittelu

Reaktiivinen aikataulusuunnittelu (reactive scheduling) ei pyri käsittelemään epävarmuustekijöitä alkuperäisessä aikataulusuunnitelmassa. Sen sijaan se keskittyy alkuperäisen aikataulun korjaamiseen odottamattomien aikatauluun vaikuttavien asioiden ilmaantuessa, reagointiin. Alkuperäisessä aikataulussa ei ole suunniteltuja varokeinoja sen pettäessä.

Reaktiivinen tekniikka voi olla hyvinkin yksinkertainen. Tyypillisimmillään se voi olla tunnettu 'right shift rule' menetelmä (Sadeh et al., 1993; Smith 1994). Tämän säännön mukaan kaikkia tehtäviä joihin aikataulun hajoaminen resursseista tai riippuvuuksista johtuen vaikuttaa, yksinkertaisesti siirretään oikealle eli tuonemmaksi. Tämä menetelmä voi johtaa hyvinkin huonoihin tuloksiin projektin onnistumisen kannalta. Toinen reaktiivisen aikatauluttamisen ääripää on koko olemassa olevan aikataulun siivuttaminen, kun aikataulun pettäessä ryhdytään toimenpiteisiin. Tällöin voidaan jo puhua kokonaan uuden aikataulun tekemisestä. Menetelmän pääpyrkimyksenä on, että uusi aikataulu poikkeaa mahdollisimman vähän alkuperäisestä ja muutoksista johtuva myöhästyminen minimoidaan. [Herroelen and Leus, 2003]

5.3.2. Todennäköisyyteen perustuva projektisuunnittelu

Todennäköisyysteorioihin ja -laskentaan pohjautuva projektisuunnittelu (stochastic project scheduling) on kehitetty mm. mittaamaan projektin todennäköisyyttä valmistua tietyssä ajassa. Kokonaisen projektin valmistuminen ja tehtävät jotka määrittelevät sen keston, ovat aina kyseenalaisia johtuen tehtävien ja satunnaisten tehtävien vaihtelevasta kestosta. Toimittaja (projektin toteuttaja) voi käyttää tätä menetelmää ennustamaan kuinka todennäköistä projektin toteutus on sovitussa aikataulussa, sekä arvioidakseen omat kykynsä täyttää sopimuksen vaatimukset. Tämä on hyvä varmistaa aina ennen tarjouksen tekemistä toteutettavan kokonaisuuden omistajalle eli ostajalle. Projektin toteutustodennäköisyys on toimittajalle erittäin arvokasta tietoa, mutta tästä menetelmästä hyötyy myös omistaja. Omistaja voi määritellä ja analysoida riskit, joita aikatauluun ja toteutukseen liittyy.

Menetelmässä käytetään mm. tehtävien järjestämistä mahdollisimman optimaalisesti. Projektin työtehtävien ja aikataulun arvioimiseen sekä niiden keskinäiseen järjestämiseen mm. tehokkuuden parantamiseksi ja aikataulun optimoimiseksi on kehitetty lukuisia erilaisia laskentamalleja, algoritmeja. Tällaisia ovat mm. PERT- menetelmän kaltaiset arviointimetodit. [Ruuska, 2001; Herroelen and Leus, 2003]

5.3.3. Sumean logiikan aikataulusuunnittelu

Sumean logiikan aikataulusuunnittelun teoria (fuzzy scheduling) mukaan tehtävien kestoajkojen todennäköisyyksien jakautumista ei tunneta, koska ei ole olemassa tarpeeksi kokemustietoa aiheesta. Usein arviot tehdään asiantuntijoiden toimesta, mutta tehtävät ovat hyvin tilannekohtaisia ja ainutkertaisia. Projektin johto joutuu tekemään päätöksiä asioista hyvinkin epämääräiseltä ja epävarmalta pohjalta. Epämääräisissä tilanteissa sumean logiikan suunnittelu-metodi suosittelee käyttämään ns. sumeita arvoja tehtävien keston mallintamisessa, mieluummin kuin todennäköisyyslaskentaan perustuvia lukuja. Todennäköisyyksien sijaan käytetään tehtävien ositteluja ja edelleen osatehtävän todennäköisyyttä sisältyä johonkin tehtäväkokonaisuuteen.

Rommelfanger (1990) on kehittänyt käytännöllisen tavan selvittää tehtäväkokonaisuuksien sumeiden osa-alueiden tiedot. Tämän mukaan asiantuntijat antavat pessimistisimmän ja optimistisimmän arvionsa joistakin tärkeistä tehtäväkokonaisuuksista määrittelemällä niiden vaihteluvälin. Vaihtelu muodostuu tehtävistä, joiden todennäköisyys esiintyä tehtäväkokonaisuudessa on kohtuullinen,, sekä tehtävistä joiden esiintymistodennäköisyys on suuri. Myös sumean suunnittelumenetelmän metodissa on kehitetty lukuisia laskentamalleja arvioinnin avuksi. Suunnitteluprosessin lopputuloksena saadaan sumea aikataulusuunnitelma, joka sisältää sumeat aloitus- ja lopetusajankohdat tehtäville. Tällaisia aikatauluja on kuitenkin tulkittava varauksellisesti. [Herroelen and Leus, 2003]

5.3.4. Ennakoiva aikataulusuunnittelu

Ennakoiva aikataulusuunnittelu (proactive scheduling) varautuu aikataulujen muutoksiin mm. pitämällä vararesursseja varattuna eri tehtävien hoitamiseen, mikäli niiden edistyminen sitä vaatii. Alkuperäiseen aikatauluun lisätään suojatut kestoajat.

Puhdas resurssien vajaakäyttö on melko epärealistista ja kallista. Lisäksi projektityön ongelmat eivät yleensä ole näin yksinkertaisesti ratkaistavissa. Tilastojen perusteella aineellinen suojele yleensä pidentää tehtävien kestoja. Yleensä tehtävät, jotka edellyttävät vararesurssien käyttöä keskeytystilanteessa, kestävät kauemmin, koska samalla joudutaan selviytymään itse tehtävien ja

mahdollisesti projektin keskeytyksestä. Perustuen tilastoihin tehtävien suojattu kesto-aika on yhtä suuri, kuin alkuperäinen arvio lisättynä keskeytyksillä ja niiden aiheuttamilla viipeillä, joita tehtävän suorituksen aikana voidaan odottaa. Viive syntyy keskimääräisestä ajasta joka menee itse epäonnistumiseen ja ajasta joka kuluu asian korjaamiseen. Tämän vuoksi tämä lähestymistapa ei sovellu projekteihin, joissa pääresursseina ovat ihmiset.

Metodia on pyritty parantamaan mm. käyttämällä tehtävien aikaisimman ja myöhäisimmän mahdollisen alkamis- ja päättymisajankohdan periaatteita ja löytämällä niiden avulla optimaalisen suunnitelman projektin toteutukselle. Myös tämän metodin tueksi on kehitetty useita laskentamalleja. [Herroelen and Leus, 2003]

5.3.5. Herkkyysanalyysi

Herkkyysanalyysiin (sensitivity analysis) perustuva aikataulusuunnittelu keskittyy "mitä jos?" -tyyppisiin kysymyksiin. Menetelmä pyrkii antamaan vastauksia useisiin kysymyksiin: mitkä ovat muutosmarginaalien rajat, jotta kokonaisuus pysyy optimaalisena, mikä on optimaalinen kustannus jos muutoksia ilmaantuu, mikä on muutoksen jälkeen optimaalinen ratkaisu, milloin alkuperäinen aikataulu pysyy optimaalisena, milloin kohteen toiminta pysyy optimaalisena, millaisia herkkyysanalyyssejä kannattaa käyttää jotta ratkaisujen vankkuus on optimaalinen ja millaisia herkkyysanalyyssejä voidaan tehdä ilman yksityiskohtaisia tietoja ratkaisusta? Nämä ovat hyvin mielenkiintoisia kysymyksiä tulevaisuuden tutkimukselle. Näihin on myös kehitetty monia arviointi- ja laskentamalleja

Toinen mielenkiinnon kohde on vielä melko tutkimaton alue. Miten määrittellään, mitkä parametrimuutokset ovat sellaisia, joiden seurauksena ei jouduta uudelleenaikatauluttamiseen, vaan selvittää pienillä suunnitelmien korjauksilla, esimerkiksi right shift -menetelmällä. [Herroelen and Leus, 2003]

6. Projektin hallinta ja seuranta

Projektinhallinnan yksi tärkeimpiä tehtäviä on seuranta, joka on kriittisen tärkeää onnistumisen kannalta. Seurannan tulee olla säännönmukaista ja tehokasta. Projektiseurantaa tehdään useilla eri tavoilla, yksi keskeisimmistä seurantavälineistä on aikatauluseuranta.

Kun poikkeuksia havaitaan, niihin on puututtava välittömästi. Ongelmiin tartutaan heti ja niihin etsitään mahdollisimman tehokas ratkaisu. Näin vältetään vielä suuremmilta vahingoilta. Heti projektin alkuvaiheessa olisikin hyvä aina tehdä ns. riskikartoitus ja riskinhallintasuunnitelma. Kun mahdolliset ongelmat kartoitetaan etukäteen, niihin ollaan paremmin valmistauduttu.

Ongelmat tunnistetaan ehkä helpommin ja on olemassa jo valmiita ratkaisumalleja. Erilaiset yllätykset kuuluvat myös projektityön luonteeseen, näihin onkin jo hankalampi varautua mutta niiden olemassaolo on syytä tiedostaa. Samat projektinhallinnalliset ongelmat toistuvat kerta toisensa jälkeen. Aikaisemmista kokemuksista ja varsinkin virheistä tulee oppia.

Aikatauluseuranta perustuu kolmen erillisen aikataulusuunnitelman versioon. Perusaikataulu eli alkuperäinen suunnitelma säilyy muuttumattomana koko projektin alusta loppuun. Toteutuma-aikataululla kuvataan mitä tarkasteluajankohtaan mennessä on tapahtunut. Aikataulujen seuranta voidaan tehdä mm. viikoittaisella työaikaraportoinnilla, johon kerätään toteutuneet työtunnit ja tehtävien eteneminen. Raportointi tulee tehdä tehtäväluettelon tarkkuudella. Toteutuneita verrataan arvioihin, poikkeamat löydetään ja niiden syyt selvitetään. Tarvittaessa tarkennetaan arvioita, työmenetelmiä tms. asiaan vaikuttavia tekijöitä. Tästä syntyy lopputuloksena kolmas aikatauluversio eli täsmennetty suunnitelma. Tässä on otetaan huomioon jo olemassa olevat tapahtumat ja miten projektin oletetaan jatkuvan. Raportoinnille on olemassa valmiita työkaluja, mm. Microsoftin MS-Project. Kattavalla raportoinnilla pysytään tilanteen tasalla ja kerätään kallisarvoista tietoa tulevien projektien varalle. [Huotari et al, 1995, Pelin, 2002]

7. Onnistunut projekti

Projektin onnistuminen arvioidaan sille asetettujen tavoitteiden perusteella, miten hyvin valmis tuote vastaa tavoitteisiin. Onnistumista ei mitata pelkästään hyvällä tuotteella, vaan että se on aikaansaatu sovitussa aikataulussa ja sovitussa budjetissa. Laadun käsite keskittyy usein liikaa sisällölliseen laatuun, tällöin vaarana on aikataulujen ja budjetin ylittyminen. Aikataulun ja budjetin onnistumista on helpompi mitata, koska ne ilmoitetaan mitattavissa olevilla suureilla. Sisällöllisen laadun tavoitteita on hankalampi mitata, koska ne ovat osittain arvostuskysymyksiä. Tuote voi olla määritelmien mukainen, mutta ei silti aivan sitä mitä on oikeasti haluttu. Projekti on onnistunut jos odotukset ovat täyttyneet. Projektipäällikkö vastaa projektin tavoitteiden täyttymisestä. [Pelin, 2002]

Ruuska [2001] kuvaa kolmen tekijän samanaikaista hallintaa triangelidraamana. Projektin tulostavoitteen muodostavat lopputuotteen laatu, aikataulu ja kustannukset. Muutos missä tahansa näistä kolmesta tekijästä vaikuttaa aina kahteen muuhun. [Ruuska, 2001]

8. Yhteenveto

Projektin aikataulusuunnittelu kannattaa aina tehdä hyvin huolellisesti. Kaikki projektin kulkuun vaikuttavat tekijät on syytä huomioida suunnitelmia tehtäessä. Varsinainen suunnittelu on hyvä jättää mahdollisimman lähelle suoritettavaa tasoa, koska he tuntevat parhaiten konkreettiset työmääriin ja keston vaikuttavat tekijät. Projektin toteutus ja kokonaiskesto voidaan optimoida kiinnittämällä ammattitaitoisimmat ja sopivimmat resurssit kullekin tehtävälle. Työtehtävien riippuvuussuhteet ja siten keskinäinen työjärjestys on myös suunniteltava hyvin. Huolellisella suunnittelulla ollaan lähempänä onnistunutta projektia, mutta muutosten huomioiminen, suunnitelmien ylläpidon ja seurannan osuus on myös hyvin merkittävä.

Eri suunnittelu- ja arviointimenetelmiä käyttäen voidaan päästä hyvinkin realistisiin aikataulusuunnitelmiin. Lopullista ratkaisukeinoa ei varmasti koskaan löydetä, joka takaisi projektin onnistumisen. Onnistuminen on kiinni lukuisista asioista, mutta suunnittelun tärkeyttä ei voi kyllin korostaa. Sano taanko, että hyvin suunniteltu ja hallittu projekti lisättynä onnellisten yhteensattumien summalla johtaa erittäin onnistuneeseen lopputulokseen.

Viiteluettelo

- [Chapman, 1997] James R. Chapman, *Project Schedule Estimating*, 1997. Available as http://www.hyperthot.com/pm_sked.htm
- [Herroelen and Leus, 2003] Willy Herroelen and Roel Leus, Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research* **165** (2005) 289 – 306.
- [Huotari *et al*, 1995] Jouni Huotari, Matti Kangas ja Paavo Moilanen, *Microsoft Project for Windows, Hyötykäyttäjän opas*. Gummerus, Jyväskylä, 1995.
- [Joensuun yliopisto, 2006] Metsäteknologia. Joensuun yliopisto, Metsätieteellinen tiedekunta, Metsäteknologia, opetusmateriaali, dokumentit, kriittinen polku, 2006. http://www.forest.joensuu.fi/opiskelu/metsatek/opetus/documents/kriittinenpolku_001.pdf
- [Järvinen, 2004] Jari Järvinen, *Projektinhallinnan Perusteet 5*, 2004. Saatavilla http://www.multimedia.jyu.fi/jari/dlg_160/.
- [Ke and Liu, 2005] Hua Ke and Baoding Liu, Project scheduling problem with stochastic activity duration times, *Applied Mathematics and Computation* **168** (2005) 342-353.
- [Pelín, 2002] Risto Pelín, *Projektinhallinnan käsikirja*. Gummerus, Jyväskylä, 2002.
- [Ruuska, 2001] Kai Ruuska, *Projekti hallintaan*. Gummerus, Jyväskylä, 2001.

[Tilastokeskus, 2006] Tilastokeskus, Tilastokoulutus, 2.10 Hajontaluvut – keskihajonta ja varianssi. http://www.stat.fi/tk/tp/verkkokoulu/vk/tt/oppitunnit/tt02/tt02_10/view.

Ytimet tietokoneissa

Ville Niemi & Tuukka Pasanen

Tiivistelmä

Tutkielmassa tarkastellaan tietokoneiden ytimiä ja niiden historiallisia piirteitä. Tutkielmassa käsitellään ytimien kolme päätyyppiä: monoliittinen, modulaarinen ja mikroydin. Ytimistä tarkastelemme lähemmin UNIXia, Linuxia ja Darwinia. Lisäksi luomme katsauksen klusteroinnin ja ytimien yhtymäkohtaan.

Avainsanat ja -sanonnat: UNIX, Linux, Darwin, ydin, kernel, monoliittinen ydin, modulaarinen ydin, mikroydin, Apple, Microsoft, klusterointi, virtuaalisointi

CR-luokat: D.4.1, D.4.4, F.1.1, I.2, K,2

1. Johdanto (VN & TP)

Monet nykyisistä standardeista kuten FTP, sähköpostiliikenne ja WWW ovat syntyneet UNIXin mukanaan tuomasta mahdollisuudesta monen ohjelman suorittamiseen yhtä aikaa. Ideana oli, että pienet yksikertaiset ohjelmat suorittavat pienen osan kokonaisuutta, niitä yhdistämällä pystyttiin luomaan sitten monimutkaisia kokonaisuuksia. UNIX on toiminut POSIX-standardin kautta suunnan näyttäjänä monelle tietotekniikan kehitys askeleelle. Näihin kuuluvat myös Linux ja Darwin, joita tutkimuksessamme tullaan käsittelemään.

Itsessäänhan UNIX on vain tuotemerkki, jonka omistaa tällä hetkellä SCO UNIX [SCO, 2005]. Monesti kuitenkin UNIXilla viitataan tietyn tyyppisesti rakennettuun käyttöjärjestelmään, kuten tässä tutkielmassa, sekä POSIX-standardia noudattavaan ytimeen [Tanenbaum, 1992]. UNIX on jakautunut ajan mittaan hyvin moneksi erilaiseksi ytimeksi, joista osa on ollut menestyneempiä kuin toiset. Yksi uudemmista hybridi ytimistä on Applen käyttämä Darwin. Luonteensa vuoksi Darwin on otettu tutkielmaamme edustamaan ytimien yhtä yleisintä tyyppiä, mikroytimiä.

Toinen tutkittava ydin on Linux, joka pyrkii toteuttamaan UNIXin kaltaisen ytimen siten, että POSIX standardin mukaan kirjoitetut ohjelmat ovat ytimen kanssa lähdekooditasolla yhteensopivia. Lisäksi Linux-ytimessä yhdistellään myös innovatiivisella tavalla kahden erityylisten ytimen ominaisuuksia.

Yleisesti ottaen ytimet vaikuttavat, vaikkakin pinnan alta, melko määrävästi nykyiseen tietotekniikkaan. Tällä perusteella niiden tutkiminen auttaa ymmärtämään monia

tietotekniikkaan liittyviä kysymyksiä, jotka muuten jäisivät avoimiksi tai vaille riittävää selitystä. Tämän tutkimuksen tavoitteena on antaa yleiskuvaa siitä, mitkä seikat ovat vaikuttaneet tietotekniikan nykyiseen tilaan.

2. UNIX - Monoliittinen ydin (VN &TP)

Ydin on ohjelma, joka käynnistyy ensimmäisenä tietokoneessa virran päälle kytkemisen ja laitteiston alustamisen jälkeen [Wikipedia, 2005]. Perinteisesti se on ladattu massamuistilaitteelta muistiin ja tämän jälkeen koneen hallinnan on ottanut kokonaan muistiin ladattu ydin. Monoliittinen ydin on ydintyyppi, jossa kaikki tarvittava on yhdistetty yhdeksi ajettavaksi ohjelmaksi.

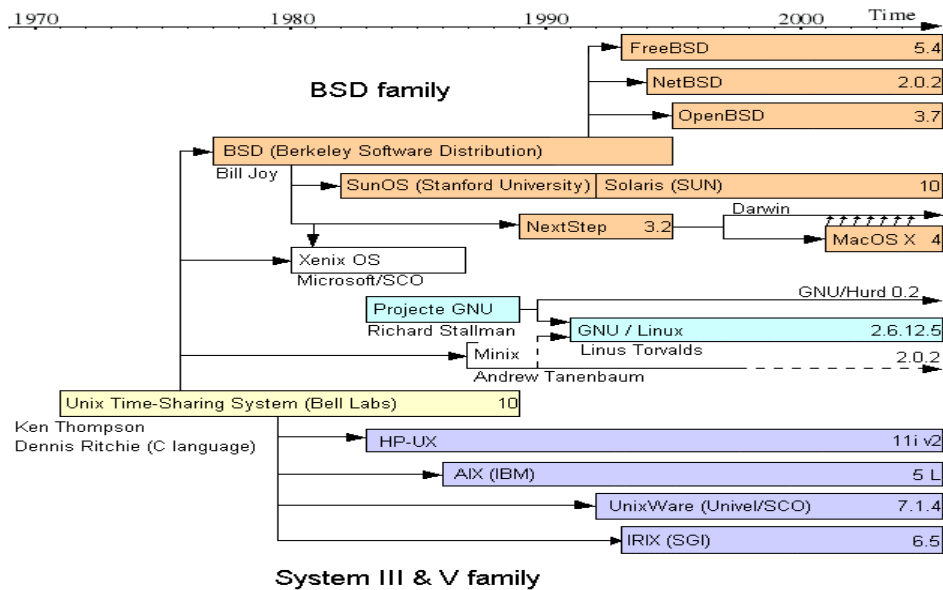
Käynnistymisensä jälkeen ydin suorittaa itsensä loppuun alkuperäisenä ilman, että siihen voidaan lisätä uusia ominaisuuksia. Kuitenkin ytimen tiloihin voidaan vaikuttaa, mikä tarkoittaa sitä, että sitä voidaan ohjata esimerkiksi näppäimistöllä. [Wikipedia, 2005] Idea yhdestä yhtenäisestä ajettavasta ajon aikana muuttumattomasta ytimestä eroaa esimerkiksi mikroytimestä, jota tullaan käsittelemään myöhemmin.

2.1 Historian havinaa (TP)

Tutkiessamme UNIXin historiaa huomasimme, että se on aina kertojansa mukainen. Tämä tarkoittaa, UNIX kaanonin mukaisia historioita on useita, vaikka on olemassa muutamia päivämääriä, jotka suunnilleen kaikki versiot asettavat samoille kohdille tai ainakin samalle ajan jaksolle.

Tämä vaikeuttaa yksiselitteisen historian kirjoittamista oleellisesti. Siksi tämä pieni historia onkin kokoelma Andrew Tanenbaumin [1992] ja suomalaisten Kari Alhon ja Juha Puhakan [1991] kirjoista.

Tässä historiassa UNIXista puhutaan yhtenä tietyntyyppisellä käyttöjärjestelmänä, vaikka sitä se ei todellakaan ole. UNIX on ennemminkin ajatus kuin varsinainen yhtenäinen kenttä tai yksi käyttöjärjestelmä. Kuva 1 selventää tätä seikkaa. Siinä esitetään, miten yhdestä alkulähteestä lähtien UNIX on jakaantunut moniin erilaisiin puihin.



Kuva 1. UNIXin jakautuminen useaksi eri ytimeksi [Wikipedia, 2005f]

2.2 Alku 1945-1955 (VN)

Tanenbaumin [1992] mukaan tietokoneiden historia juontaa kauas aikaan, ennen kuin UNIXia missään muodossa oli edes kuvitelmissa. Vaikka monet pitävätkin UNIXia ensimmäisenä todellisena käyttöjärjestelmänä, näin asia ei varsinaisesti ole. Tanenbaum lisää, että nykyiset tietokoneiden käyttäjät tuskin edes mieltäisivät 1940-1950-luvun tyhjiöputkikoneita tietokoneiksi.

Teknisistä ratkaisuistaan johtuen tyhjiöputkikoneet olivat suurimmaksi osaksi mekaanisia. Niitä ohjelmoitiin konekielellä, joka lävistettiin reikäkortteihin. Nämä reikäkortit käytiin läpi sitten yksitellen ja suoritettiin ohjelma. Tietokoneet sijaitsivat niille pyhitetyissä huoneissa ja niitä käyttämässä oli siihen koulutettu ihminen; hän oli niin kutsuttu operaattori.

2.3 Seuraava sukupolvi (1955-1965) (VN)

Aikanaan tyhjiöputket korvattiin transistoreilla. Uudempi puolijohdetekniikka johti lopulta siihen tietotekniikan vallankumoukseen, jonka jälkimainingeissa tänäkin päivänä eletään. Koneita alettiin ohjelmoida Fortranilla [Alho ja Puhakka, 1991]. Tanenbaumin [1992] toteaa, että reikäkortit kyllä säilyivät, mutta kieli vaihtui. Fortranin rinnalla käytettiin symbolista konekieltä, eli assembleria. Symbolisen konekielen muuttaminen koneesta toiseen oli tuskaista, ellei jopa monessa tapauksessa mahdotonta. Tämä taas johtuu siitä, että symbolinen konekieli on monesti vain itsensä kanssa yhteensopivaa. Toisin sanoen siitä löytyy niin monta murretta kuin on konetyyppejä. [Tanenbaum,

1992].

Tämä tuskastutti tietenkin ohjelmoijia, jotka joutuivat omaksumaan täysin uuden kielen aina, kun uusi konetyyppi tuli käyttöön. Ohjelmat alkoivat olla monimutkaisia, kuten pankki- tai kirjanpitosovelluksia. Näin ollen niiden uudelleen kirjoittaminen oli äärimmäisen kallista ja aikaa vievää puuhaa. [Alho ja Puhakka, 1991] Toinen ongelma oli, että kalliilla koneella ei voinut ajaa kuin yhtä prosessia, aina siihen asti kunnes prosessi oli suoritettu loppuun; muut joutuivat odottamaan vuoroaan. Prosessi saattoi olla pitkä ja yksikin virhe pakotti aloittamaan prosessin uudestaan, mikä taas osaltaan pidensi odotusaikaa. [Tanenbaum, 1992]

2.4 Uuden käyttöjärjestelmän aamunkoi MULTICS (1965-1980) (VN & TP)

Uudet tuulet puhalsivat! Bell Labs, General Electronics (GE) ja Massachusetts Institute of Technology (MIT) kehittivät yhdessä mullistavaa projektia, jonka tuli ratkaista kaikki nämä ongelmat. Sen nimeksi oli tarkoitus tulla MULTICS. [Alho ja Puhakka, 1991] Hienoa siinä oli, verrattaessa vanhoihin järjestelmiin, että se oli eräajojärjestelmä, joten siinä voitiin ajaa monia prosesseja yhtä aikaa. [Tanenbaum, 1992]

Tanenbaumin [1992] mukaan Bellin laboratorio jäädettiin hankkeen. Tämä jätti yhden tutkijan etsimään jotain mielenkiintoista tekemistä. Hänen nimensä oli Ken Thompson. Joutessaan hän ajatteli kirjoittaa MULTICS:n uudestaan symbolisella konekielellä silloiselle minitietokoneelle PDP-7:lle. Tästä vaiheesta on tietenkin olemassa toinenkin versio asiasta. Se menee näin: sattumalta Thompson ja Ritchie kuitenkin löysivät vähän käytetyn PDP-7-laitteen, josta puuttui lähes täysi varusohjelmisto. Samoihin aikoihin Ken (Thompson) työskenteli myös ”Space Travel” -nimisen pelin kanssa ja ryhtyi siirtämään sitä PDP:lle [Alho ja Puhakka, 1991]

Nimi UNIX tulee Tanenbaumin [1992] mukaan erään Ken Thompsonin kollegan vitsistä, jossa hän kuvaili mahdollottoman tuntuista projektia kirjainyhdistelmällä UNICS (Uniplexed Information and Computing Service). Englanniksi se äännetään EUNUCS joka tarkoitti kastroitua MULTICS:ia. Loppujen lopuksi termi UNICS kääntyi muuntoon UNIX.

Tanenbaum [1992] kertoo, että ensimmäinen versio UNIXista tehtiin siten, että se toimi ainoastaan PDP-7-koneessa, sillä se oli kirjoitettu symbolisella konekielellä eli assemblerilla. PDP-7:stä ohjelma siirrettiin uudempaan PDP-11/20-koneeseen ja kirjoitettiin B-kielestä lähtöisin olevalla C-kielellä uudestaan. Tämä takasi, että UNIX oli vihdoinkin siirrettävissä järjestelmästä toiseen. Käyttäjä kirjoitti vain pienen määrän konekielistä koodia ja loput hoidettiin sitten korkeamman tason kielellä, niin kutsutulla kääntäjällä [Alho ja Puhakka, 1991]

2.5 UNIX syntyy (VN)

Tanenbaumin [1992] mukaan 1971 syntyi ensimmäinen UNIX, niin kutsuttu First edition. Tämä oli vielä konekielellä kirjoitettu ja suunniteltu PDP-11/20:lle. Tätä seurasivat sitten C-kielellä 1973 kirjoitetut versiot neljä ja viisi. [Alho ja Puhakka, 1991]. Bell Labs julkaisi UNIXin virallisesti vasta vuonna 1975, silloinkin sitä levitettiin lähinnä yliopistoihin. Thompsonin ja Ritchien viimeinen taidonnäyte UNIXin osalta oli version 7 julkaisu vuonna 1979 [Tanenbaum, 1992]. Tämä versio on pohjana suurimmalle osalle nykyisistä UNIX-varianteista ja -versioista [Alho ja Puhakka, 1991]

Koska UNIX:n pohja oli siirrettävä, se siirrettiin VAX-koneisiin, Berkeleyn yliopiston toimesta. [Alho ja Puhakka, 1991]. Tämä oli mahdollista, koska UNIXin lähdekoodi oli kaikkien saatavilla. Ensimmäinen BSD eli 1BSD (Berkley System Distribution) ei kuitenkaan heti toiminut VAX-koneessa, vaan se tyytyi esikuvansa mukaan pyörimään vain PDP-11/20 laitteissa. [Tanenbaum, 1992]. Alhon ja Puhakan [1991] mukaan yksi oudoista myyteistä on, mihin katosi 2BSD. Sitä ei koskaan julkaistu, vaan siirryttiin suoraan 3BSD:n. Parannettu versio 4BSD pyöri sitten vihdoin VAX-koneissa. Samaan aikaan AT&T oli tehnyt oman version UNIX:sta, nimeltä 32V, joka myös oli tarkoitettu VAX-koneisiin. 32V oli vain VAX:lle siirretty versio UNIX Versio 7:stä. [Tanenbaum, 1992].

BSD:ssä oli alusta asti monia uniikkeja, tähän päivään asti yltäneitä uusia ominaisuuksia, esimerkiksi idea virtuaalimuistista ja TCP/IP [Tanenbaum, 1992]. Tanenbaumin mukaan BSD esitteli myös monia nykyisin itsestään selviä työkaluja, kuten vi (visual editor). Tämä puolestaan johti siihen, että myöhemmin monet kaupalliset toimijat, kuten DEC ja SUN, pohjasivat mieluummin BSD:n julkaisuun kuin viralliseen AT&T:n UNIX System V:n [Tanenbaum, 1992]

2.6 Mikä ihmeen Systemi V.. voittoko? (VN)

Alho ja Puhakka [1991] kertovat, että Bell Labs:n emoyhtiö, AT&T, julkaisi 1984 seuraajan System III:lle, joka oli System V. Kuten ei ole tietoa BSD:n versiosta 2, ei myöskään tiedetä, mihin System:n neljäs versio katosi. System V:tä voidaan kuitenkin pitää niin sanottuna oikeana UNIXina, josta otetaan mallia, kun katsotaan, minkä kanssa pitäisi olla yhteensopiva [Tanenbaum, 1992].

Tanenbaumin [1992] mukaan kaiken kehityksen kohtaloksi oli 1980-luvun loppupuolella koitua se, että vaikka BSD ja UNIX tulivat samasta paikasta, eivät ne voineet ajaa samoja ohjelmia. Ratkaisuksi tähän kehitettiin POSIX-standardi, joka kuvasi, miten kaiken tulisi toimia. Sen laati IEEE standardointiryhmä ja se kelpasi niin BSD:n kuin AT&T:n leirillekin. Kuitenkin vielä tämän laihan sovunkin jälkeen, UNIX ja BSD leiri jatkoivat torailuaan, mikä johti lopulta siihen, että UNIX oli hajoamispiisteessä. [Tanen-

baum, 1992]

2.7 Microsoftin aika nousee (TP)

Wikipedian [Wikipedia, 2005a] mukaan Microsoft julkaisi PC-DOS-käyttöjärjestelmän, jota se on tehnyt IBM:lle omissa nimissään MS-DOS:na vuonna 1981. Tämän jälkeen Microsoft julkaisi suhteellisen nopeaan tahtiin uuden graafisen käyttöliittymänsä vuonna 1985, joka oli melkein suora kopio Applen vastaavasta. Käyttöliittymän nimi oli Windows 1.0. [Microsoft, 2005]. Microsoftin [2005] mukaan Windows 1.0 ei ole käyttöjärjestelmä, koska se pyörii MS-DOS:n päällä. Seuraajasta, vuonna 1993 julkaistusta Windows versiosta 3.11, tuli hitti. Sen seuraaja, vuonna 2001 julkaistu Windows XP, siivittää tälläkin hetkellä x86 koneiden voittokulkua maailmalla. [Microsoft, 2005]

Mielestämme Microsoftin julkaisut osuivat oikeaan aikaan, koska UNIX-kenttä oli hajallaan ja Microsoft pystyi tarjoamaan oman, alaspäin yhteensopivan käyttöjärjestelmänsä, jonka kaiken lisäksi sai työpöydälle. Etuna oli myös se, että saman käyttöjärjestelmän sai myös palvelinkoneisiin. Tietenkään ei pidä sivuuttaa UNIX-palvelimien hintaan, joka oli 60 000 – 80 000 markkaa, nykyrahassa siis 10 000 – 13 000 euroa [Nikkanen, 2000]. Jos MS-DOS:a vertaillaan aikanaan toiseen suosittuun käyttöjärjestelmään CP/M:ään, huomataan siinäkin merkittävä ero. CP/M nimittäin pyöri Z80-prosessorilla, ja se oli jo aikanaan vanhentunutta tekniikkaa. Z80:aan verrattuna x86-suoritin oli hyvin nopea ja kustannustehokas [Wikipedia, 2005j]. Kaiken kaikkiaan MS-DOS oli jotain ihan muuta kuin UNIX-valmistajat pystyivät tarjoamaan. Näin ollen Windows saikin nopeasti jalansijaa ja naureskelun korvasi kyräily.

3. Linux - modulaarinen monoliittinen ydin (VN & TP)

Alkuperäinen UNIX on täysin monoliittinen ytimensä osalta [Tanenbaum, 1992]. Tämähän tarkoitti käytännössä sitä, että mikäli jotain ominaisuuksia halutaan lisätä ytimeen, on se käännettävä lähdekoodeistaan alusta alkaen uudestaan C-kääntäjällä [Wikipedia, 2005]. Linux toi tähän ajatukseen muutoksen. Ydin voitiin kääntää monoliittiseksi, eli yhdeksi ajettavaksi ohjelmaksi, mutta siihen voitiin ajon aikana ladata niin kutsuttuja moduuleja. Nämä moduulit pystyvät lisäämään isojakin kokonaisuuksia ytimeen [Salzman, 1991]. Tämä oli aikanaan mielenkiintoinen uudistus ja myöhemmin tullaan huomaamaan yllättäviä samankaltaisuuksia mikroytimen ja modulaarisen monoliittisen ytimen välillä. Ilkikurinen ajattelija voisi jopa sanoa niiden lähenevän toisiaan sekä ajatus- että toteutustasolla.

3.1 Linux lyhyt historia (TP)

On olemassa kaksi selitystä siitä, miksi Linus Torvalds aluksi halusi UNIXin kaltaisen käyttöjärjestelmän. Toisen mukaan hän piti UNIXin perusideasta, mutta vihasi Helsingin yliopiston VAX-VMS-koneita [Nikkanen, 2000]. Toinen taas oli, että hänen sydämensä sykähti, kun hän käytti Digitalin Ultrixia [Moody, 2000]. Yhtä kaikki välittämättä syystä, hän kuitenkin halusi itselleen UNIX-tyylisen käyttöjärjestelmän. Oikeaa UNIX-järjestelmää Torvalds ei kyennyt ostamaan, sillä sellainen maksoi 60 000 – 80 000 markkaa [Nikkanen, 2000].

Moodyn [2000] mukaan Linuxin syntymiseen eniten vaikuttava tekijä oli, ettei yksikään UNIX pyörinyt Linuksen koneessa. Vaihtoehtona oli käyttää Andrew Tanenbaumin Minix-käyttöjärjestelmää. Tanenbaum [1992] luonnehtii Minixiä opetuskäyttöön kirjoitetuksi UNIX-klooniksi 386-koneille. Tämä ei kuitenkaan tyydyttänyt Torvaldsia. [Nikkanen, 2000]. Vaikka Minixistä saikin lähdekoodin, sitä ei saanut muuttaa, eikä salaa tehtyjä muutoksia voinut jakaa eteenpäin. [Moody, 2000].

Alun perin terminaaliemulaattorina ja postinlukuohjelmanan alkanut projekti sai yllättäen ytimen piirteitä. Tämän pohjalta Torvalds kirjoittikin lopulta Linux-ytimen version 0.01 [Nikkanen, 2000]. Moody [2001] tosin väittää, että Torvalds ei ollut koskaan ajatellut, että kukaan muu voisi kiinnostua projektista. Tämä taas johti siihen, että hän kirjoitti sitä vanhan hakkerien tunnuslausahduksen mukaan, 'Just for fun', eli 'ihan vaan huvikseen'. [Moody, 2000]. Nikkasen [2000] mukaan asiat alkoivat todenteolla rullata eteenpäin vasta, kun Torvalds pyysi itselleen tilaa Helsingin yliopiston FTP-palvelimelta.

Tässä vaiheessa Torvalds sitten kirjoitti kuuluisan viestin Minix-keskusteluryhmään. Tämä tapahtui vuonna 1991 elokuussa. Pääperiaatteissaan viesti sisälsi vain ilmoituksen, että hän on kirjoittamassa jotain samanlaista kuin Minix. Se ei ollut vielä valmis, mutta kunhan jotain kokeiltavaa on, hän lähettäisi lisää informaatiota. [Torvalds,1991] Moodyn [2001] mukaan tämän jälkeen Linuxin kehitys suorastaan räjähti käyntiin. Tässä huumassa vuonna 1991 julkaistiin ensimmäinen versio, joka oli 0.01 [Nikkanen, 2000]

3.2 Koodaamista aamusta iltaan (TP)

Lumipallo oli lähtenyt vyörymään, koska Torvalds oli ilmoittanut, että hän ottaa vastaan korjauksia [Nikkanen, 2000]. Tässäkin vaiheessa Torvalds [Torvalds, 1991a] oli sitä mieltä, ettei Linuxista tule yhtä ammattimaista kuin esimerkiksi HURD-ytimeistä. HURD on GNU-projektin ydin, joka viimeistelee GNU-käyttöjärjestelmän [FSF, 2005]. Viimeinen puutuva osa tästä palapelistä 1990-luvun alussa oli HURD eli ydin [Moody, 2000]. Moody lisää vielä, että HURD oli se, joka olisi toteuttanut Richard Stallmanin

unelman: alusta loppuun vapaasta käyttöjärjestelmästä. Tämä tarkoitti, että kaikilla oli lupa ja vapaus käyttää sitä miten parhaaksi näkivät. [Moody, 2000]

Vuoden 1991 joulun aikoihin Torvalds [Torvalds, 1991b] postitti toisen ilmoituksen, jossa hän kertoi aikovansa julkaista Linux 1.0 -version seuraavan vuoden aikana. Hän sanoi, että kaikenlaista puutui vielä, ja pyysi ihmisiä auttamaan häntä. Tietenkään mikään ei mene niin kuin suunnitellaan ja version 1.0 ilmestyminen kesti hiukan suunniteltua pitempään.[Nikkanen, 2000]. Nikkasen mukaan Torvalds oli siihen mennessä osoittanut hyviä johtajan taitoja ja pitänyt projektin hallussaan, vaikka se olikin hajallaan ympäri maapalloa.

Torvalds oli kaikessa kiireessään jopa kerennyt sotia niin kutsutun fleimisodan käyttöjärjestelmätutkija Tanenbaumin kanssa. Aiheena oli ollut, miksi kukaan haluaa tähän maailmanaikaan tehdä enää uutta monoliittista ydintä. [Tanenbaum, 1992a]

3.3 Suuri tulevaisuus (TP)

Linux-ytimen kehityksen tyyliksi tuli, että pariton versionumero oli niin sanottu kehitysydin, kun taas parillinen versionumero oli niin kutsutusti vakaa ydin. Kehitysydin sisälsi paljon uusia ominaisuuksia, mutta se saattoi päivästä riippuen olla täysin käyttökelvoton tai vain osittain kelvollinen. [Nikkanen, 2000]

Ytimen versioita tuli vuodesta toiseen tasaiseen tahtiin, vaikka eräissä vaiheissa Torvalds alkoikin palaa loppuun. Tämä johtui siitä, että hän oli ottanut työn piilaaksosta, firmasta nimeltä Transmeta [Nikkanen, 2000]. Yritys oli aikoinaan kovin salamyhkäinen tuotteestaan. Nykyisin tiedämme, että siellä kehiteltiin uutta suoritinta nimeltään Crusoe. [Transmeta,2005]. Samaan aikaa Torvaldseille oli lisäksi syntynyt kaksi tyttöä. Tämän vuoksi hän joutui elämään valtavan painolastin alla, joka koostui ytimeen liittyvistä uusista ominaisuuksista ja korjauksia sisältävien sähköpostien tulvasta [Nikkanen, 2000].

Oli pakko aloittaa uusi menetelmä. Luotiin käytäntö, jossa Torvaldsilla oli niin kutsuttuja upseereja, jotka tarkastivat korjausehdotuksia ja sitten vasta kokeiltuina lähettivät sopivat ehdotukset Torvaldsille [Moody, 2000].

3.4 Kuka varasti ja keneltä (VN & TP)

SCO (Santa Cruz Operating system) syytti Linuxin kehittäjiä ja yrityksiä siitä, että ne käyttivät sen immateriaalioikeuksin suojattua koodia. [Dvorak, 2003] Tämä ei ollut SCO:n, entisen Calderan, historiassa ensimmäinen uhkayritys rahastaa hiukan heppoisilla syytöksillä. Ennen kuin Calderasta tuli SCO, oli se onnistunut rahastamaan Microsoftia MS-DOS-oikeudenkäynnissä. Silloin syytös oli ollut, että ohjelma tulostaa virheilmoituksen, mikäli sen huomattiin pyörivän DR-DOS:n alaisuudessa. DR-DOS oli

päätynyt Calderalle Novellita, samaiselta yritykseltä, jolta se lopulta osti UNIX-oikeudet ja SCO UNIX -tuotemerkin. [BBC,2000]

Linuxia vastaan käyty hyökkäystaktiikka oli tietenkin erilainen, vedottiin immateriaalioikeuksiin, joka tarkoitti käytännössä sitä, että Linuxin ydin sisälsi koodia, joka joko oli suoraan peräisin SCO UNIXista, tai sitten sinne oli kopioitu patentoitua materiaalia. Ongelmana oli, että kun suuret yrityksen aikoinaan lisensoivat UNIXin koodin itselleen, saivat ne vapaan oikeuden käyttää lähdekoodia. [Tanenbaum, 1992].

Nyt nämä jätit olivat alkaneet olla Linux-myönteisiä. Tässä nykyinen SCO näki mahdollisuuden rahastukselle. SCO siis väitti, että yritykset yrittivät päästä maksamasta lisenssimaksujaan. Tämä taas tapahtui luomalla Linuxista kopio, jota sitten kaikki saisivat käyttää ilman SCO:lle maksettavaa lisenssimaksua [Dvorak, 2003]. Aluksi SCO julisti, että heille tuli maksaa kaikista Linux-koneista lisenssimaksu. SCO lähettikin monille yrityksille maksukehotuksen. Tietävästi vain yksi yritys on maksanut kyseessä olevan lisenssin ja sekin on SCO:n tytäryhtiö [SCO, 2005]. Tämän jälkeen SCO vielä haastoi sattumanvaraisesti muutamia isoja Linuxia käyttäviä yrityksiä oikeuteen. Nämä oikeudenkäynnit olivat melkoinen farssi ja syytteet oli kyhätty nopeasti kokoon. Oikeudenkäynnit kuivuivatkin kokoon melko nopeasti, mutta tämä seikka nosti esiin tärkeän keskustelun tekijänoikeuksista. [Dvorak, 2003].

Vaikuttaisi siltä, että Linux on nyt pisteessä, jossa sitä ei enää voi pysäyttää. Tämä ei ole tapahtunut hujauksessa, vaan monen vuoden aikana. Valtavat kasvukivut ovat leimanneet Linux-yhteisöä viime vuodet ja epäilemättä ne tulevat jatkumaan. Uusin 2.6 versio ytimestä alkaa lunastaa lupauksia, joita Linuxille on asetettu vuosien varrella. Luultavasti kaikki jatkuu kuten ennenkin, uusi versio ytimestä tulee taas ensi kuussa.

4. Darwinin evoluutio (VN)

Darwinin esi-isänä voidaan pitää UNIXia, mutta ensimmäinen varsinainen Steve Jobsin UNIX tyylinen ydin oli NeXT, vuodelta 1988 [Apple, 2005]. Tässä vaiheessa Jobs ei enää ollut Applen palveluksessa. Jobs oli vaihtanut työpaikkaa NeXT Computer Inc:iin, joka valmisti NeXTStep nimistä graafista työasemaa [Wikipedia, 2005c]. Sittemmin Apple osti NeXT Computerin ja Steve Jobs palasi juurilleen tuoden mukanaan suurimman osan NeXT:in työntekijöistä [Wikipedia, 2005d]. Paluunsa jälkeen Jobs aloitti projektin, koodinimeltään Rhapsody. Rhapsodyn piti alun perin olla vain Mac OS X-palvelin, mutta siitä lopulta muokattiin myös Darwin-ydin [Wikipedia, 2005e].

Darwinia voidaan pitää mikroytimenä, mutta siinä on myös piirteitä monoliittisesta ytimestä. Mikroytimen peruseriaate on se, että itse ydin on minimaalinen ohjelma, jossa on vain välttämätön tietokoneen käynnistymistä varten. Toisin kuin monoliittisessä ytimessä, joka käynnistyessään sisältää kaikki, mitä ajon aikana tullaan

tarvitsemaan, mikroydin käynnistää palvelinohjelmia (daemon) ytimen ulkopuolelle niin kutsuttuun käyttäjän-avaruuteen. [Apple, 2005]

4.1 Seison jättiläisten harteilla! (VN)

Darwin-ydin pohjautuu 4.4 BSD-lite2:een, joka myös on haarautunut useammaksi BSD-variantiksi [Apple, 2005]. (Katso myös kuva 1) Kuitenkaan Darwin ei ole puhdasoppinen mikroydin, kuten Minix [Tanenbaum, 1992]. Osa Darwinin ominaisuuksista on peräisin Mach 3.0:sta. Itse asiassa, Darwin on Mach 3.0, jonka päälle on liimattu BSD-ydin. BSD-ydin hoitaa syötön ja tulostuksen, POSIX-standardin mukaisen API:n (Application Programming Interface) ja käyttäjän-avaruuden hallinnan. [Apple, 2005] BSD-kerroksen alla on Mach 3.0, joka puolestaan on täysiverinen mikroydin. Mach 3.0 tarjoaa Darwinin BSD-kerrokselle resurssien jakamisen, muistin hallinnan ja prosessorin kanssa työskentelyn. Kuten asiaan kuuluu, ydin keskustelee BSD-tason kanssa, niin kutsuttujen piippujen (socket) kautta. [Apple, 2005]

4.2 The piper at gates of dawn (TP)

Mikä sitten erottaa mikroytimen monoliittisesta ytimestä? Onko se vain *koko* kysymys? Lyhyt vastaus lienee: niiden ero kommunikoida käyttäjän-avaruuden kanssa. Mikroytimen pysyessä lähinnä pelituomarina, monoliittinen ydin hoitaa kaiken, mitä koneessa tapahtuu. [Scott, 1997]

Kun monoliittinen ydin voi tehdä suoran järjestelmäkutsun, joutuu mikroydin avaamaan piipun, jonka läpi kommunikoidaan määritellyn protokollan avulla [Scott, 1997]. Omana mielipiteenämme voimme esittää, että edellä kuvatun kaltainen piippurakenne tuo mukaan tarpeetonta monimutkaisuutta, joka Darwin ytimessä on melko hienosti kierretty juuri tällä kahden ytimen yhtäaikaisella toimimisella.

4.3 Mistä aurinko nousee, mikäli se nousee? (VN)

Darwin on kehittymässä voimakkaasti suuntaan, joka mahdollistaa sen ajamisen muissakin ympäristöissä kuin pelkästään PowerPC-suorittimilla. Tämä taas tarkoittaa, että Apple on astumassa Microsoftin kanssa samoille markkinoille.[Apple, 2005] Tämä ei kuitenkaan ole välttämättä Darwinin mielenkiintoisin kehityssuunta, sillä vuonna 1999 Apple julkaisi Darwinista avoimen lähdekoodin, josta syntyi OpenDarwin [Apple, 2005]. OpenDarwin taas ylläpitää Apple-riippumatonta versiota Darwin ytimestä ja lisää siihen omia tarpeelliseksi näkemiään ominaisuuksia.

5. Missä olemme nyt? (TP)

Mielestämme tämän hetkinen tietotekninen tilanne on mielenkiintoisempi kuin koskaan ennen. Kautta aikojen kysymys on kuulunut, mistä saisimme lisää laskentatehoa? Nyt kysymys on taipunut muotoon, mistä saamme tarpeeksi tuotantotehoa tuottamaan tämän laskentatehon?

Aina tietokoneiden aamunkoista lähtien ongelmana on ollut niiden valmistus. Koneet olivat isoja, kolhoja ja menivät helposti rikki. [Tanenbaum, 1992]. Tanenbaum jatkaa, että siksi näiden koneiden käyttöä tuli helpottaa ja käyttöastetta tehostaa niiden korkean hinnan vuoksi.

Kun ensimmäiset nykyisen kaltaiset, piin ominaisuuksiin nojaavat, prosessorit julkaistiin, monet ajattelivat, että tässä on kaikki mihin olemme pyrkineet. Nyt meillä on aivan hillitön laskentateho verrattuna mekaanisiin laskukoneisiin tai putkilaitteisiin, jopa kymmenen vuotta sitten olleet huippukoneet ovat kuin kivikaudelta. [About, 2005]. Mitä muuta voimmekaan enää vaatia? Tietenkin parempia, nopeampia ja ennen kaikkea hienompia tekniikan saavutuksia; nälkä kasvaa syödessä. Itse asiassa Mooren lain mukaan prosessorien laskentateho kasvaa kaksinkertaiseksi noin puolentoista vuoden välein. [About, 2005a] Uudet ongelmat alkavatkin yllättäen vallata alaa. Tähän asti piisirua on piiskattu eteenpäin lisäämällä energiaa, joka siihen syötetään. [About, 2005] Tämä energia on sitten siirtynyt lämpönä pois prosessorin päältä. Siitä se on tavalla tai toisella sitten jäähdytetty pois.

5.1 Termodynamiikan toinen sääntö (VN)

Termodynamiikan toisen lain mukaan lämpö johtuu aina kylmempään esineeseen. Tarkemmin sanoen, suljettu järjestelmä pyrkii kohti lämmön tasaantumista niin, että kaikkialla järjestelmässä vallitsee sama lämpötila. Sääntö on taannut sen, että tähän asti prosessorit on saatu jäähdytettyä lämpötilaltaan noin 40-80C asteeseen.

Teknisesti prosessorit nojaava vieläkin 1970-luvun innovaatioihin ja keksintöihin [About, 2005]. Nykyinen trendi, jossa nopeutta kasvatetaan lisäämällä transistorien määrää sentillä johtaa siihen, että auringon pinnan kuumuus saavutetaan ensi vuosikymmenen alussa. [Itviikko, 2004]. Auringon pintahan ei tietenkään ole kuin noin 5526.85 Celsiuksen lämpöinen. [Arnet, 2004]. Ei tarvitse paljonkaan ajatella, huomatakseen tässä melkoisen ongelman.

5.2 Mitenkäs nyt suu pannaan (VN)

Mikään ei estä sitä, että voisimme jäädä tähänkin. Meillä on vihdoin laskentatehoa, Internet ja kaikkia mukavia hyötyohjelmia. Mutta näin ei vain valitettavasti voida tehdä. Tietokoneet olivat tarkoitettu alun perin pelkästään laskemiseen. [Tanenbaum, 1992].

Tämän vuoksi niiden vahvoja puolia eivät ole se, mitä niiltä oikeasti odotetaan. Niiden sanavarastoon eivät kuulu seuraavan kaltaiset fraasit: ”Tapahtuma vaikuttaa todella hyvältä ja pysyy mukavasti annettujen parametrien sisällä, vaikkakin lopussa oli hiukan häikkää, tunnistan sen asiaan kuuluvaksi” tai ”Hups, jotain meni vikaan analysoin tässä saamaani kuvavirtaa ja siinä on selvästi vihreä kissa vaikka siinä pitäisi olla ruskea koira”. [SiteTerrific, 2000].

Binaarilogiikalla on vaikeaa toteuttaa suhteellisia asioita, sellaisia joissa koneen on tehtävä päätös syötteen perusteella. Syötteen, josta sillä ei ole kuin pieni ”aavistus” siitä, mitä tuleman pitää. Esimerkiksi voidaan sanoa että, tietokone saadaan kyllä ymmärtämään puhetta; siinä ei ole ongelmaa. Mutta ongelmia syntyy heti, kun puhumaan tulee toinen ihminen tai puhujalla on kurkku kipeänä. Tietenkään kukaan ei halua Linnunradan käsikirja liftareille -tyylisiä Siriuksen kyberneettisen kauppakomppanian kauppaimia AIP-koneita (Aito Inhimillinen Persoonallisuus). Nämä koneet ovat omassa luokassaan niin ärsyttäviä, ettei niitä myytäisi, elleivät ne olisi ainuita vaihtoehtoja, joita saatavilla on. [Adams, 2000].

6. Klusteritekniikka ja mahdottoman mallintaminen (TP)

Kuten edellä todettiin tietokoneet alun alkaen tarkoitettiin yksin omaan laskemaan kaavoja tai ongelmia, jotka veivät ihmisiltä tarpeettomasti aikaa [Tanenbaum, 1992]. Olemme todenneet, että nykyiset normaalit pöytäkoneet vastaavat teholtaan entisiä niin kutsuttuja superkoneita. Tietenkin vaatimukset ovat kasvaneet tehon kasvun myötä. Ennen riitti, että tilitiedot sai helposti laskettua. Nykyään tarvitaan tieto siitä, mitä tapahtuu, kun ytimet halkeavat ja ydinlataus laukeaa. [Tietoviikko, 2001]

Edellä puhuttiin myös siitä, ettei nykyisen kaltaisella tekniikalla ole mahdollista päästä enää paljon pidemmälle. Tietenkin varaa on vielä, mutta tuskin mikään ihmisen rakentama kestää kokoaikaista auringonpinnan lämpötilaa. Onkin katsottava pidemmälle. On yritettävä toisaalta elää sillä mitä on, mutta saatava siitä kuitenkin maksimaalinen hyöty. Tässä vaiheessa astuvat kuvaan niin kutsut klusterit. Klusteri tarkoittaa pähkinäkuoressa jotain tapaa, liittää useita koneita toisiinsa siten, että niitä voidaan käyttää, kuin ne olisivat yhtä konetta. [Techtarget, 2005]

Tekniikkaa seuraavan ja siitä artikkeleita kirjoittavan Techtargetin [2005] sivuston määritelmän mukaan klustereita voidaan käyttää laskennallisen kuorman jakamiseen, tai laskennan osien suorittamiseen, eri koneissa yhtä aikaa. Mikäli laskennallinen ongelma voidaan jaotella pienempiin palasiin. [Beowulf, 2005]

6.1 Beowulf – Mytologinen skandinaavi hirviön tappaja? (TP)

Tekniikoita klusteroinnin toteuttamiseen on monia. Tämän tutkimuksen tiimoilta on tietenkin parempi, että tutkimme Linux -tekniikkaa.

Beowulfin historia ulottuu vuoteen 1993 [Beowulf, 2005]. Beowulf on perinteinen klusterointiratkaisu siinä mielessä, että siinä tarvitaan yksittäisiä koneita, jotka sitten toimivat kuin ne olisivat yksi ainoa kone. [Techtarget, 2005] Voidaanko siis tällä tekniikalla toteuttaa Orwellin klassikko, scifi-fantasian mukainen, kaiken tarkkailujärjestelmä? [Orwell, 1999]

Orwellhan [1999] kuvaa yhteiskuntautopiassaan, miten kaikki laitteet ovat liitetty yhteen ja ihmisiä tarkkaillaan jatkuvasti, kaiken muun lisäksi ainakin televisioruutujen välityksellä, niin kutsutun Isoveljen toimesta. Tämä paranoialta haiskahtava tarinahan voitaisiin periaatteessa toteuttaa, mikäli voitaisiin jollain tavalla saada riittävästi prosessointitehoa, jolla sitten tarkkailtaisiin kaikkea, mitä ihmiset tekevät ja jopa ajattelevat.

6.2 Echelon, tarkkailua aamusta iltaan (VN)

Echelon on projekti, joka seuloa muun muassa informaatiota verkosta [Wikipedia, 2005h]. Asiaa on tutkinut myös Euroopan Parlamentin asettama komitea, jonka loppuyhteenveto on hiukan moniselitteinen [Europarlamentti, 2001]. Euroopan Parlamentin asettama komitea ei kuitenkaan kiellä Echelonin olemassaolon mahdollisuutta ja esittääkin sitä vastaan suojautumista.

Echelonin perusidea on seuloa asiasanalistalla verkossa liikkuvan informaation joukosta mielenkiintoista materiaalia Yhdysvaltojen ja sen liittolaisten käyttöön [Europarlamentti, 2001]. Järjestelmää voidaan käyttää teollisuusvakoiluun, toisen valtion kimppuun käymiseen tai toisen valtion kommunikation vaikeuttamiseen. [Wikipedia, 2005h]

6.3 Miten klusterointi liittyy ytimiin? (TP)

Klusterointi on ytimen kannalta hyvin haastava tehtävä. [OpenMosix, 2005]. OpenMosix on toinen Linux-ytimen ympärille keskittyvä klusterointiratkaisu. Hienoa klustereissa ei ole se, että monet erilliset tietokoneet tekevät yhtä aikaa samaa asiaa, vaan se, että käyttäjälle tämä on täysin läpinäkyvää. Käytännössähän tämä merkitsee sitä, että esimerkiksi sovelluksen käyttäjä kuvittelee parhaassa tapauksessa ajavansa sovellusta yhdellä ainoalla koneella.[OpenMosix, 2005].

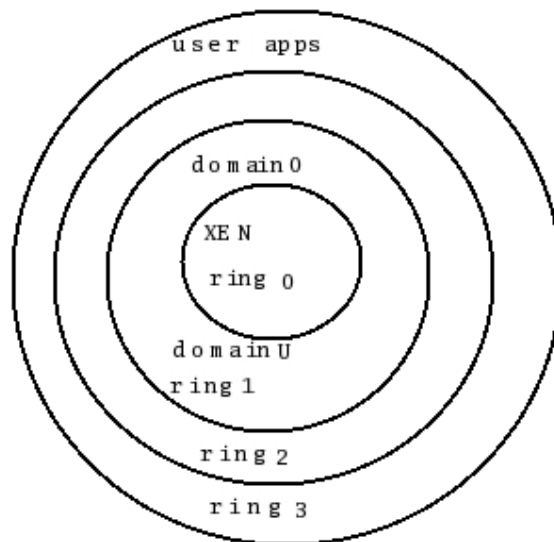
Tämä tuo tietenkin ytimelle aivan uusia ongelmia verrattuna siihen, että ydin pyörisi vain yhdessä koneessa. Se, miten tietovuota pitkin kulkeva tieto klusteroidaan usealla koneella samanaikaisesti, on jo itsessään ongelmallista [IBM, 2005]. Ongelma on tietenkin se, että tieto voi olla tulossa miltä tahansa koneelta, millä hetkellä tahansa. [IBM,

2005] Esimerkiksi suosittu hakukone Google on valtava klusteri. [TNL, 2005] Se on nykyisellä mittapuulla huimaavan tehokas tiedon tallennus- ja seulontalaitteisto. [TNL, 2005] Yksin Googlen ylläpitämä palvelu Google groups on säilönyt yli teratavun verran viestejä, joita on postitettu uutisryhmiin [Google, 2005]. Pitkälle vietyä ajatuksena voidaan pohtia, mihin kaikkeen näitä klustereita voidaan käyttää. Yksi vaihtoehto on yrittää kehittää klusterien ja ytimien ympärille jonkinlaista keinoelämää tai tekoälyä [Silva, 1999].

7. Virtuaaliset koneet (VN)

Idea koneesta koneen sisällä lienee alun perin Alan Turingin [Wikipedia,2005i]. Nykyisin tämä idea on kuitenkin jalostunut, ja saanut sellaisia muotoja, joissa voidaan käynnistää pyörivän ytimen sisälle toinen, täysin toimiva ydin. Rosenin [2005] mukaan tekniikka on jo niin pitkällä, että koneeseen voidaan käynnistää aivan toisentyypinen ydin, kuin siinä käynnistyshetkellä pyöri. Tämä tarkoittaa sitä, että toinen käynnistyvä ydin luulee olevansa ainoa pyörivä ydin koko koneessa.

Rosen [2005] käsittelee Linuxin erästä virtuaalisointikonetta XEN:ä. Kuva 2 osoittaa, miten XEN jakautuu ytimessä.



Kuva 2. Xen-virtuaalisoinnin rengasmainen rakenne [Rosen, 2005a]

Rosen [2005] mukaan sisimpään kehään käynnistyy varsinainen virtuaalinen kone. Tämän jälkeen seuraa kehä 1, joka itsessään on jo eräänlainen virtuaalinen kone. Tälle kehälle käynnistetään muut ytimet. Käyttäjän ohjelmat pyörivät kehällä kolme, joka on käyttäjänavaruus [Rosen, 2005]. Rosen mielestä tällainen tekniikka mahdollistaa klusterin ajamisen yhdessä koneessa siten, että ajettavat ohjelma ovat turvassa toisiltaan.

Hiukan samanlaista järjestelmää tarjoaa IBM uudella Cell-prosessorillaan. Ideana

siinä on se, että yhdessä prosessorissa voi olla monia prosessoreja. [PCStats, 2005] Tämä tarkoittaa sitä, että yhdessä prosessorissa voi pyöriä monta samanaikaista prosessoria, eli se on oikeasti moniajava prosessori [PCStats, 2005]

Tässä vaiheessa voimme pysähtyä pohtimaan kysymystä, mitä tapahtuu kun yhdistämme virtuaalikoneiden voiman Cell-prosessorin kykyyn todellisen moniajon kanssa? Ikuinen agenttien takaisin kytkentä?

8. Tekoäly – kuka sitä tarvitsee? (VN & TP)

Herää tietenkin kysymys, tarvitaanko älykkäitä koneita ja kuinka älykkäitä. Douglas Adams [2000] on kirjassaan nerokkaasti ilmentänyt ongelman masentuneella robotilla, Marvinilla. Marvin sanailee Maailmanlopun-ravintolan autotallissa, mistä hän soittaa sinne saapuneille ”ystävillään”, tähän tapaan:

”Työnnä vastakkaiseen suuntaan, Marvin, Niin minulle sanotaan, avaa ilmalukko numero kolme, Marvin. Marvin, nosta paperi, ajattele paperi, vaikka minulla on kokonaisen planeetankokoiset aivot”. [Adams, 2000]

Mitä tapahtuukaan, kun koneet ovat älykkäämpiä kuin luojaansa? Voiko näin edes tapahtua? Masentuuko kone siitä, että sillä teetetään sen mielestä täysin ala-arvoisia töitä?

Alan Turing loi kuuluisan Turingin testin. Sillä mitataan koneen inhimillisiä piirteitä [Koikkalainen, 2001]. Vuonna 1991 Hugh Loebner aloitti kilpailun, jossa vuosittain kilpailaan siitä, kuka pystyy vastaamaan Turingin haasteen [Loebner, 2005]. Loebner on tyytyväinen, koska kukaan ei vielä ole voinut toteuttaa kyseistä konetta tai ohjelmaa siten, että valvotuissa olosuhteissa olisi voinut loputtomasti huijata kysyjää [Koikkalainen, 2001]. Vaikka Turingin testiä ei, ainakaan vielä näillä ideoilla, ole kyetty toteuttamaan, on tämä älyllinen haaste antanut mielestämme paljon tietotekniikalle.

9. Yhteenveto (VN & TP)

Tietokoneet ovat tulleet pitkän matkan huoneen kokoisista jättiläisistä nykyisiin pikkuruisiin laitteisiin. Tekniikka on kehittynyt ja muuttunut aikojen saatossa. Ollaan opittu tekemään tarkempaa työtä ja hienompia komponentteja. Silti kaikki innovaatiot nojaavat muutama vanhaan oivallukseen ja niiden tarjoamiin mahdollisuuksiin.

Ensimmäinen oivallus oli tietenkin tyhjiöputkien keksiminen. Siitä taas seurasi mahdollisuus siirtyä vähitellen pois mekaanisista koneista, kohti täysin digitaalisia koneita. Kuten edellä kuvattiin, yksi merkittävä idea muutti lähes kokonaan sen, miten ymmärrämme koneet ja mitä niiltä nykypäivänä vaadimme. Tämä idea oli UNIX, jolle nykyinen kehitys on paljon velkaa. Näinkin merkittävän innovaation syntymisen mahdollisti muutaman ihmisen intohimo ja tutkimisen halu.

Ei voida silti sanoa, että UNIX:kaan olisi muuttanut mitään tietotekniikan perusteista. Ne pohjaavat edelleen Boolean logiikkaan ja Turingin määritelmiin. Voidaan sanoa, että nykyykoneet ovat mestariteoksia; eräänlaisia Turingin koneita, olematta kuitenkaan niin mestarillisia ja ajattomia, kuin itse määritelmänsä. UNIX on poikunut joukon ”lapsia”, joista osa on hiipunut historian unholaan ja toiset taas ovat vielä olemassa.

Käsitlemistämme ytimistä löytyy eroja ja yhtenäisyyksiä. Kaikkien perustehtävä on sama: pyörittää koneen eri osia mahdollisimman tehokkaasti ja luotettavasti. On hyvin hankalaa sanoa, mikä ydin hoitaa tehtävänsä parhaiten, ehkä se onkin vain makuasia? Se kuitenkin voidaan todeta, että UNIX ei ole enää vuosiin kilpaillut Linuxin ja Darwinin kanssa. Toisaalta ilman UNIXia Linux ja Darwin eivät kilpailisi keskenään.

Mikäli laskentatehoa halutaan jatkuvasti kasvattaa, tullaan pian nykyisten prosessoreiden ääri rajoille. Tämä tulee mitä ilmeisemmin heijastumaan myös tulevaisuuden ytimiin. Mielestämme nykyiset ytimet ovat aikansa eläneitä. Voidaan olettaa, että perinteisten yksiprosessori koneiden aikakausi on loppusuoralla, ensimmäiset moniprosessori koneet on jo esitelty. Kaukana ei ole ajatus siitä, että kun vielä nykypäivänä muisti ja nopeus kasvavat mielettömällä nopeudella, niin muutaman vuoden päästä prosessorien lukumäärä on paras myyntivaltti. Usean prosessorin koneissa tulee olemaan hankaluuksia, mikäli niissä edelleen käytetään nykyisen kaltaisia ytimiä ja klusterointia; todellinen tehon kasvu jää saamatta.

Minkälainen sitten voisi olla tulevaisuuden ydin? Ilmeistä lienee, että sen on voitava hoitaa useita prosessoreita samanaikaisesti. Riittääkö siihen yksi ydin, vai tarvitaanko useampi keskenään kommunikoiva ydin? Näiden kysymysten ratkaiseminen ei ole tämän tutkielman aihepiirissä, mutta niiden esittäminen on.

Viiteluettelo

- [About, 2005] About Inc., Microprocessors Work, <http://inventors.about.com/gi/dynamic/offsite.htm?site=http://www.intel.com/education/mpworks/>, [10.12.2005]
- [About, 2005a] About, Inc., Moores Law, <http://inventors.about.com/gi/dynamic/offsite.htm?site=http://research.microsoft.com/%257Egray/Moore%5FLaw.html>, [10.12.2005]
- [Adams, 2000] Douglas, Adams, *Linnunradan käsikirja liftareille*, Juva, 2000
- [Alho ja Puhakka, 1991] Kari Alho ja Jukka Puhakka, *Unix käyttöjärjestelmä, käyttäjän opas*, Helsinki, 1991
- [Apple, 2005] Apple Inc., Apple history, <http://apple2history.org/history/appy/ahc.html>, [15.12.2005]

- [Arnet, 2004] Bill Arnet, Nine planets: Sun, <http://www.nineplanets.org/sol.html>, [10.12.2005]
- [BBC,2000] BBC News, Caldera vs Microsoft - the settlement, <http://news.bbc.co.uk/1/hi/business/600488.stm>, [21.12.2005]
- [Beowulf, 2005] Beowulf, Beowulf History, <http://www.beowulf.org/overview/history.html>, [19.12.2005]
- [Dvorak, 2003] John C. Dvorak, SCO versus IBM and unix, <http://www.dvorak.org/scotimeline/>, [05.12.2005]
- [Europarlamentti, 2001] Euroopan Parlamentti, Yksityistä ja talouselämän viestintää sieppaavan maailmanlaajuisen järjestelmän (Echelon-sieppausjärjestelmän) olemassaolo (2001/2098(INI)), http://www.europarl.eu.int/omk/sipade3?SAME_LEVEL=1&LEVEL=2&NAV=X&DETAIL=&PUBREF=-//P//TEXT+REPORT+A5-2001-0264+0+DOC+XML+V0//FI, [19.12.2005]
- [FSF, 2005] Free Software Foundation, Introduction to the Hurd, <http://www.gnu.org/software/hurd/hurd.html#TOCintroduction>, [05.12.2005]
- [Google, 2005] Google inc., Google fun facts, <http://www.google.com/press/funfacts.html>, [19.12.2005]
- [IBM, 2005] IBM Corp., General Parallel File System, <http://www-03.ibm.com/servers/eserver/clusters/software/gpfs.html>, [19.12.2005]
- [Itviikko, 2004] Taloussanomien Itviikko-lehti, Prosessorien virrankulutus räjähtää, <http://www.itviikko.com/uutiset/uutinen.asp?UutisID=59542>, [10.12.2005]
- [Koikkalainen, 2001] Pasi Koikkalainen, AI2001, <http://erin.mit.jyu.fi/pako/kurssit/AI2001/11/lect1/lect1.html>, [19.12.2005]
- [Loebner, 2005] Hugh Loebner, What is the Loebner Prize? <http://www.loebner.net/Prizef/loebner-prize.html>, [19.12.2005]
- [Microsoft, 2005] Microsoft Corp., Windows timeline, <http://www.microsoft.com/windows/WinHistoryProGraphic.aspx>, [05.12.2005]
- [Moody, 2000] Glyn Moody, Kapinakoodi, , []
- [Nikkanen, 2000] Tuula Nikkanen, *Linuxin tarina: Linus Thorvalds - mies menestyksen takana*, Helsinki, 2000
- [OpenMosix, 2005] Open Mosix, OpenMosix, <http://openmosix.sourceforge.net/>, [19.12.2005]
- [Orwell, 1999] George Orwell, *Vuonna 1984*, Porvoo, 1999
- [PCStats, 2005] PCStats.com, IBM's CELL Processor: Preview to Greatness?, <http://www.pcstats.com/articleview.cfm?articleID=1727>, [19.12.2005]
- [Rosen, 2005] Rami Rosen, <http://www.linuxjournal.com/article/8540>, <http://www.linuxjournal.com/article/8540>, [19.12.2005]

- [Rosen, 2005a] Rami Rosen, Figure 1, <http://www.linuxjournal.com/articles/web/2005-09/8540/8540f1.png>, [19.12.2005]
- [Salzman, 1991] Peter Jay Salzman, The Linux Kernel Module Programming Guide, <http://www.faqs.org/docs/kernel/>, [05.12.2005]
- [SCO, 2005] SCO Corp., Linux license FAQ, <http://www.thescogroup.com/scosource/linuxlicensefaq.html>, [05.12.2005]
- [Scott, 1997] James Scott, Meet Mach, <http://www.stepwise.com/Articles/Technical/MeetMach.html>, [15.12.2005]
- [Silva, 1999] Anderson Silva, Artificial Intelligence on Linux, <http://linuxgazette.net/issue43/silva.ai.html>, [19.12.2005]
- [SiteTerrific, 2000] SiteTerrific Web Solutions, The Net's Original Fuzzy Logic Archive - Since 1994, <http://www.austinlinks.com/Fuzzy/>, [10.12.2005]
- [Tanenbaum, 1992] Andrew Tanenbaum, *Modern Operating Systems*, Englewood Cliff, 1992
- [Tanenbaum, 1992a] Andrew Tanenbaum, Linux is obsolete, http://people.fluidsignal.com/~luferbu/misc/Linus_vs_Tanenbaum.html
- [Techtarget, 2005] Techtarget, cluster computing, http://searchdatacenter.techtarget.com/sDefinition/0,290660,sid80_gci762034,00.html, [19.12.2005]
- [Tietoviikko, 2001] Tietoviikko, Mmailman nopein tietokone esiteltiin Yhdysvalloissa, http://www.tietoviikko.fi/doc.do?f_id=174378, [19.12.2005]
- [TNL, 2005] tnl.net, How many google machines, http://www.tnl.net/blog/entry/How_many_Google_machines, [05.12.2005]
- [Torvalds,1991] Linux Torvalds, What would you like to see most in minix?, <http://www.li.org/linuxhistory.php>
- [Transmeta,2005] Transmeta corp., Why Crusoe, <http://www.transmeta.com/crusoe/>, [05.12.2005]
- [Wikipedia, 2005] Wikipedia, Kernel (computer science), http://en.wikipedia.org/wiki/Monolithic_kernel, [05.12.2005]
- [Wikipedia, 2005a] Wikipedia, PC-DOS, <http://en.wikipedia.org/wiki/PC-DOS>, [05.12.2005]
- [Wikipedia, 2005c] Wikipedia, Next Computer, http://en.wikipedia.org/wiki/NeXT_Computer, [15.12.2005]
- [Wikipedia, 2005d] Wikipedia, Steve Jobs, http://en.wikipedia.org/wiki/Steve_jobs#Return_to_Apple, [15.12.2005]
- [Wikipedia, 2005e] Wikipedia, Rhapsody OS, [http://en.wikipedia.org/wiki/Rhapsody_\(OS\)](http://en.wikipedia.org/wiki/Rhapsody_(OS)), [15.12.2005]
- [Wikipedia, 2005f] Wikipedia, , <http://en.wikipedia.org/wiki/Image:Unix.png>.

[Wikipedia,2005h] Wikipedia, ECHELON, <http://fi.wikipedia.org/wiki/ECHELON>, [19.12.2005]

[Wikipedia,2005i] Wikipedia, Turingin kone, http://fi.wikipedia.org/wiki/Turingin_kone, [19.12.2005]

[Wikipedia,2005j] Wikipedia, CP/M, <http://en.wikipedia.org/wiki/CP/M>, [19.12.2005]

Multimodaalisuus käyttöliittymissä

Juuso Näsi

Tiivistelmä.

Teknologian kehitys on tehnyt mahdolliseksi yhä laajempien ja vaativampien sovellusten kehittämisen. Sovellusten käyttöliittymissä tämä on merkinnyt uusia aistikanavia käyttäjän ja sovelluksen väliseen kommunikointiin. Useisiin aistikanaviin perustuvat eli multimodaaliset käyttöliittymät tarjoavat monia mahdollisuuksia uudenlaisten sovellusten kehittämisessä, mutta asettavat myös suuria haasteita ohjelmistokehitykselle. Tässä tutkimuksessa käsitellään multimodaalisten käyttöliittymien erityispiirteitä verrattuna perinteisiin käyttöliittymiin. Tutkimus esittelee kaksi esimerkkiä multimodaalisesta sovelluksesta ja arvioi multimodaalisuudesta saatavaa lisäarvoa ohjelmistokehityksessä. Lisäksi pohditaan tulevaisuuden suurimpia ongelmia ja esteitä käyttöliittymien kehittämisessä.

Avainsanat ja -sanonnat: Multimodaalisuus, datafuusio, WIMP, VoiceXML, MVC, virtuaalitodellisuus, immersio, CAVE.

CR-luokat: D.2.11, H.5, I.3.6

1. Johdanto

Multimodaalisten käyttöliittymien kehittäminen on perinteisesti ollut haastavaa ja aikaa vievää siitä saatuun hyötyyn nähden. Uusia vuorovaikutusmuotoja on yritetty jo pitkään tuoda sovellusten käyttöliittymiin, mutta vain harvat käytännön toteutukset ovat onnistuneet. Multimodaaliset käyttöliittymät ovat yksi ohjelmistokehityksen haastavimmista osa-alueista. Uudet modaliteetit lisäävät kuitenkin mahdollisuuksia kehittää ihmisläheisempiä sovelluksia, joiden käyttäminen olisi entistä helpompaa ja luonnollisempaa.

Tämä tutkimus käsittelee multimodaalisten käyttöliittymien ominaispiirteitä ja suurimpia eroavaisuuksia perinteisiin käyttöliittymiin verrattuna. Tutkimus yrittää etsiä multimodaalisten käyttöliittymien hyviä puolia, mutta myös heikkouksia. Yksinkertaisena esimerkkinä multimodaalisista sovelluksista käsitellään multimodaalisia WWW-selaimia, jotka sisällyttävät perinteisen käyttöliittymän rinnalle ääneen perustuvan käyttöliittymän. Toisessa esimerkissä käsitellään multimodaalisten käyttöliittymien merkitystä kehitettäessä virtuaalitodellisuusmaailmoja ja niihin liittyviä sovelluksia. Tutkimus esittelee immerstiivisen CAVE-virtuaaliympäristön, joka on kehitetty luomaan käyttäjälle mahdollisimman uskottava illuusio todellisuudesta. Lopuksi arvioidaan

multimodaalisuuden merkitystä tulevaisuuden käyttöliittymissä ja pohditaan kehityksen tulevaa suuntaa.

2. Multimodaalisuus käyttöliittymissä

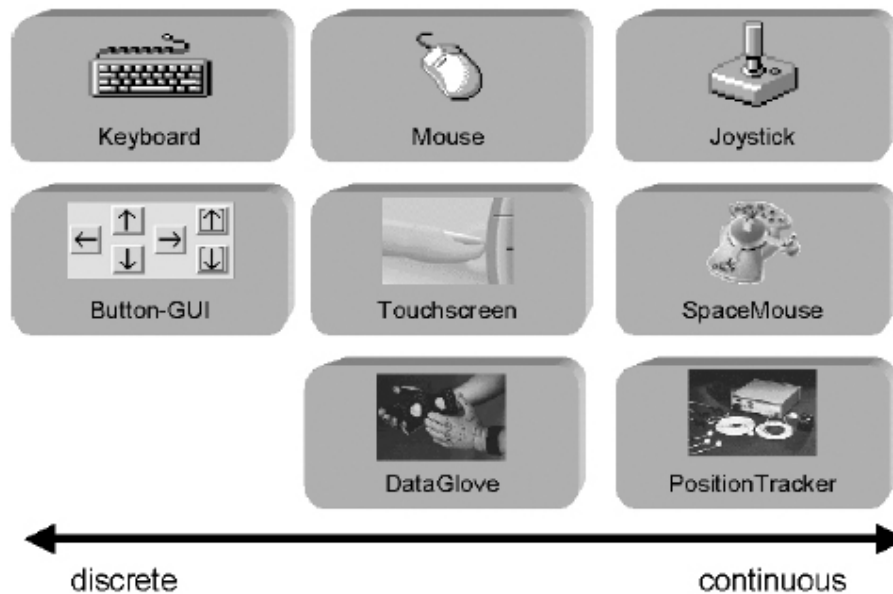
Tässä luvussa esitellään multimodaalisuus terminä ja tarkastellaan tarkemmin eri aistikanaviin perustuvia modaliteetteja. Lisäksi arvioidaan multimodaalisen ja perinteisen käyttöliittymän suurimpia eroavaisuuksia. Luvun lopussa pohditaan multimodaalisuuden mukanaan tuomia mahdollisuuksia sekä haasteita käyttöliittymien ohjelmistokehitykselle.

2.1. Perinteisestä käyttöliittymästä multimodaaliseen

Perinteisellä käyttöliittymällä tarkoitetaan yleensä graafista WIMP-käyttöliittymää. Mäkitammen [2006] mukaan WIMP-käyttöliittymissä (Window, Icon, Mouse, Pointer) vuorovaikutus on yleensä tyypiltään vuorottelevaa ja perustuu yhteen syöte- ja tulostekanavaan. WIMP-käyttöliittymässä käyttäjä ohjaa sovellusta näppäimistön ja hiiren avulla sekä saa tulosteet sovellukselta monitorin näytölle. WIMP-käyttöliittymä on toimivaksi testattu ja käytännön sovelluksissa vallitseva käyttöliittymätyyppi, mutta se ei mahdollista kovin monipuolista vuorovaikutusta käyttäjän ja sovelluksen välillä. Nykyajan PC-sovelluksista suurin osa on WIMP-käyttöliittymään pohjautuvia, ja tämä tilanne luultavasti säilyy vielä pitkään tulevaisuudessakin.

Multimodaalinen käyttöliittymä puolestaan sisältää vähintään kahteen eri vuorovaikutuskanavaan liittyvää kommunikointia sovelluksen ja käyttäjän välillä [Nigay and Coutaz, 1993]. Määritelmä on silti joskus ristiriitainen, koska joidenkin tulkintojen mukaan multimodaalisessa käyttöliittymässä vuorovaikutuskanavien pitää olla samanaikaisesti käytössä ja toisissa tulkinnoissa monen vuorovaikutuskanavan pelkkä olemassaolo riittää. Multimodaaliset käyttöliittymät ovat lähes poikkeuksetta toteutukseltaan WIMP-käyttöliittymiä monimutkaisempia ja laajemman yleistymisen esteenä ovatkin usein erilaiset toteutusongelmat.

Multimodaalisessa käyttöliittymässä voi olla teoriassa rajoittamaton määrä eri tapoja vastaanottaa käyttäjän syötteitä ja antaa käyttäjälle tulosteita. Kaikki syötteet ja tulosteet perustuvat johonkin vuorovaikutuskanavaan eli modaliteettiin. Käyttöliittymien yleisimpiä modaliteetteja ovat visuaalinen eli näköaistiin perustuva, auditiivinen eli kuuloaistiin perustuva ja haptinen eli tuntoaistiin perustuva. Jokaista modaliteettia voi edustaa suuri määrä erilaisia laitteita, joilla annetaan järjestelmälle syötteitä tai saadaan järjestelmältä palautetta.



Kuva 1. Erilaisten kosketussyötekanavien vertailu [Althoff *et al.*, 2001].

Kuvassa 1 on esimerkinomaisesti kuvattu yleisimmät haptiikkaan liittyvät syötelaitteet. Syötelaitte vaikuttaa sovelluksen käytettävyyteen melko paljon ja siksi eri tyyppiin tarkoituksiin on yleensä omat laitteensa. Joskus laite voi olla jopa sovelluskohtainen kuten esimerkiksi lentosimulaattoreissa ja joissakin tarkkuutta vaativissa tietokonepeleissä.

Yleisimpien käytössä olevien modaliteettien lisäksi multimodaalisessa käyttöliittymässä voi Mäkitammen [2006] mukaan olla edustettuna vestibulaarinen (tasapaino- ja liikeaisti), olfaktorinen (hajuaisti) tai gustatorinen (maku-aisti) modaliteetti. Nämä ovat kuitenkin kiistatta marginaalisia vuorovaikutusmuotoja ja sovellusalue on varsin rajattu. Tasapaino-, haju- ja makuaistiin perustuvat käyttöliittymät tuntuvat vielä hyvin kaukaisilta normaalisovelluksissa, mutta esimerkiksi virtuaalitodellisuuksia suunniteltaessa niillä voi olla merkittävä rooli. Edellisten lisäksi Mäkitammi pitää omana modaliteettinaan aivotoimintaan perustuvaa vuorovaikusta, jossa aivokäyttöliittymä rekisteröi syötteet käyttäjän ajatuksista. Aivotoimintaa tulkitsevien käyttöliittymien tulevaisuuden mahdollisuudet ja riskit ovat muihin modaliteetteihin verrattuna omaa luokkaansa. Niiden mukana tulisi myös lukuisia eettisiä kysymyksiä, koska pitkälle kehitettynä ajatusten tulkintaa voitaisiin käyttää vääriin tarkoituksiin.

2.2. Multimodaalisuuden mahdollisuudet ja haasteet

Multimodaaliset käyttöliittymät tuovat mukanaan paljon uusia mahdollisuuksia perinteisiin WIMP-käyttöliittymiin verrattuna. Vuorovaikutuskyvyiltään rajoittuneiden erityisryhmien kuten näkövammaisten kohdalla multimodaaliset käyttöliittymät paikkaavat usein perinteisten käyttöliittymien jättämiä aukkoja. Multimodaalinen sovellus voi tavoittaa suuremman käyttäjäryhmän kuin perinteinen WIMP-sovellus. Multimodaalisten käyttöliittymien yhtenä tarkoituksena on tehdä kommunikoinnista luontevampaa, ihmisläheisempää ja helpompaa. Multimodaalisuus tekee yleensä käyttöliittymistä todellisuutta jäljitteleviä ja siten kommunikointi on useimmiten luontevampaa kuin perinteisissä WIMP-käyttöliittymissä. Uusien mahdollisuuksien hyödyt eivät kuitenkaan ole saavutettavissa täysimääräisinä ennen kuin toteutuksen monimutkaisuus vähenee ja tavalliset käyttäjät tottuvat uusiin vuorovaikutusmuotoihin jokapäiväisessä elämässä.

Multimodaalisiin käyttöliittymiin sisältyy mahdollisuuksien ohella myös haasteita, jotka useimmiten liittyvät sovelluksen toteutukseen. Monia eri syötekanavia hyödynnettäessä on kiinnitettävä huomiota erityisesti ristiriitaisten syötteiden tulkintaan. Yksinkertaisimmillaan käyttöliittymä voisi priorisoida jollakin modaaliteetilla saadut syötteet ensisijaisiksi ja hylätä muut samanaikaiset syötteet, mutta tämä veisi suurimman osan multimodaalisuuden antamasta lisäarvosta käyttöliittymille. Ratkaisuksi ongelmaan on kehitetty erilaisia laskukaavoja sekä integrointitekniikoita. Laskukaavoja ei voida kuitenkaan yleistää palvelemaan kaikkia sovelluksia vaan ne on yleensä suunniteltava tapauskohtaisesti.

01

		USE OF MODALITIES	
		Sequential	Parallel
FUSION	Combined	ALTERNATE	SYNERGISTIC
	Independent	EXCLUSIVE	CONCURRENT
		Meaning No Meaning	Meaning No Meaning
LEVELS OF ABSTRACTION			

Kuva 2. Multimodaalisten syötteiden datafuusio [Nigay and Coutaz, 1993].

Nigay ja Coutaz [1993] esittävät kuvassa 2 oman mallinsa syötteiden luokittelusta ja datafuusiosta multimodaalisessa käyttöliittymässä. Datafuusiosta tavoitteena on muodostaa sovellukselle yksi moneen syötteeseen perustuva tehtävä. Heidän luomassaan fuusiokaaviossa on neljä eri syöteryhmää, joista jokaisen käsittely tapahtuu omalla tavallaan. Kaaviossa käyttöliittymän eri modaaliteettien käyttö on jaettu peräkkäiseen ja rinnakkaiseen käyttötapaan, joista rinnakkainen on sovelluksen kannalta huomattavasti vaativampi. Fuusion toteutustapa on puolestaan jaettu sen mukaan, tulkitako syötteet yhdessä vai itsenäisinä syötteinä. Yksinkertaisimmillaan multimodaalinen käyttöliittymä voi siis tulkita peräkkäisiä modaaliteetteja ja tulkitsee syötteet toisiinsa nähden itsenäisinä, jolloin kaaviossa päädyttäisiin poissulkevaan (exclusive) ryhmään. Vaativammat multimodaaliset sovellukset kuitenkin edellyttävät yleensä joko jaksottaisten syötteiden yhdistelyä (alternate-ryhmä) tai rinnakkaisien syötteiden itsenäistä tulkintaa (concurrent-ryhmä). Kaikista vaativin tapaus on synergia-ryhmä (synergistic), jossa rinnakkaisia syötteitä yhdistellään datafuusiossa. Nigay ja Coutaz eivät kaaviossaan ota kuitenkaan kantaa siihen, miten syötteitä käytännössä yhdistellään, vaan se jää sovelluskehittäjän omien laskukaavojen tehtäväksi. Syötteiden laadun arviointi onkin yksi keskeisimpiä sovelluskohtaisia ongelmia datafuusiossa, eikä siihen ole yleispätevää ratkaisua. Toinen suuri avoin kysymys datafuusiossa on, miten sovellus osaa päätellä syötekokonaisuuden alku- ja päättymispisteen syötevirrasta, ennen kuin yksittäisistä syötteistä muodostetaan yksiselitteinen käsky sovellukselle. Tämäkin ongelma lienee kuitenkin melko sovelluskohtainen. Saarisen [2006] mukaan datafuusiossa on usein tiettyyn modaaliteettiin perustuvia syötteitä, jotka ovat muihin syötekanaviin nähden dominantteja. Lisäksi eri modaaliteetit voivat tukea toistensa sisältämää viestiä ja vähentää virheellisen tulkinnan mahdollisuutta.

Modaaliteettien fuusion tarkoitus on yhdistellä useista syötteistä yksi käsky järjestelmälle, mutta yleensä sovelluksissa tarvitaan myös vastakkaista toimenpidettä tulosteiden kohdalla. Multimodaalisten sovellusten tulosteet eivät rajoitu yhteen vuorovaikutuskanavaan vaan mahdollisuuksia on yhtä paljon kuin syötteissäkin. Tulosteiden antaminen eri palautekanavia pitkin edellyttää fission, jossa jokin tapahtuma kuvataan käyttäjälle usean modaaliteetin avulla [Saari- nen, 2006]. Fission onnistunut toteutus on perusedellytys käyttöliittymän toimivuudelle ja käytettävyydelle.

3. Käytännön sovelluksia

3.1 Tarkasteluperusteista

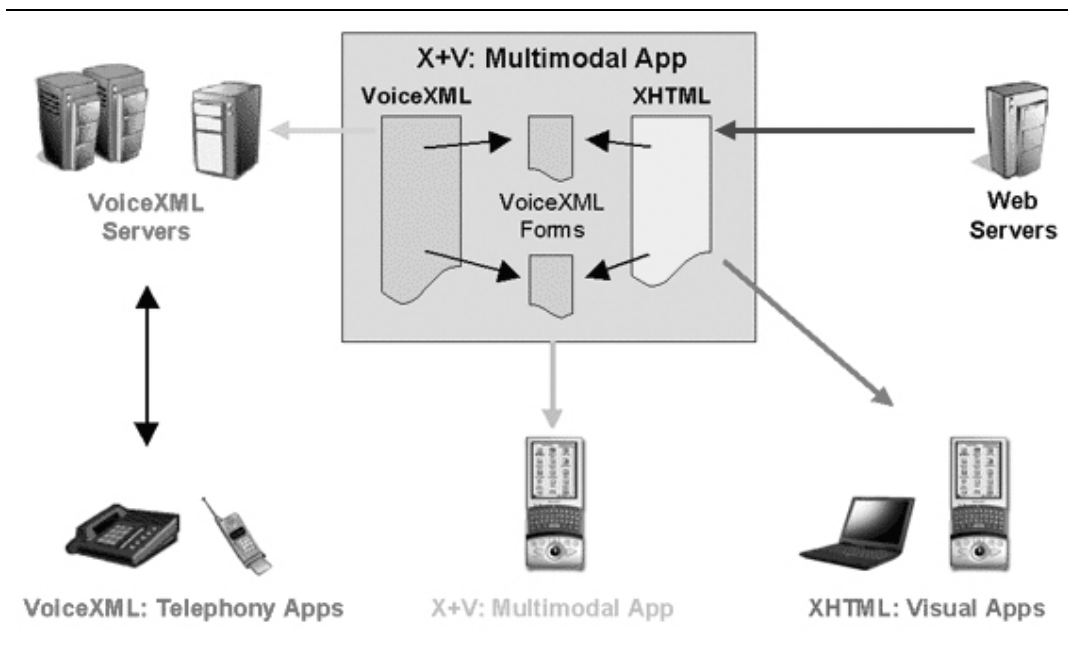
Tässä luvussa esitellään kaksi multimodaalisen käyttöliittymän sisältävää sovellusta. Molempien sovellusten kohdalla esitellään lyhyesti sovellusalueen ominaispiirteitä ja siihen liittyviä erityistarpeita. Sovelluksien käyttöliittymiä on pyritty arvioimaan monelta eri kannalta ja erityisesti on pohdittu multimodaalisuuden sovellukselle mahdollisesti antamaa lisäarvoa.

3.2 Multimodaalinen WWW-selain

IBM:n ja Operan yhteistyössä kehittämä multimodaalinen WWW-selain on yksi viimeaikaisista yrityksistä tuoda multimodaaliset käyttöliittymät jokapäiväiseen Internet-käyttöön. Selaimen kehittämisen yhtenä päätavoitteena on vähentää käyttäjän riippuvuutta perinteisestä työpöytätietokoneesta ja mahdollistaa samojen tietoverkkopalveluiden käyttö myös muissa ympäristöissä [Opera]. Multimodaalinen WWW-selain sisältää perinteisen visuaalisen käyttöliittymän lisäksi auditiivisen laajennuksen, joka mahdollistaa käyttäjän ja WWW-sivun välillä ääneen perustuvan vuorovaikutuksen. Zhangin [2004] mukaan äänikäyttöliittymällä on yleensä kaksi merkittävää erityistehtävää, joista puheen tunnistus huolehtii syötteiden analysoinnista ja puheen tekstiksi muuttava osa palautteen antamisesta käyttäjälle. Äänimodaliteetin sisältämän selaimen toteutus perustuu XHTML+Voice-teknologiaan [Le Hors et al., 2001], jossa äänivuorovaikutus integroidaan XHTML-formaattiin. XHTML on rakenteisen tiedon esittämiseen tarkoitettu merkkauskieli, jolla voidaan toteuttaa WWW-sivuja.

Multimodaalisen WWW-selaimen arkkitehtuurissa eri modaliteetit yhdistetään itsenäisinä moduuleina XHTML- ja VoiceXML-dokumenttien muodostaman dataytimen ympärille. Maes ja Ferial [2002] suosittelivat toteutukseen MVC-mallia, jossa käyttöliittymä eriytetään ydintoiminnot sisältävästä mallista. MVC (Model, View, Controller) on laajalti käytetty suunnittelumalli, jonka suurin etu on sovelluksen ydintoimintojen (Model) täydellinen riippumattomuus käyttöliittymästä (View-Controller). Tästä syystä uusia modaliteetteja voidaan periaatteessa lisätä helposti käyttöliittymään koskematta sovelluslogiikkaan [Näsi, 2006]. MVC on kuitenkin melko harvinainen suunnittelumalli multimodaalisten sovellusten kohdalla, koska sen tuki eri modaliteettien samanaikaisuudelle on varsin rajallinen. Multimodaalinen WWW-selain ei kuitenkaan edellytä täydellistä samanaikaisuutta vaan eri modaliteettien käyttö on luonteeltaan vuorottelevaa. Jos useita syöte- ja tulostevirtoja käytettäisiin samaan aikaan, jokin pidemmälle kehitetty agenttiarkkitehtuuri olisi varmasti MVC-mallia parempi suunnittelulähtökohta. MVC-malli tarjoaa kuitenkin WWW-selaimen tapauksessa helpon lähtökohdan lisätä uusia View-Controller-

pareja ydintoiminnot sisältävään malliin. Tämä mahdollistaa esimerkiksi erilaisia mobiililaitteita varten räätälöidyt käyttöliittymät tai jopa käyttäjäkohtaisen räätälöinnin. Jokaisen View-Controller-parin toteutus voi olla täysin erillinen [Näsi, 2006], mutta niiden on kommunikoitava mallinsa kanssa arkkitehtuuritasolla määritellyllä tavalla. Kuva 3 [IBM] esittää multimodaalisen sovelluksen arkkitehtuuria. Kuvassa sovelluksen eri käyttöliittymämodaliteetit kommunikoivat XHTML+VoiceXML-ydinmallin kanssa. Äänimodaliteetin toiminta vaatii lisäksi äänentunnistuksesta ja tekstin ääneksi muuntamisesta huolehtivan VoiceXML-palvelimen. Tämä palvelin toimii eräänlaisena sovittimena sovelluksen ja äänikäyttöliittymän välillä.



Kuva 3. Multimodaalisen sovelluksen arkkitehtuuri [IBM].

Operan ja IBM:n multimodaalinen WWW-selain on arkkitehtuurinsa puolesta kehittynyt ja periaatetasolla toimiva ratkaisu, mutta siihen liittyy silti joitakin merkittäviä ongelmia. Perinteiseen käyttöliittymään tehdyt laajennukset eivät ole yhtä luotettavia kuin tavallinen PC-tietokoneen näytöltä käytettävä graafinen käyttöliittymä. Perinteinen käyttöliittymä on verraten yksinkertainen ja helppo toteuttaa mobiili- ja puhekäyttöliittymiin verrattuna. Maes ja Ferial [2002] listaavat mobiilikäyttöliittymien keskeisiksi ongelmiksi laitteiden näppäimistöjen sekä näyttöjen pienen koon, mikä vaikeuttaa ja hidastaa käyttöä. Äänikäyttöliittymien huonoiksi puoliksi he mainitsevat äänentunnistuksen virhealttiuden ja äänitulosteiden esittämisen ongelmat. Tulevaisuuden laitteissa ongelmat tuskin katoavat kokonaan, vaikka kehitys

tulee luultavasti etenkin mobiililaitteiden käyttöliittymissä olemaan kovan kilpailun ansiosta erittäin nopeaa.

Maes ja Ferial [2002] luettelivat useita ongelmia kolme vuotta sitten ilmestyneessä artikkelissaan, mutta tähän päivään mennessä monet niistä on uskoakseni jo voitettu. Äänimodaliteetti on selaimissa melko uutta teknologiaa ja siten varmasti nopeasti kehittyvää. Testatessani multimodaalisia WWW-selaimia käytännössä ne osoittautuivat yllättävän helppokäyttöisiksi ja äänimodaliteetti oli varsin toimiva. Multimodaalinen selain ei kuitenkaan ole vielä joka tietokoneen vakiosovellus, vaan sen yleistyminen on vasta edessä. Etenkin mobiililaitteissa äänimodaliteettiin perustuva selain voi nopeuttaa Internetin käyttöä merkittävästi, koska mobiililaitteiden näppäimistöt ovat usein hankalia ja hitaita käyttää.

Operan selain ei suinkaan ole ainoa markkinoilla oleva multimodaalinen WWW-selain, vaikka se onkin luultavasti saanut eniten julkisuutta. Testaamistani Opera- ja Netfront-selaimista ei löytynyt suuria eroja toisiinsa nähden. Perinteiseen selaimen verrattuna ainoa ulospäin näkyvä ero on äänen nauhoituspainike, joka ottaa syötteitä vastaan käyttäjältä. Selain antaa myös selkeää englanninkielistä äänipalautetta käyttäjälle. Äänimodaliteetti on kuitenkin vain lisäominaisuus, ja multimodaalista WWW-selainta voi käyttää myös perinteiseen tapaan aivan kuten mitä tahansa muutakin WWW-selainta. Kuvassa 4 on ruudunkaappaus multimodaalisesta NetFront 3.1 selaimesta, joka on ominaisuuksiltaan suhteellisen yksinkertainen Operan vastaavaan selaimen verrattuna. Äänimodaliteetin käyttöä varten selaimen työkalupalkissa on mikrofonikonni, jolla syötteitä voi nauhoittaa.



Kuva 4. NetFront 3.1 multimodaalinen WWW-selain.

Yksi multimodaalisten selainten suurimmista hyödyistä on käyttäjän sijainnin merkityksen vähentyminen. Käytännössä samat sovellukset ovat eri vuorovaikutuskanavien avulla käytettävissä missä tahansa ja useiden eri laitteiden kautta. Uusien modaliteettien myötä tietoverkkopalveluiden saataavuus on huomattavasti entistä parempi, mutta käytettävyyso Ongelmien takia

uudet käyttöliittymämodaali- teetit vievät todennäköisesti vain pienen osan perinteisen PC-tietokoneen osuudesta palveluiden käytön välineenä. Muita multimodaalisen WWW-selaimen mukanaan tuomia hyötyjä on mm. erityisryhmien huomioiminen, koska jatkossa myös näkövammaisilla on mahdollisuus käyttää Internetiä. Multimodaalinen WWW-selain tulee siis varmasti tarpeeseen, ja pidemmälle kehitettynä sen merkitys kasvanee entisestään.

3.3 Multimodaalisuus virtuaalitodellisuudessa

3.3.1 Virtuaalitodellisuuksista

Virtuaalitodellisuudella (VR, Virtual Reality) tarkoitetaan keinotekoisia todellisuutta. Feiner ja MacIntyre [1996] esittävät omassa tulkinnassaan, että virtuaalitodellisuuden tarkoitus on jäljitellä todellista maailmaa ja luoda käyttäjälle uskottava illuusio todellisuudesta. Virtualitodellisuus on alun perin määriteltä käyttäjän täysin sisäänsä ottavaksi maailmaksi, mutta sittemmin virtuaalitodellisuudeksi on kutsuttu myös todellisuutta jäljitteleviä tietokoneohjelmia ja vastaavia erityislaitteista riippumattomia todellisuuden jäljittelyyn pyrkiviä järjestelmiä. Virtuaalitodellisuudella on useita eri muotoja, jotka Claydonin [2003] tutkimuksessa luokitellaan seuraavasti:

- Keinotodellisuus (Artificial reality) perustuu tilavaikutelman luomiseen esimerkiksi kolmiulotteisen kuvan avulla. Keinotodellisuuden toteuttaminen ei vaadi erityislaitteita.
- Lisätty todellisuus (Augmented reality) perustuu todellisen maailman esittämiseen vahvistettuna erilaisilla efekteillä.
- Etäläsnaolo (Telepresence) perustuu kamerateknologiaan, jossa käyttäjälle luodaan kuva fyysisesti toisaalla olevasta ympäristöstä ja käyttäjä voi yleensä liikkeillään vaikuttaa kameroiden suuntaan.
- Etätoiminta (Teleoperation) perustuu etäällä olevan laitteen kauko-ohjaukseen esimerkiksi videokuvan avulla.
- Upottavat järjestelmät (Immersive systems) ovat yleensä multimodaalisia ympäristöjä, joissa käyttäjä on fyysisesti järjestelmän sisällä joko osittain tai kokonaan. Upottavat järjestelmät ovat pisimmälle kehittynyt virtuaalitodellisuuden muoto.

Virtuaalitodellisuus mielletään useimmiten viihdekäyttöä varten rakennetuksi ympäristöksi, mutta nykyisin virtuaalitodellisuutta käytetään myös esimerkiksi oppimisjärjestelmissä. Opetuskäytössä virtuaalitodellisuus voi tuoda paljon uutta lisäarvoa joillakin oppimisen osa-alueilla. Bricken [1991] luettelee VR-sovelluksien eduiksi opetuskäytössä simulaatio-, ryhmätyö- ja visualisointimahdollisuudet ja painottaa ”tekemällä oppii” -periaatetta. Multimodaaliset virtuaalitodellisuusjärjestelmät ovat perinteistä opetusta mielenkiintoisem-

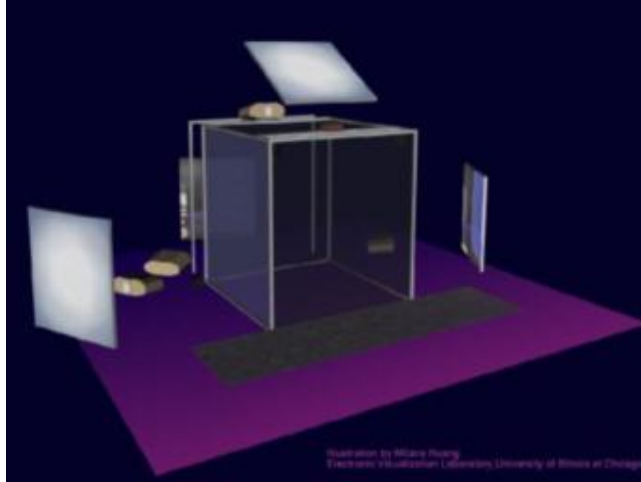
pi tapa oppia, ja erikoisissa tilantessa opitut asiat jäävät yleensä paremmin muistiin. Bricken kuitenkin huomauttaa, että itse teknologia ei paranna opetusta, vaan siihen tarvitaan toimivia ja hyvin suunniteltuja käytännön sovelluksia. Asian voisi ääritapauksissa kääntää myös ylösalaisin ja väittää, että huonosti suunnitellut VR-oppimisympäristöt voivat haitata opetusta. Esimerkiksi käytettävyysongelmat ovat yleisiä tämäntyyppisissä sovelluksissa niiden monimutkaisuuden vuoksi.

Monimutkaisia virtuaalitodellisuuksia suunniteltaessa multimodaalisuus on keskeisessä asemassa. Virtuaalimaailmojen toteutuksissa käytetään yleensä moneen aistikanavaan perustuvia tulosteita, jotta käyttäjän kokemus olisi mahdollisimman todentuntuinen. Upottavissa virtuaalitodellisuuksissa modaliteettien käyttö on yleensä kaikkein monipuolisinta. Virtuaalitodellisuuden toteuttamisessa tuki reaaliaikaisuudelle on yksi keskeisimpiä tavoitteita [Bryson *et al.*, 1994], koska muussa tapauksessa virtuaalimaailman todentuntuisuus on varsin heikko. Multimodaaliset VR-sovellukset ovat reaaliaikaisuuden kannalta ongelmallisia, koska usean modaliteetin samanaikainen käsittely vaatii järjestelmältä paljon tehoa. Tästä syystä VR-ympäristöjen tulosteiden laatu on kehittynyt suunnilleen samaa tahtia muun tietotekniikan kehityksen kanssa.

3.3.2 CAVE-virtuaaliympäristö

Upottavia VR-järjestelmiä suunniteltaessa multimodaalisuuden eri muotojen hyödyntämisessä on rajana vain mielikuvitus. Hyvä esimerkki usean modaliteetin VR-järjestelmästä on projektoriheijastuksiin perustuva CAVE, joka on täysin immersiivinen virtuaalitodellisuusympäristö. CAVE-järjestelmän on kehittänyt The Electronic Visualization Laboratory (EVL) vuonna 1991, ja sen jälkeen on toteutettu useita CAVE:n jatkoprojekteja. CAVE-järjestelmää kehitettäessä haluttiin luoda VR-järjestelmä [Cruz-Neira *et al.*, 1993], jossa vältetään silloisten muiden järjestelmien ongelmat kuten huono resoluutio ja eristyneisyys todellisesta maailmasta. Lisäksi olemassaolevien VR-järjestelmien ongelmana oli, että ne oli tarkoitettu vain yhdelle käyttäjälle kerrallaan, kun taas CAVE suunniteltiin monen samanaikaisen käyttäjän järjestelmäksi.

Kuva 5 esittää CAVE-virtuaalimaailmaa, jossa käyttäjä on fyysisesti projektorien avulla maisemoidussa huoneessa. Useimmiten CAVE-järjestelmissä on visuaalisen tulostemodaliteetin apuna käytössä myös kolmiulotteinen äänimaisema, jolla tila saadaan elävämmäksi. Lisäksi käyttäjä voi saada haptiikkaan perustuvaa palautetta järjestelmältä. CAVE-ympäristö reagoi esimerkiksi käyttäjän liikkeeseen VR-ympäristön sisällä.



Kuva 5. CAVE-virtuaalimaailma [Pape, 2001].

CAVE-virtuaalimaailma on yksi tunnetuimpia immersiiivisiä virtuaalitodellisuusjärjestelmiä, ja sen avulla pystytään luomaan erittäin todentuntuisia ympäristöjä. Multimodaalisten käyttöliittymien kannalta CAVE on ensimmäisiä suuria hankkeita, joissa käyttäjän ja järjestelmän välinen vuorovaikutus tapahtuu usealla modaliteetilla ja kuvanlaatu on suhteellisen hyvä.

4. Yhteenveto ja katsaus tulevasta

Käyttöliittymien monipuolistuminen on vielä varhaisessa vaiheessa, joten suurin osa kehityksestä on vasta edessäpäin. Teknologian puolesta on jo olemassa edellytykset rakentaa kohtuullisella vaivalla multimodaalisia käyttöliittymiä, mutta varsinaiset käytännön massasovellukset odottavat vielä läpimurtoaan. Tällä hetkellä käytössä olevat multimodaaliset käyttöliittymät liittyvät useimmiten joidenkin erityisryhmien tai tutkimuksen tarpeisiin. Suuria sovellusalueita ovat lisäksi tietokonepelit ja viihdeteollisuus, jotka ovat koko historiansa ajan yrittäneet jäljitellä todellista maailmaa aisteineen ja kommunikaatiokanavineen. Näilläkin sovellusalueilla multimodaalisuus tulee tulevaisuudessa todennäköisesti lisääntymään.

Tietojenkäsittelytieteessä tulevaa kehitystä ja muutoksia on kuitenkin erittäin vaikea ennustaa, eikä käyttöliittymien kehittyminen liene poikkeus. Tulevaisuuden käyttöliittymissä vuorovaikutusmuodot kasvavat lähes varmasti, mutta kehitys ei välttämättä koske perinteisiä sovelluksia vaan enemmänkin erityistarpeita omaavia järjestelmiä. Liialliset vuorovaikutuskanavat hankaline syötteineen voivat tehdä sovellusten toiminnasta virhealttiimpaa, joten uusien modaliteettien lisäämisen yksinkertaisiin ja toimiviin sovelluksiin ei pitäisi olla

mikään itseisarvo. Multimodaalisten käyttöliittymien järkevällä kehittämisellä voidaan silti saavuttaa suuria etuja ja kehittää ennen tuntemattomia palveluita.

Viiteluettelo.

- [Althoff *et al.*, 2001], Frank Althoff, Gregor McGlaun, Björn Schuller, Peter Morguet and Manfred Lang, Using multimodal interaction to navigate in arbitrary virtual VRML worlds. *ACM International Conference Proceeding Series*, ACM Press, **15**, 2001, 1–8.
- [Bricken, 1991] Meredith Bricken, Virtual reality learning environments: potentials and challenges. *ACM SIGGRAPH Computer Graphics* **25**, 3, (Jul. 1991), 178–184.
- [Bryson *et al.*, 1994] Steve Bryson, Steven Feiner, Frederick Brooks, Philip Hubbard, Randy Pausch and Andries van Dam, Research frontiers in virtual reality. In: *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press, 1994, 473-374.
- [Claydon, 2004] Mark Claydon, Puhekäyttöliittymä osana virtuaaliodellisuusympäristöjä. Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, Julkaisu **B-2004-3**, 54-69.
- [Cruz-Neira *et al.*, 1993] Cruz-Neira, Leigh, Barnes, Cohen, Das, Engelmann, Hudson, Papka, Siegel, Vasilakis, DeFanti and Sandin, Scientists in Wonderland: A Report on Visualization Applications in the CAVE Virtual Reality Environment. *Proceedings of IEEE 1993 Symposium on Research Frontiers in Virtual Reality*, 1993, 59-66.
- [Feiner and MacIntyre, 1996] Steven Feiner and Blair MacIntyre, Future multimedia user interfaces, *Multimedia Systems* **4**, (1996), 250-268.
- [IBM] Why IBM? - Leadership in multimodal. International Business Machines, <http://www-306.ibm.com/software/perovasive/multimodal/>, viitattu 12.12.2005.
- [Le Hors *et al.*, 2001] Arnaud Le Hors, Chuck Powers and Håkon Wium Lie, XHTML+Voice Profile. World Wide Web Consortium, <http://www.w3.org/Submission/2001/13/>, viitattu 12.12.2005.
- [Maes and Ferial, 2002] Stephane Maes and Chummun Ferial, Multi-Modal Browser Architecture – Overview on the support of multi-modal browsers in 3GPP, World Wide Web Consortium, <http://www.w3.org/2002/mml/2002/MM-Arch-Maes-20010820.pdf>, viitattu 16.11.2005.
- [Mäkitammi, 2006] Satu Mäkitammi, Multimodaalisten Non-WIMP käyttöliittymien arkkitehtuurit. Ilmestyy raportissa Roope Raisamo (toim.), *Käyttöliittymien ohjelmistoarkkitehtuurit*, Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, 2006.

- [Nigay and Coutaz, 1993] Laurence Nigay and Joelle Coutaz, A design space for multimodal systems: concurrent processing and data fusion. In: *Proceedings INTERCHI'93*, 1993, 172-178.
- [Näsi, 2006] Juuso Näsi, Sovelluslogiikan eriyttäminen käyttöliittymästä. Ilmestyy raportissa Roope Raisamo (toim.), *Käyttöliittymien ohjelmistoarkkitehtuurit*, Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, 2006.
- [Opera] Multimodal Browser. Opera Software, <http://www.opera.com/products/verticals/multimodal/>, checked 12.12.2005.
- [Pape, 2001] Dave Pape, The CAVE Virtual Reality System, <http://www.evl.uic.edu/pape/CAVE/>, checked 15.12.2005.
- [Saarinen, 2006] Rami Saarinen, Multimodaalisuus, PAC-Amodeus ja moniagenttijärjestelmät. Ilmestyy raportissa Roope Raisamo (toim.), *Käyttöliittymien ohjelmistoarkkitehtuurit*, Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, 2006.
- [Zhang, 2004] Wei-wei Zhang, User Interface Design with New Techniques. University of Tampere, Department of Computer Sciences, Report **B-2004-4**, 116-131.

Opetuksellinen läsnäolo ja opettajan rooli verkko-opetuksessa

Suvi Peltomäki

Tiivistelmä.

Tämä tutkimus käsittelee tutkivan oppimisen yhteisöä ja erilaisia läsnäolon muotoja. Tutkimus syventää erityisesti opetuksellisen läsnäolon käsitettä ja tuo esille sen päätehtävät esimerkkeineen. Opiskelijan rooli vertaisryhmäkeskustelun johtajana määritellään aiempien tutkimusten ja esimerkkitapauksen avulla. Lopuksi pohditaan verkko-opiskelun valvontaa opettajan ja opiskelijan näkökulmasta.

Avainsanat ja -sanonnat: opetuksellinen läsnäolo, verkko-opetus, tietokonevälinen kokous, tutkivan oppimisen yhteisö.

CR-luokat: J.4, K.3.1

1. Johdanto

Internetin ja graafisten selainten leviäminen 1990-luvulla avasi mahdollisuudet opetuksen siirtämiseksi verkkoon. Verkko-opetus uutena opetusmuotona on kokenut monenlaisia haasteita, sillä sen lähestymistapa on erilainen kuin perinteisen luokkaopetuksen. Luokkaopetuksella on vuosisatojen, jopa muutaman vuosituhannen historia takanaan, jonka aikana se on muotoutunut varsin optimaaliseen muotoon. Luokkaopetukseen verrattuna verkko-opetus on vielä vasta alkutekijöissään ja siksi sitä yritetään parantaa tutkimalla uusia ratkaisuja ja soveltamalla niitä luokkaopetuksesta. Verkko-opetusmuotojen nykytutkimus keskittyy tutkimaan sellaisia aiheita kuin visualisointia, suosittelujärjestelmiä, yhteisöllisyyden luomiseen verkkokurssilla, eri läsnäolomuotojen mittaamista sekä erilaisten teknologioiden soveltamista oppimisympäristöissä.

Oppimiskokemuksen optimointi vaatii oikeat tekijät, joihin lukeutuvat mm. läsnäolon eri muodot. Garrison ja muut [2000] jaottelevat kurssiin liittyvät läsnäolon muodot sosiaalisiksi, kognitiiviseksi ja opetukselliseksi läsnäoloksi. Tämä tutkimus pyrkii käsittelemään näistä erityisesti opetuksellista läsnäoloa. Aihepiiriin liittyy ainakin seuraavat kysymykset: Miten jaotella läsnäolon komponentit mielekkäästi, että niitä voidaan arvioida määrällisesti ja laadullisesti? Missä suhteessa läsnäolon eri muotoja tarvitaan parhaimman mahdollisen oppimiskokemuksen aikaan saamiseksi? Kuinka välttyä opettajan ylikuormittamiselta kurssilla, jossa opiskelijat saattavat oman aikataulunsa takia tarvita apua läpi vuorokauden? Millaiset asiat kannattaa opettaa verkossa ryhmäkeskustelulla ja millaiset lineaarisena luentona?

Haaste tulevia oppimisympäristöjä ja niissä erityisesti sosiaalista läsnäoloa ajatellen on, että edelleenkin ei ole olemassa kasvokkain tapahtuvan vuorovaikutuksen korvaavia

eleitä, jotka olisivat yhtä tehokkaita verkossa. Sosiaalisen läsnäolon sovellusmuotoja on kuitenkin kehitelty verkkokursseille ja niistä esimerkkinä on Garrisonin ja muiden [1999] tutkimus. Lisäksi, opetuksellisen läsnäolon näkökulmasta, opettajan ja opiskelijan välinen kommunikointi ei ole vielä yltänyt luokkahuoneopetuksen tasolle. Toinen haaste olisi päästä verkko-opetuksessa yhtä korkealle ammattitaidon tasolle kuin vuosisatojen aikana tavallisen opetuksen avulla on päästy. Kyseessä eivät ole aivan vaatimattomat haasteet verkko-opetuksen lyhyen historian takia.

2. Tutkimusongelman kuvaus

Tutkimusongelma on opetuksellisen läsnäolon optimointi verkkokurssilla siten, että oppiminen ja opettaminen sujuvat tehokkaimmin olosuhteisiinsa nähden. Tutkimuskysymys voitaisiin muotoilla näin: millaiseksi opettajan rooli muodostuu verkko-oppimisympäristössä? Toisaalta myös: miten verkkokursseilla ilmenetään opetuksellista läsnäoloa? Mielenkiinnon kohteisiin kuuluu myös, mikä on opettajan rooli nykyisin järjestettävissä verkkokursseissa, samoin kuin minkälaista on suomalaisten opettajien mielenkiinto verkkokursseja kohtaan ja pystytäänkö niitä järjestämään jo peruskoulussa. Viimeisin aihe on karsittu pois tästä tutkimuksesta. Jatkotutkimusta ajatellen tutkimusta voisi syventää selvittämällä kuinka Suomessa koulutetaan verkkokurssien vetäjiä. Lukuisia erittäin mielenkiintoisia kysymyksiä löytyy siis selvitettäväksi.

Tämä tutkimus rajoittuu pääasiassa aihealueen kartoittamiseen sekä jo julkaistujen tutkimusten eheyttämiseen kokonaisuudeksi. Koska esimerkiksi Garrison ja muut [2001] ovat osoittaneet, että opetuksellista läsnäoloa voidaan mitata. Yleisesti ottaen tutkimus on kuitenkin kvalitatiivinen, sillä tutkimuksen ohessa ei suoriteta kvantitatiivisia kokeita vaan nimenomaan muiden tutkimuksiin nojaten yhdistetään tietoa selkeämmäksi kokonaisuudeksi. Opetuksellisen läsnäolon määrittely on pitkälti asioiden sovittamista tiettyihin kategorioihin; mikä toiminta luetaan opetukselliseen suunnitteluun, diskurssin mahdollisuuksien helpottamiseen tai suoraan ohjaamiseen. Kukin näistä kategorioista jakautuu osiin ja tämä tutkimus kuvailee, käsittelee, arvioi ja tarjoaa esimerkkejä näiden kategorioiden sisällöistä.

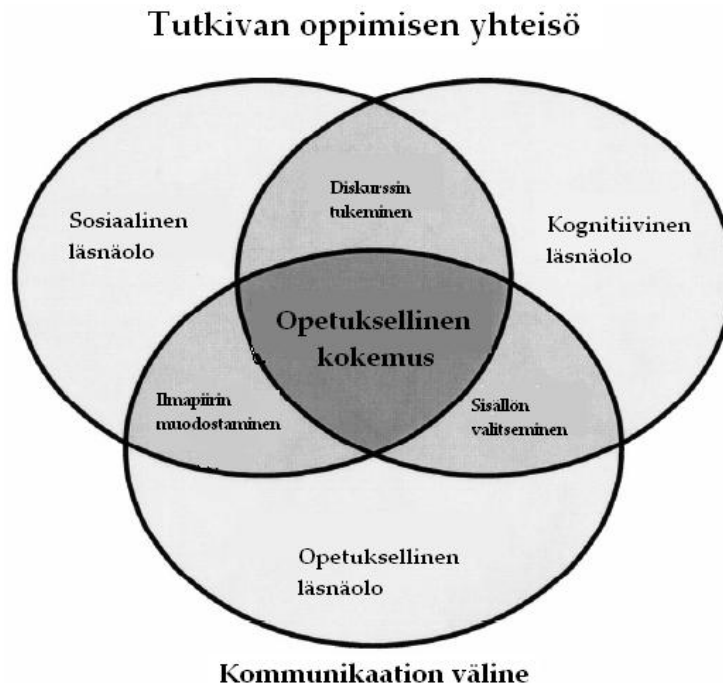
Voidaan olettaa, että opettajan rooliin netissä liittyvät teoriat ovat vieraita monille opettajille. Opettajat jakaisiin kolmeen ryhmään: opettajat, jotka eivät vedä verkkokursseja, eivätkä haluaisikaan vetää; ne, jotka eivät vedä verkkokursseja, mutta sopivassa tilanteessa olisivat kiinnostuneita hyödyntämään verkkoa; ja ne, jotka jo nykyään vetävät verkkokursseja. Toisin sanoen, tutkimuksen tarkoitus on tuoda julki opetuksellisen läsnäolon rooli monissa muodoissaan, mistä opettajat voisivat ammentaa tietoutta verkkokurssiopetukseensa.

3. Läsnäolon muodot verkkopohjaisessa oppimisympäristössä

Internetin välityksellä tapahtuva opetus on haaste tekniselle toteutukselle, haaste opetta-

jalle ja haaste itse opiskelijalle. Oppimisympäristöjä voidaan rakentaa ja ottaa käyttöön, mutta niiden mielekkyys käyttäjille on tärkein komponentti. Mikäli käyttö ei ole mielekästä, käyttöä tuskin jatketaan kovin kauaa epäonnistuneiden kokeilujen jälkeen. Luokkahuonetilannetta mukailleen, läsnäolo on tärkeää: ilman opiskelijan kognitiivista läsnäoloa oppimista ei tapahdu, koska tietoa teorioista tai toimintatavoista ei edelleenkään voida siirtää esimerkiksi johdolla ihmisen aivoihin. Ilman sosiaalista tai opetuksellista läsnäoloa kognitiivinen läsnäolo ei onnistu optimaalisesti eikä oppiminen ei ole tehokasta. Alla on määritelty mitä kukin näistä läsnäolotyypeistä tarkoittaa.

3.1. Läsnäolon määritelmät



Kuva 1: Tutkivan oppimisen yhteisö. Muokattu alkuperäisestä [Garrison et al., 2000].

Tutkimukseni pääaihe, opetuksellinen läsnäolo, on osa kolmen läsnäolotyypin kokonaisuutta *tutkivan oppimisen yhteisö* -mallissa, (katso Kuva 1). Tätä mallia pidetään välttämättömänä tehokkaan verkko-oppimisen saavuttamiseksi. Sosiaalinen läsnäolo määritellään Garrison ja muiden [2000] mukaan ”oppilaiden kykyä tuoda itseään esiin sosiaalisesti ja emotionaalisesti tutkivan oppimisen yhteisössä (community of inquiry).” Shortin ja muiden [1976] mukaan sosiaalinen läsnäolo on ”muiden projisointikyky median välityksellä tapahtuvassa kommunikaatiossa ja siitä seuraava ihmisten välisten suhteiden heijastuminen”. [Short et al., 1976, p. 65]. Kognitiivinen läsnäolo määritellään Garrisonin ja muiden [2000] mukaan sinä kuinka laajasti minkä tahansa tutkivan oppimisen yhteisön/ ympäristön opiskelijat onnistuvat konstruoimaan merkityksiä ylläpidetyn kommunikaation kautta. Opetuksellinen läsnäolo puolestaan pitää sisällään kognitiivisen ja sosiaalisen prosessin mahdollistamisen ja ohjaamisen henkilökohtaisesti merkityksellisen ja opetuk-

sellisesti merkitsevien oppimistulosten realisoimiseksi [Garrison et al., 2000].

Kukin läsnäolon muodoista tukee niin vahvasti toisiaan, ettei niitä sinänsä pysty erottamaan toisistaan tinkimättä oppimisprosessin laadusta. Tavallaan opetuksellinen ja sosiaalinen ovat nimenomaan kognitiivisen läsnäolon tukielementtejä ja pystyvät sinänsä olemaan erillään kognitiivisesta läsnäolosta. Esimerkiksi liiallinen opetuksellinen läsnäolo voi johtaa siihen, ettei opiskelija itse joudu näkemään riittävästi vaivaa asian itselle selvittämisen eteen, kun opettaja pureskelee asian hänelle valmiiksi jättämättä tilaa omalle päätteilylle. Sosiaalinen läsnäolo puolestaan voi johtaa siihen, että oppimispäämäärät unohtuvat täysin ja painotus siirtyy vain sosiaaliseen kanssakäymiseen. Kutakin kolmesta läsnäolotyypistä on hyvä olla, mutta tasapainotus niiden välillä on taitolaji.

3.2. Opetuksen sosiaalinen komponentti

Opettajan välittömyys (teacher immediacy) on tärkeä sosiaalisen läsnäolon osa. Sillä tarkoitetaan "sanatonta ilmaisua, joka vähentää fyysistä ja psykologista välimatkaa opettajien ja opiskelijoiden väliltä." [Garrison et al., 2000]. Mitä lyhempi tämä välimatka on, sitä helpompaa heidän välinen vuorovaikutuksensa on. Tämä opettajan sosiaalinen läsnäolo parantaa oppimista, sillä opiskelijoiden on helpompi ymmärtää asioita, kun opettaja saa ylläpidettyä opiskelijoiden mielenkiintoa omalla käytöksellään. [Rourke et al., 1999]

Luokkatilanteessa opettajan välittömyys tarkoittaa esimerkiksi, että opettaja katsoo yleisöön heille puhuessaan, kutsuu opiskelijoita nimeltä ja liikkuu rennosti. Koska verkkokursseilla on paljon oppilas-oppilas- ja oppilas-opettaja -vuorovaikutusta, eivätkä kasvokkain tavatessa ilmenevät sosiaaliset eleet ole mahdollisia, on verkkoon kehittynyt oma sosiaalinen elekielensä. Hyvää oppimista tukevan, lämpimän, avoimen ja luottavaisen ilmapiirin luominen on yleensä opettajan tehtävä, mutta verkossa myös opiskelijat ottavat opetuksellisia rooleja. Tällöin opetuksellinen läsnäolo ja välittömyys ovat paitsi itse opettajan, myös opiskelijoiden aikaansaamia. [Rourke et al., 1999.]

3.3. Opetuksellinen läsnäolo

Perinteisessä luokkahuoneopetuksessa opettaja osallistuu aina opetustilanteeseen. Verkkop opiskelu ei tässä mielessä poikkea perinteisestä mallista, vaan myös siellä opettaja on useimmiten läsnä, vaikkei opettaja fyysisesti olekaan samassa paikassa opiskelijan kanssa. Verkkokurssilla itseopiskelu korostuu enemmän kuin luokkahuoneessa, mutta opettajalta voi saada kuitenkin aina palautetta tarpeen tullen. Puhdas itseopiskelu ilman ohjaajaa eroaa edellisistä siksi, että vaikka henkilö saisi asiantuntijatiedon jostain lähteestä, koska kukaan ei ole antamassa palautetta oppimisprosessin etenemisestä ja onnistumisesta. Itseopiskelu johtaa usein epäorganisoiutuun oppimisprosessiin eikä se välttämättä pidä sisällään muuta kuin ulkoa opiskelua.

Opettajalla on useanlaisia vastuita kurssin tyypistä ja monimuotoisuudesta riippuen; joskus enemmän, joskus vähemmän. Yleisesti opettajien ajatellaan olevan opetuskokemuksen suunnittelijoita, asiantuntijoita, tiedon välittäjiä ja oppimisen ohjaajia. Nämä roolit siirtyvät myös verkkoon, vaikka Anderson ja muut [1999] niitä hieman eri sanoin kuvaile-

vatkin. He kirjoittavat, että opetuksellinen läsnäolo pitää sisällään oppimisjärjestyksen (learning sequences) suunnittelua, eksperttiedon jakamista asiasta sekä aktiivisen oppimisen mahdollistamista ja helpottamista [Anderson et al., 2001a]. Luvussa 4 pureudutaan tarkemmin näihin opetuksellisen läsnäolon ominaispiirteisiin kategorioittain.

4. Opetuksellisen läsnäolon päätehtävät

Anderson ja muut [2001] ovat identifioineet kolme päätehtävää ja vastuualuetta, jotka on otettava tärkeinä osina huomioon, jos halutaan verkkokeskustelun olevan arvokasta opiskelijoiden oppimisen kannalta. Opetuksellisen läsnäolon kolme päätehtävää ovat: 1. opetuksellinen suunnittelu ja organisointi 2. diskurssin mahdollisuuksien helpottaminen 3. suora ohjaaminen. Opetuksellinen läsnäolo alkaa jo ennen kuin kurssi itse, sillä se pitää sisällään myös opettajan ennakkoon tekemät suunnitelmat, järjestelyt ja valmistelut. Läsnäolo jatkuu koko kurssin läpi palautteen antamisen, osallistumiseen kannustamisen ja ohjaamisen muodossa. Opetuksellinen läsnäolo on äärettömän tärkeää kunnollisen oppimisen kannalta. Tämä on todettu jo luokkaopetuksessa, sillä opettajat toimivat asiantuntijan roolissa ja tarjoavat tiedon sellaisessa muodossa, josta opiskelijat voivat sen omaksua helposti.

Kaikkien näiden tehtävien toteuttaminen on aikaa vievä ja monimutkainen prosessi, mutta jokainen kohta on tärkeä keskustelun onnistumisen kannalta. Seuraavissa alakohdissa käsitellään tarkemmin kutakin roolia ja esitetään esimerkkejä siitä, kuinka ne ilmenevät varsinaisessa kontekstissaan eli verkkokurssin keskustelussa. Kustakin pääkategoriasta on tiivistetty kolme pääaktiiviteettiä, mutta mukaan liitetyistä taulukoista (Taulukko 1, Taulukko 2, Taulukko 3) löytyy useampiakin mainintoja erilaisista tehtävistä.

4.1. Opetuksellinen suunnittelu

Opettajan ensimmäinen päätehtävä on suunnitella opetuksellinen kokemus, joka pitää sisällään opetuksen suunnittelua ja hallinnointia, kuten myös osaamisen arviointia. Anderson ja muut [2001] viittaavat Laurillardin ja muiden [2000] työhön ja kertovat olevansa samaa mieltä, että opettajan tehtävä on rakentaa kertomuksellinen polku järjestelmän välityksellä opiskelijoille. Tämän polun on oltava ohjeita antava, ja sen tulee kertoa kaikista tehtävistä aktiviteeteista, jotta opiskelijat olisivat tietoisia niin kurssin suorista kuin epäsuorista tavoitteista ja tehtävistä. Yleisen, korkeamman tason kommentit opettajan suunnalta opiskelijoille kurssin etenemisestä ja sisällöstä ovat tärkeitä luomaan motivaatiota ja orientoimaan opiskelijoita.

Oppimisprosessin edistämiseen voidaan käyttää tietokonevälitteistä kokousta (computer mediated conferencing), joka voi tapahtua synkronisesti tai asynkronisesti. Tällaisissa konferensseissa tai keskusteluissa opettajan ja/tai muiden opiskelijoiden tuomat uudet näkökannat ovat vahvassa roolissa. Samalla opiskelija rakentaa ja tuottaa kognitiivisesti omia merkityksiä heijastelemalla omia mielipiteitään muiden mielipiteisiin. Tietokonevälitteinen kokous on siis hyvä tapa saada opiskelijat aktiivisesti mukaan

oppimisprosessiin. Opettajan merkitys tässä yhteydessä on järjestää kyseinen oppimistilanne ja tukea sen onnistumista antamalla palautetta ja avustamalla kurssilaiset pysymään oikeassa aiheessa.

Taulukossa 1 on esimerkkejä opettajan tehtävistä, jotka kuuluvat opetukselliseen suunnitteluun ja organisointiin. Kun keskitytään kurssiin kuuluvaan verkkokeskusteluun opetuksellisen suunnittelun vaiheessa, ovat opettajan kolme tärkeintä tehtävää valita kurssin sisällöstä keskusteluun sopivat aiheet, keskustelustrategian toteutus sekä keskusteluun osallistumisen vaatimusten ja odotusten määrittäminen.

Opetuksellinen suunnittelu ja organisointi

Tehtävä (indikaattori)	Esimerkki
Opintosuunnitelman asettaminen.	"Tällä viikolla keskustelemme seuraavasta aiheesta..."
Työskentelytapojen suunnittelu.	"Jaan teidät viiden hengen ryhmiin, joissa käytte keskenänne väittelyä..."
Aikataulun asettaminen.	"Tämä tehtävä pitää olla tehtynä perjantaihin klo 12.00 mennessä."
Apuvälineen tehokas hyödyntäminen.	"Kun lisäät kommentteja keskusteluun, yritä samalla ottaa kantaa myös muiden esittämiin kommentteihin."
Netiketin (netiquette) asettaminen.	"Pitäkää viestit kohtuullisen lyhyinä."

Taulukko 1: Opetuksellisen suunnittelun esimerkkejä. Muokattu alkuperäisestä [Anderson et al., 2001a].

4.2. Diskurssin mahdollisuuksien tukeminen

Opettajan toinen rooli on olla mukana sosiaalisen oppimisympäristön luomisessa, jotta oppiminen olisi aktiivista ja menestyksekkästä. Diskurssin helpottaminen ja tukeminen kurssilla on tärkeää, sillä se motivoi, ylläpitää mielenkiintoa ja saa opiskelijat osallistumaan paremmin käytäviin keskusteluihin. Myös luokkatilanteessa keskustelu opiskelutavasta asiasta jäisi melko vaisuksi, jos opettaja ei esimerkiksi tarjoa virikkeitä keskusteluun, jaa puheenvuoroja ja osoita kysymyksiä osallistujille. Opettajan kuuluu johdatella, muttei paljastaa ratkaisua, joten ryhmäkeskusteluissa parasta antia on antaa opiskelijoiden päätyä tiettyyn ratkaisuun diskurssin kautta.

Tähän tehtävään liittyen opettajan kuuluu käydä lukemassa ja kommentoimassa opiskelijoiden kirjoittamia kommentteja (post) säännöllisesti. Keskustelussa opettaja tarjoaa mallin toiminnalle kommentoimalla ja tukemalla opiskelijoiden osallistumista, kannustaen hiljaisempia opiskelijoita osallistumaan ja koittamalla supistaa niiden kommentteja, jotka ovat verkossa keskustelutavoiltaan suorastaan ylitsepursuvia. Opettajan rooli keskustelussa on täten huomattavasti opiskelijan roolia vaativampi.

Anderson ja muut [2001] ovat sitä mieltä, että keskustelun edistämiseen liittyvät toimenpiteet eivät kuulu esim. kurssin sisäiseen epäviralliseen "chattiin" vaan opettaja tekee

parhaansa osallistumalla varsinaiseen ryhmäkeskusteluun, kuten kaikki muutkin kurssilaiset – eri roolissa vain. Taulukossa 2 on listattuna esimerkkejä opettajan eri tehtävistä diskurssin mahdollisuuksien tukemisessa. Tärkeimmiksi tehtäviksi voisi tiivistää kurssilaisten houkuttelun keskustelemaan, samojen ja eriävien mielipiteiden tunnistaminen sekä oppimista tukevan ilmapiirin muodostaminen.

Diskurssin mahdollisuuksien tukeminen

Tehtävä (indikaattori)	Esimerkki
Samojen ja eriävien mielipiteiden tunnistaminen.	”Mikko, Suvi on tarjonnut vahvan vasta-argumentin esittämäsi mielipiteeseen. Haluaisitko kommentoida sitä?”
Konsensuksen tai yhteisymmärryksen löytäminen.	”Mikko ja Suvi, käytännössä olette asiasta samaa mieltä, vaikka ilmaisettekin sen hieman eri tavoin.”
Opiskelijoiden kontribuutioiden kannustaminen, tiedostaminen ja tukeminen.	”Kiitos selventävistä kommentteistasi!”
Oppimista tukevan ilmapiirin muodostaminen.	”Älkää olko liian kriittisiä. Ääneen ajattelu foorumilla on suorastaan suotavaa, sillä täällähän niitä ideoita olisi hyvä rakentaa.”
Keskustelun kannustaminen; opiskelijoiden houkuttelu kommentoimaan aihetta.	”Olisiko kellään tähän mielipidettä? Kertokaa ihan vapaasti mitä mieltä tästä asiasta olette.”
Prosessin tehokkuuden arviointi.	”Vaikuttaisi siltä, että keskustelu on poikennut pahasti sivuraiteelle. Yritetään keskittyä keskustelun varsinaiseen aiheeseen.”

Taulukko 2: Diskurssin mahdollisuuksien tukemisen esimerkkejä. Muokattu alkuperäisestä [Anderson et al., 2001a].

4.3. Suora ohjaaminen

Kolmas ja viimeinen opettajan päätehtävä Andersonin ja muiden [2001] mukaan, on toimia asiantuntijana, joka tietää opetettavasti asiasta huomattavasti enemmän kuin muut ja täten jakaa tietoaan muille. Tässä roolissa opettajan on myös avustettava opiskelijoita tarjoamalla suoraa ohjausta [Anderson et al., 2001a]. Materiaalin ja tiedon jakaminen opiskelijoille sekä kysymysten esittäminen on hyvin perinteistä tavallisessakin opetuksessa, joten suorasta ohjaamisesta on muodostunut yksi tärkeimmistä opettajalle kuuluvista toiminnoista.

Verkkokeskusteluissa, joissa pyritään toteuttamaan tutkivan oppimisen yhteisö -mallia (Kuva 1), eräs vaikeimmista asioista on saada opiskelijat pysymään aiheessa siten, että tärkeimmät asiat tulevat käsitellyiksi. Tämän tehtävän hoitaminen yleensä vaatii pedagogisesti melko pätevää opettajaa eikä vertaisryhmien keskusteluissa, kurssin opiskelijan ohjatessa keskustelua, välttämättä päästä yhtä hyvin tuloksiin juuri puutteellisten opetus-taitojen vuoksi. Vertaisryhmän ohjaaminen on kuitenkin tärkeätä kognitiivisen oppimisen kannalta, joten sitä käsitellään erikseen luvussa 5.

Anderson ja muut [2001] kuvailevat, että suoraan ohjaamisen kuuluu intellektuaalinen ja akateeminen ohjaaminen ja asiantuntijatiedon jakaminen opiskelijoille. Vaikka oikean-

laisen ilmapiirin luominen oppimiselle kuuluu diskurssin mahdollisuuksien tukemiseen, yhdistää Davie [1989] ilmapiirin luomisen myös tiedon jakamiseen kirjoittamalla, että "ohjaajan täytyy olla kykenevä asettamaan ja kommunikoidaan opiskelijoille kurssin tai seminaarin intellektuelli ilmapiiri ja toimia oppineen akateemikon (scholar) mallina. Opiskelijoilla ja opettajalla odotuksiin kuuluu, että opettaja kommunikoi opiskelijoille sellaista tietoa, jota on laajennettu opettajan henkilökohtaisilla intresseillä, innostuksella ja syvän tason ymmärryksellä."

Verkkokurssilla keskustelu voi joutua aivan väärälle raiteelle, mikäli kukaan ei ole toimimassa opettajan roolissa ohjaten pysymään asiassa. Sosiaalisen läsnäolon komponentti voi opiskelijoiden keskuudessa olla liian vahva, jolloin kurssilaiset keskittyvät höpöttelemään omiansa, eivätkä puhu laisinkaan kurssiin liittyvistä asioista. Tämän huomaa helposti myös tavallisessa luokkahuonetilanteessa, jossa pitäisi tehdä töitä ryhmässä. Mikäli tilannetta ei valvota tai sillä ei ole ohjaajaa, se johtaa pian siihen, että oppilaat keskustelevat kuulumisiaan, eivätkä työstä ryhmätyön aihetta. Nevgi ja Tirri [2000] raportoivat samanlaisista kokemuksista esimerkkikurssinsa kurssilaisten keskuudessa, missä varsinaisesta opiskeltavasta asiasta poikkeaminen keskustelussa aiheutti epämielekkyyden kokemuksia. Myöskään liiallinen toistelu ei toteuta oppimispäämääriä, ja siksi päuutta tietoa sisältämättömistä viesteistä ei todellisuudessa anneta ns. osallistumispisteitä (vrt. kognitiivista läsnäoloa mittaava pistetaulukko luvussa 6).

Suoran ohjaamisen kategoriaan kuuluvat tärkeimmät tehtävät voidaan tiivistää näihin kolmeen: materiaalin välittämiseen opiskelijoille, väärinymmärrysten diagnosointiin sekä arvioinnin ja palautteen jakamiseen.

Suora ohjaaminen

Tehtävä (indikaattori)	Esimerkki
Kysymysten tai sisällön esittäminen opiskelijoille.	"Mikko on tätä mieltä asiasta...Mitä mieltä itse olet siitä?"
Keskustelun ohjaaminen tiettyihin tarkkoihin aiheisiin.	"Tuota näkökulmaa ei hyödytä tätä keskustelua. Jospa keskittyisitte mieluummin tähän aiheeseen..."
Keskustelun yhteenveto, referointi.	"Keskustelu alkoi Villen kommentista...Juha esitti oman näkemyksensä aiheesta... Suvi jatkoi... Päädyimme lopputulokseen..."
Asian ymmärtämisen varmistaminen tarjoamalla arviointia ja selittävää palautetta.	"Liikut oikeilla jäljillä, mutta...vielä tällainenkin näkökulma olisi. Tämä on tärkeää, koska..."
Väärinymmärrysten korjaaminen.	"Muista, että puhumme yleisesti tietojenkäsittelyn opiskelijoista, emme vain vuorovaikutteisen teknologian opiskelijoista.."
Lisämateriaalin tarjoaminen eri lähteistä, esim. kirjoista, verkosta ja henkilökohtaisista kokemuksista.	"Keskustelin tästä aiheesta joskus professori McKenzen kanssa. Keskustelussa viitattiin artikkeliin, joka löytyy osoitteesta http://www... "

Taulukko 3: Suoran ohjaamisen esimerkkejä. Muokattu alkuperäisestä [Anderson et al., 2001a].

Anderson ja muut [2001] toteavat, että vain opettajan aktiivisen keskusteluun puutumisen avulla voidaan saada tietokonevälitteisistä kokouksista ja yhteistoiminnallisesta oppimisesta hyödyllisiä ja opettavia opetuskeinoja. Vaikuttaa siltä, että heillä olisi väheksyvä asenne itsenäisesti tai itseohjautuvasti (self-directed) tapahtuvaa oppimista kohtaan. Itsenäinen opiskelu toisaalta on kokonaan oma tutkimusalueensa, kun taas Anderson ja muut [1999, 2001] keskittyvät tutkimuksessaan käsittelemään tutkivan oppimisen yhteisöjä ja erityisesti verkkokeskusteluja.

4.4. Kuormittavuus opettajan kannalta

Harapniuk ja muut[1998] laskivat, että opiskelijoiden kanssa vuorovaikutus vei opettajalta noin 7,5 tuntia viikossa sellaisella verkkokurssilla, joka kesti 13 viikkoa. Tavallisilla verkon ulkopuolisilla yliopiston kursseilla opettaja on opiskelijoiden kanssa tekemisissä noin 2-6 tuntia viikossa; 6 tuntia, jos kurssissa on esim. kaksi luentoa ja yhdet harjoitukset viikossa. Jos kurssi on näin "intensiivinen", se harvemmin kestää kolmeatoista viikkoa. Näiden lisäksi on vielä lukuisia tunteja kurssin suunnittelua, luentojen valmistelua ja tehtävien tarkastamista. Koska tämän kaiken valmistelun lisäksi vielä viedään 7,5 tuntia opettajan aikataulusta yhteydenpitoon, on hänen aikataulunsa totisesti tiukka. Lisäksi verkossa tarvitaan tukea opettajalta usein iltaisin, joten kyseessä on myös kustannusseikka: kuka maksaa palkan kaikista ylityötunneista?

Useimmiten laadukkaan verkkokurssin suunnittelu vaatii usean ihmisen työpanoksen. Tarvitaan oppimisympäristön suunnittelijat, sen ohjelmoijat, kurssin sisällön tuottajat ja vielä ne, jotka ovat mukana ohjaamassa kurssia. Joskus toki ahkera opettaja hoitaa useamman näistä tehtävistä, mutta kaikkiin tuskin kukaan jaksaa panostaa. Vähintäänkin teknistä tukea useimmat opettajat tarvitsevat avukseen. Assistentin palkkaaminen saattaisi olla asiallista, koska verkkokurssilaiset tarvitsevat apua myös iltaisin. Kurssin materiaalin tuottaminen voi olla opettajalle niin aikaa vievä tehtävä, että assistentti saattaa joutua hoitamaan palautteen antamisen kokonaan.

5. Vertaiskeskustelu ja opetuksellinen läsnäolo

Luku käsittelee yleisesti verkkokursseilla käytössä olevaa toimintatapaa, jossa opiskelija nostetaan puheenjohtajan rooliin vertaiskeskustelussa. Luvussa esitellään myös aiheeseen liittyvää tutkimusta ja käydään läpi suomalaisen yliopistokurssin kautta millaisia ongelmia ja hyviä puolia tällaiseen oppimismuotoon saattaa liittyä.

5.1. Opiskelija opettajan roolissa

Opiskelija voi soveltaa verkkokurssilla opetuksellisen läsnäolon periaatteita kuin opettaja, jos hänet asetetaan esimerkiksi jonkun keskustelun johtajaksi tai puheenjohtajaksi (moderator). Keskustelun johtajan tehtävä on paljon samantyyppinen kuin opettajalla, sillä hänen tulisi pitää käynnissä keskustelua, ohjata keskustelun suuntaa ja tarpeen tullen

tarjota materiaalia kurssilaisille luettavaksi. Johtajan rooliin asettuminen tarjoaa opiskelijalle mahdollisuuden päästä kokeilemaan paitsi opettajan tai johtajan, myös asiantuntijan roolia, jolloin hän joutuu ottamaan keskusteltavasta asiasta tarkemmin selvää, jotta osaa vastata kaikkiin kurssilaisten esittämiin kysymyksiin ja väitteisiin. Johtajan tulee myös kyetä tarpeen tullen selittämään vaikeat asiat niin hyvin, että ne varmasti selviävät keskustelun osallisille. Verkkokeskustelussakin opiskelija pääsee siis soveltamaan kaikkia kolmea opetuksellisen läsnäolon päätehtävää.

Vertaiskeskustelun johtamisesta tekee erityisen mielenkiintoista esimerkiksi se, että opiskelija pääsee tai joutuu asiantuntijan rooliin, mikä vaatii opetusmateriaaliin perehtymistä. Opiskelija siis joutuu ensin oppimaan teorian voidakseen opettaa sen muille. Se, että ”opettaja” on osana opiskelijoiden vertaisryhmää, voi johtaa siihen, että hän kykenee muotoilemaan opetettavan asian niin, että vertaisryhmä ymmärtää sen helpommin. Voi kuitenkin käydä myös päinvastoin. Samanlaista opettamista on toki olemassa myös luokkatilanteessa [Schermerhorn et al., 1976], mutta verkko tarjoaa tähän joustavamman aikakehyksen. Keskustelufoorumit ovat hyvä esimerkki joustavuudesta, sillä ”opettajan” ei tarvitse vastata muiden kysymyksiin samalla hetkellä vaan hän voi tarpeen tullen perehtyä aiheeseensa lisää, palata asiantuntevan vastauksen kanssa ja jatkaa keskustelun virittämistä aiheesta opiskelijoiden keskuudessa.

5.2. Vertaisryhmien johtamista koskeva tutkimus

Vertaisryhmien ohjaamat verkkokeskustelut olivat tutkimuksen kohteena Rourken ja muiden [2002] tutkimuksessa. He tutkivat aihetta eräällä yliopiston verkkokurssilla, jossa opiskelijat oli jaettu ryhmiin ja kaikki ryhmät vuorotellen toimivat kurssin verkkokeskustelun puheenjohtajina. Keskusteluiden kommenttien sisällöt analysoitiin Taulukkojen 1-3 listaamien kategorioiden mukaisesti kvantitatiivisellä sisältöanalyysillä eli menetelmällä (vrt. [Angeli et al., 1998]), jonka Berelson [1952] määrittelee seuraavasti: ”objektiivinen, systemaattinen ja kvantitatiivinen kuvaus kommunikaation sisällöstä.” Lisäksi kurssilaisille tehtiin lomakehaastattelu ja osittain strukturoitu haastattelu.

Rourken ja muiden [2002] tutkimustulokset viittasivat kahteen asiaan, jotka verkkokursseja suunnittelevien tulee jatkossa muistaa. Ensinnäkin, että verkkokeskustelut ovat hyödyllisiä korkean tason ajatteluprosessien saavuttamiseen, mutta ei matalan tason oppimistavoitteiden saavuttamisessa. Keskustelujen käyttö jälkimmäisen saavuttamiseen on tehotonta ja aikaa vievää asynkronisessa tekstipohjaisessa muodossa. Toiseksi, korkeamman tason tavoitteisiin päästään, kun keskustelussa esitetään mielipiteitä, henkilökohtaisia kokemuksia ja analogioita, sillä nämä lisäävät ymmärrystä ja tekevät asioista konkreettisempia. Nimenomaan vastakkaiset perspektiivit häiritsevät keskustelijoiden alkuperäistä mielipidettä, jolloin asiaa joutuu pohtimaan tarkemmin. Tämä saavutetaan kriittisen interaktion avulla, koska nimenomaan ryhmän sisäinen konsensus eli samanmielisyys estää kriittisen ajattelun ja korkeamman tason ajatteluprosessien kehittymisen.

5.3. Vertaiskeskustelun johtamisen taitolaji suomalaisessa kontekstissa

Esimerkki kurssista, jolla vertaisryhmät johtivat pareittain keskustelua, on Tampereen yliopistossa järjestetty Internet-pohjaiset oppimisympäristöt -kurssi [IPOP, 2004]. Suuri osa opetuksellisen läsnäolon komponenteista jäi tukematta luultavasti sen takia, ettei johtajille oltu tarjottu tietoa etukäteen opetuksellisen läsnäolon vaatimuksista. Osa keskustelujen johtajista tarjosi lisämateriaalia ja moni teki parhaansa ylläpitääkseen keskustelua, mutta moni muu tehtävä jäi toteuttamatta. Varmasti moni opettaja ei ole näistä läsnäolon vaatimuksista myöskään tietoinen ja se onkin yksi syy tälle tutkimukselle.

Kuten IPOP-kurssilla [2004], myös muilla verkkokursseilla keskustelu harvemmin yltyy kovin vilkkaaksi ilman, että opiskelijat kokevat hyötyvänsä panoksestaan keskusteluun. Jos aktiivinen osanotto takaa hyvän arvosanan, opiskelijat ovat aktiivisempia kuin jos keskusteluun osallistuminen olisi täysin vapaaehtoista, eikä siihen sisältyisi positiivista kannustetta. Vertaisryhmäkeskusteluilla on potentiaalia tiettyjen yliopistossa käsiteltävien aiheiden käsittelyssä, mutta aivan peruskurssitason asioita on turha alkaa käsittelemään tällä tavoin. Opettajien tulisi siis huomioida tämä verkkokeskusteluja järjestäessään.

6. Osallistumisen valvonta verkkokursseilla

Luku käsittelee osallistumisen valvonnan ongelmaa suomalaisen yliopiston näkökulmasta, tutkijoiden tarjoamaa kognitiivisen läsnäolon arvosteluasteikkoa ja käy läpi teoria-tasolla, kuinka valvonta vaikuttaa opiskelijan käytökseen ja toimintaan verkkokurssilla.

6.1. Anonymiteetti suomalaisilla yliopistotason kursseilla

Tavallisessa yliopistossa opettaja tavataan yleensä ensimmäiseksi luentotilanteessa ennen kuin verkkokurssi käynnistyy. Opettajan läsnäolo kurssilla koetaan tästä johtuen paljon konkreettisemmaksi, koska jokainen opiskelija tietää kuka – niin nimeltä, ulkonäöltä, persoonalta kuin ammattiasemaltaan – verkkokurssilla lukee kaikki kommentit ja arvostelee työpanoksen, tai tavallisella kurssilla, kuka arvostelee viikkoharjoitukset ja tentin. Suomessa voi kuitenkin osallistua verkkokursseille, joissa opiskelija ja opettaja pysyvät toisilleen lähes täysin anonyymeinä nimeä ja verkkokeskustelussa ilmi käyviä ominaisuuksia lukuun ottamatta. On mahdollista, että tämä on tulevaisuudessa hyvinkin yleinen verkkokurssin toteutustapa.

Opiskelija siis tuntee kyllä opettajan, mutta valitettavan usein suomalaisessa yliopistossa kurssin vetäjällä ei ole aavistustakaan kurssilla istuvien opiskelijoiden nimistä. Täten luennoitsija tunnistaa ulkonäöltä tunnilla aktiivisesti kommentoivan, mutta ei osaa yhdistää kasvoja nimeen kurssiarvostelua antaessaan. Tuntiaktiivisuus on siis asia, joka jää vaille ansaitsemaansa arvostusta tavanomaisessa yliopistotason luentotilanteessa. Tästä poikkeuksena on toki pienryhmäopetus, jolloin opettaja oppii tuntemaan opiskelijat nimeltä jatkuvan vuorovaikutuksen ansiosta. Verkkokurssilla aktiivisuuden arvostelu on huomattavasti helpompaa.

6.2. Arvosteluasteikko verkkokeskustelun arvostelemiseen

Opettajille on kehitetty verkkokurssien arviointiin omia arvostelutaulukoita, joita opettaja voi soveltaa omalle kurssilleen sopivaksi. Yksi näistä arvostelutaulukoista on Bill Pelzin [2003] suunnittelema pisteasteikko, jonka mukaan verkkokeskusteluun lisättyjä kommentteja voidaan arvostella. Pisteitä kognitiivisesta läsnäolosta kyseisen pisteasteikon mukaan verkkokeskustelussa annetaan seuraavasti:

0 pistettä: Kommentti ei lisää akateemista arvoa keskusteluun: se ei tarjoa mitään uutta tietoa.

1 pistettä: Kommentti pitää sisällään ainakin yhden hyödyllisen faktan tai tietoa yleensä, mutta tieto on saatavilla myös oppikirjasta.

2 pistettä: Kommentti pitää sisällään ainakin yhden hyödyllisen faktan, jota ei löydy oppikirjasta.

3 pistettä: Kommentti lisää huomattavasti keskustelun akateemista arvoa, eikä se tieto löydy suoraan lukemalla oppikirjasta, ja joku aihe tai konsepti on selkeytetty.

4 pistettä: Kommentti pitää sisällään dokumentoitua tietoa aiheesta, joka edistää huomattavasti keskusteltavan asian ymmärrettävyyttä. Uusi tieto on selitetty ja sovellettu niin, että lukija oivaltaa käsitellyn materiaalin huomattavasti paremmin. [Pelz, 2003]

Opettaja voisi arvioida kurssilaisten osallistumista myös muun kuin kognitiivisen läsnäolon mukaan, mutta esimerkiksi sosiaalisen läsnäolon arvioiminen ei olisi läheskään yhtä mielekästä, koska kurssin painotus on oppimisprosessissa.

6.3. Valvonnan vaikutus käytökseen

Verkkokeskustelutilanne kehittää itsekontrollia, sillä kaikki oppimisympäristössä kirjoitetut kommentit ovat niin opettajan kuin muiden opiskelijoiden luettavissa. Toisin sanoen tieto siitä, että on tarkkailun kohteena minä hetkenä tahansa, pistää henkilön käyttäytymään omaa toimintaansa kontrolloiden eli päättömien kommenttien kirjoittelua vältellen ja velvollisuudet parhaansa mukaan täyttäen. Itsensä valvomisen teoriapohja tulee sosiologiasta Foucaultin [1977] Panopticon-valvontamalliin liittyvistä teorioista. Teoria liittyy instituutioihin, kuten vankilaan tai kouluun, ja koska myös verkko-opiskelu on osa instituution toimintaa, edellytetään myös siinä sopeutumista instituution sääntöihin. Opetuksessa opettaja pitää huolta sääntöjen noudattamisesta korjaamalla oppilaiden käyttäytymistä, kunnes he alkavat itse korjata ja kontrolloida omaa käytöstään. Kun siis opetussellinen läsnäolo on vahva, sitä varmemmin kurssi pysyy kuosissaan ja opiskelijat pitävät opetussuunnitelmassa.

Jo aiemmin mainitulla IPOPP-kurssilla [2004] opettaja valvoi osallistumista keskusteluihin. Opiskelijoiden vetämistä keskusteluista löytyi vain yksi tapaus, jossa opiskelija yritti erottua kurssilaisista lähes negatiivisella tavalla toteuttaen runsaasti sarkastisempaa kommentointityyliä. Todellisten sosiaalisten vihjeiden puutteessa tämä käytös aiheutti niin hyökkäviä vastareaktioita kuin sitä tukevia humoristisia kannanottoja. Opettaja ei

puuttunut keskusteluun, sillä kurssin verkko-osuudessa opetuksellista läsnäoloa toteuttivat enemmänkin kurssilaiset kuin itse opettaja. Sosiaalisen tilanteen vallitessa kurssilaiset nopeasti joko tuomitsevat negatiivisen käyttäytymisen tai alkavat tukea sitä, ja tässä tapauksessa tapahtui molempia. Kyseisen keskustelun puheenjohtaja ei puuttunut opiskelijoiden herjailuun kovin tiukasti, mutta silti ryhmän sosiaalinen paine johti siihen, että tuota negatiivista käytöstä ei enää nähty.

Läsnäolon ja aktiivisuuden seuraamisesta saattaa moni ahdistua jo normaaleilla kursseilla ilman verkko-osuutta. Moni haluaisi selvittää mahdollisimman vähällä työmäärällä ja olemalla mahdollisimman vähän paikalla, mutta on melkein itsestään selvää, että parempaa oppimista tapahtuu sosiaalisen ja kognitiivisen läsnäolon lisääntyessä. Suuri osa opiskelijoista jättäisi osallistumatta keskusteluun, jos sillä ei olisi merkitystä arvosanan kannalta, joten valvonta sanktioin (esim. bonuspisteitä antamalla tai vapauttamalla jostain toisesta tehtävästä) on tärkeää. Tästä löytynee lukuisia esimerkkejä verkkokursseilta ympäri maailman (esim. [Kognition filosofia, 2005]).

7. Yhteenveto

Sitä mukaa mitä verkko-opetus yleistyy, muokataan tavanomaiseen opetukseen liittyviä teorioita verkkoon sopiviksi. Opetuksellinen läsnäolo on noteerattu jo useissa verkkokursseissa, mutta lisätietoa aiheesta tarvitaan. Erityisesti opiskelijoiden vertaiskeskusteluissa ilmaisema opetuksellinen läsnäolo ja sen tehokkuus ovat kiinnostavia aiheita, joista tarvittaisiin lisätutkimusta. Verkko-opiskeluun liittyvää läsnäoloa tutkineen tutkimusryhmän verkkosivulta [Communities of Inquiry, 2005] voi löytää lisätietoa aiheeseen liittyvästä tutkimuksesta. Kuten tässä tutkimuksessa todettiin, aivan perusasioita ei ole hyötyä viedä tietokonepohjaiseen ryhmäkeskusteluun vaan ne tulisi opettaa jollain toisella tavalla. Kun kuitenkin halutaan syventää opiskelijoiden tietoutta jostakin tietystä kurssin asiasta, erittäin hyvä tapa on perustaa keskustelufoorumi, johon jokainen kurssilainen osallistuu kommentillaan ja mielipiteillään. Tämä menetelmä tukee tiedonhaun taitoja ja lähdekritiikkiä. Samalla monipuolinen ja kognitiivisesti vahva keskustelu rakentaa opiskelijan miellekarttaa.

Haasteita tulevien oppimisympäristöjen kehittämisessä riittää, sillä nykyisistä on vielä todella pitkä matka optimaalisiin ympäristöihin. Opetuksellisen läsnäolon muotojen on kehityttävä, samoin sosiaalista läsnäoloa indikoivien eleiden, jotta oppimisen tavoite täytyisi tehokkaasti kognitiivisen läsnäolon kautta. Suomalaisten opettajien suhtautumista verkko-opetukseen olisi mielenkiintoista tutkia ja samalla myös sitä, kuinka opettajia valmistetaan kohtaamaan verkko-opetuksen haasteet ja ongelmat. Verkko-opetuksen muuttuminen ajan myötä luontevaksi opetustavaksi vaatii asennemuutoksia ja motivaation parantumista niin opiskelijoiden kuin opettajien puolelta. Aika näyttää tuleeko jossain vaiheessa sukupolvea, joka käy koko koulutusuransa etäopiskeluna verkon kautta, sillä tällä hetkellä siihen näyttäisi olevan vielä todella pitkä aika.

Viiteluettelo

- [Angeli et al., 1998] C. Angeli, C. Bonk and N. Hara, Content Analysis of Online Discussion in an Applied Educational Psychology Course. (1998) Available as <http://crlt.indiana.edu/publications/crlt98-2.pdf>
- [Angeli et al., 2003] C. Angeli, N. Valanides and C. Bonk, Communication in a Web-based conferencing system: The quality of computer-mediated interactions. *British Journal of Educational Technology* **34**, 1 (2003), 31–43.
- [Anderson et al., 2001a] T. Anderson, L. Rourke, D. R. Garrison and W. Archer, Assessing teaching presence in a computer conferencing environment. *Journal of Asynchronous Learning Networks* **5**, 2 (2001) Available as <http://www.aln.org/alnweb/journal/jaln-vol5issue2v2.htm>.
- [Anderson et al., 2001b] T. Anderson, L. Rourke, D. R. Garrison and W. Archer, Assessing teaching presence in computer conferencing context. *Paper presented at the Annual Meeting of the American Educational Research Association*. Seattle, USA. April 10-14 (2001).
- [Berelson, 1952] B. Berelson, *Content Analysis in Communication Research*. Free Press, Glencoe, 1952.
- [Brady et al., 2003] E. Brady and D. Bedient, The effects of teacher presence on student performance and attitudes. In: *WebCT Impact 2003, 5th Annual WebCT User Conference*. (2003.) Available as <http://booboo.webct.com/2003/papers/Brady.pdf> (22.02.2005).
- [Communities of Inquiry, 2005] Communities of Inquiry. Research into Online Communities of Inquiry. <http://www.communitiesofinquiry.com/>. (22.02.2005).
- [Davie, 1989] L. Davie, Facilitation techniques for the on-line tutor, In: R. Mason and A. Kaye (Eds.), *Mindweave: Communication, Computers and Distance Education*. Pergamon Press, Oxford, 1989, 74-85.
- [Foucault, 1977] Michel Foucault, *Discipline and Punish: The Birth of the Prison*, trans. from French by Alan Sheridan. Pantheon, New York, 1977.
- [Garrison et al., 2000] D. R. Garrison, T. Anderson and W. Archer, Critical inquiry in a text-based environment: Computer conferencing in higher education. *The Internet and Higher Education*, **2**, 2-3 (2000), 87-105. Also available as http://www.communitiesofinquiry.com/documents/Critical_Inquiry_model.pdf (22.02.2005).
- [Garrison et al., 2001] D. R. Garrison, T. Anderson and W. Archer, Critical thinking, cognitive presence, and computer conferencing in distance education. *American Journal of Distance Education*, **15**, 1 (2001). Also available as http://www.communitiesofinquiry.com/documents/CogPresPaper_June30_.pdf.
- [Harapnuik et al., 1998] D. Harapnuik, T.C. Montgomerie and C. Torgerson, Costs of Developing and Delivering a Web-Based Instruction Course. In: *Proceedings of WebNet 98: World Conference of the WWW, Internet, and Intranet*. Association for the Advancement of Computing in Education. Charlottesville. (1998).
- [IPOPP, 2004] Internet-pohjaiset oppimisympäristöt -kurssin materiaali, Tietojenkäsittely-

- tieteiden laitos, Tampereen yliopisto, 2004.
- [Kognition filosofia, 2005] Kognition filosofia -kurssi, Tietojenkäsittelytieteiden laitos, Tampereen yliopisto, 2005. Available as <http://mtl.uta.fi/~hk60418/kogfil.htm>. (20.12.2005)
- [Laurillard et al., 2000] D. Laurillard, M. Stratfold, R. Luckin, L. Plowman and J. Taylor, Affordances for learning in a non-linear narrative medium. *Journal of Interactive Media in Education* (2000). Also available as <http://www.jime.open.ac.uk/00/2/>.
- [Nevgi et al., 2003] A. Nevgi ja K. Tirri, Hyvää verkko-opetusta etsimässä: oppimista edistävät ja estävät tekijät verkko-oppimisympäristössä: opiskelijoiden kokemukset ja opettajien arviot. Suomen kasvatustieteellinen seura, Turku, 2003.
- [Pelz, 2003] B. Pelz, Applying research on presence to guide online discussion. *Sloan Consortium* (2003). Available as http://www.sloan-c.org/effective/details3.asp?LE_ID=35 (21.12.2005).
- [Rourke et al., 2002] L. Rourke and T. Anderson, Using peer teams to lead online discussion. *Journal of Interactive Media in Education* **1** (2001). Available as <http://www.emoderators.com/moderators/feenberg.html>.
- [Rourke et al., 2001] L. Rourke, T. Anderson, D. R. Garrison and W. Archer, Assessing social presence in asynchronous text-based computer conferencing. *Journal of Distance Education* (2001). Available as <http://www.aln.org/alnweb/journal/jaln-vol5issue2v2.htm> (23.02.2005).
- [Schermerhorn et al., 1976] S. M. Schermerhorn, M. L. Goldschmid and B. M. Shore, Peer Teaching in the Classroom: Rationale and Feasibility. *Improving Human Performance Quarterly*, **5**, 1 (1976), 27-34.
- [Short et al., 1976] J. Short, E. Williams and B. Christie, *The Social Psychology of Telecommunications*. Wiley, Toronto, 1976.

Web Services -teknologian käyttö tietojärjestelmäintegraatioon toiminnanohjausjärjestelmissä

Jari Rajanen

Tiivistelmä

Web Services on alustariippumaton teknologia, jossa yleisesti hyväksytyjä internet-tekniikoita ja standardisoituja rajapintoja käyttävät ohjelmistokomponentit kommunikoivat keskenään. Sitä voidaan käyttää kietomaan millä tahansa tekniikalla toteutettu ohjelmisto ulospäin standardisoidun rajapinnan tarjoavan kuoren sisään. Tämän taustalla olevasta ohjelmistojen toteutustavasta riippumattoman ohjelmistojen välisen vuorovaikutuksen mahdollistaman ominaisuuden vuoksi on Web Services -teknologiaa alettu soveltaa yritysten toiminnanohjausjärjestelmiin sekä tietojärjestelmäintegraatioon.

Avainsanat ja -sanonnat: Web Services, tietojärjestelmäintegraatio, toiminnanohjausjärjestelmä.

CR-luokat: D.2, H.4

1. Johdanto

Web Services -teknologiaa hyödyntävät ohjelmistot rakentuvat komponenteista. Nämä komponentit kommunikoivat toistensa kanssa käyttäen yleisesti hyväksytyjä internet-tekniikoita. Ohjelmistokomponenttien toiminnallisuus kuvataan palveluina, joita nämä komponentit joko tarjoavat toisille komponenteille tai käyttävät toisten komponenttien tarjoamina. Web Services -teknologian kehitys on ollut vilkasta viime aikoina, kun suuret ohjelmistotalot ovat nähneet Web Services -teknologian edut ja ovat alkaneet kehittää ja hyödyntää sitä yritysten tietojärjestelmissä. Web Services -teknologian tärkeimpiä ominaisuuksia on, että palveluiden kuvaamisessa sekä informaation välityksessä käytetään XML-kieltä sekä yleisesti hyväksytyjä standardeja. Tämä mahdollistaa sen, että eri ohjelmistotuottajien Web Services -komponentit pystyvät kommunikoimaan sujuvasti keskenään.

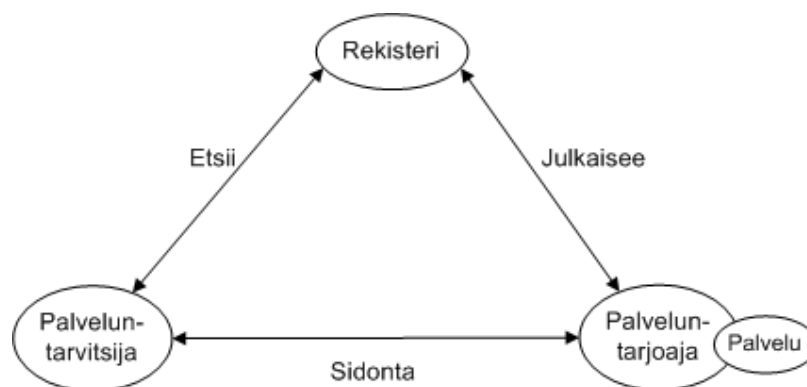
Aiempiin järjestelmiin verrattuna Web Services -teknologia tarjoaa korkeamman abstraktiotason ohjelmistojen rakenteeseen. Se muodostaa uuden abstraktin kerroksen olemassa olevien ohjelmistojen yläpuolelle ja kytkee XML-kielen käytön ohjelmien, objektien, tietokantojen ja kokonaisvaltaisten ohjelmistojen väliseen kommunikointiin. Tämä korkea abstraktiotaso piilottaa täysin pohjalla käytetyt ohjelmointitekniikat. Juuri tähän abstraktiin kerrokseen ja

standardisoitujen protokollien käyttöön perustuu mahdollisuus hyödyntää Web Services -teknologiaa ohjelmakomponenttien ja ohjelmistojen integrointiin. Web Services -teknologia ei itsessään ole mitään suoritettavia ohjelmistokomponentteja, vaan se nojautuu jollain ohjelmointikielellä tai skripteillä kirjoitettuihin suoritettaviin ohjelmistoihin. Web Services -teknologia määrittelee tehokkaan abstraktiokerroksen jota voidaan käyttää toteuttamaan ohjelmalta ohjelmalle tapahtuvaa vuorovaikutusta käyttäen olemassa olevaa verkkoinfrastruktuuria. Mutta se ei ole käyttökelpoinen ilman taustalla toimivaa ja tukevaa infrastruktuuria.

2. Web Services -teknologia

World Wide Web Consortiumin (W3C) määritelmän mukaan Web Service on ohjelmistojärjestelmä, joka identifioidaan käyttämällä Uniform Resource Identifieria (URI), ja jonka julkiset rajapinnat ja sidokset on määriteltä ja kuvailtu käyttämällä XML-kieltä. Tämän määrittelyn voivat toiset ohjelmistojärjestelmät havaita. Nämä järjestelmät voivat siten olla vuorovaikutuksessa Web Servicen kanssa kuten sen määritelmässä on kuvattu käyttämällä XML-pohjaisia viestejä, jotka kuljetetaan internet-protokollia käyttäen [W3C, 2004]. Koska kaikkien käytettyjen standardien perustana on XML, Web Services -teknologia on alusta- ja kieliriippumaton.

Web Services -teknologia toteuttaa palveluarkkitehtuurimallin (service-oriented architecture (SOA)), joka mahdollistaa löyhän kytkennän järjestelmien välille. Arkkitehtuuria on havainnollistettu kuvassa 1. Palvelun tarjoaja määrittelee palvelun rajapinnan ja toteuttaa sen toiminnallisuuden. Palvelun asiakas puolestaan kytkee tämän palvelun ohjelmistoonsa dynaamisesti. Varmistaakseen, että ulkopuoliset asiakkaat voivat palvelua käyttää, palvelun julkaisija julkaisee palvelurekisterissä tiedot palvelusta ja sen toiminnoista. [Sommerville, 2004]

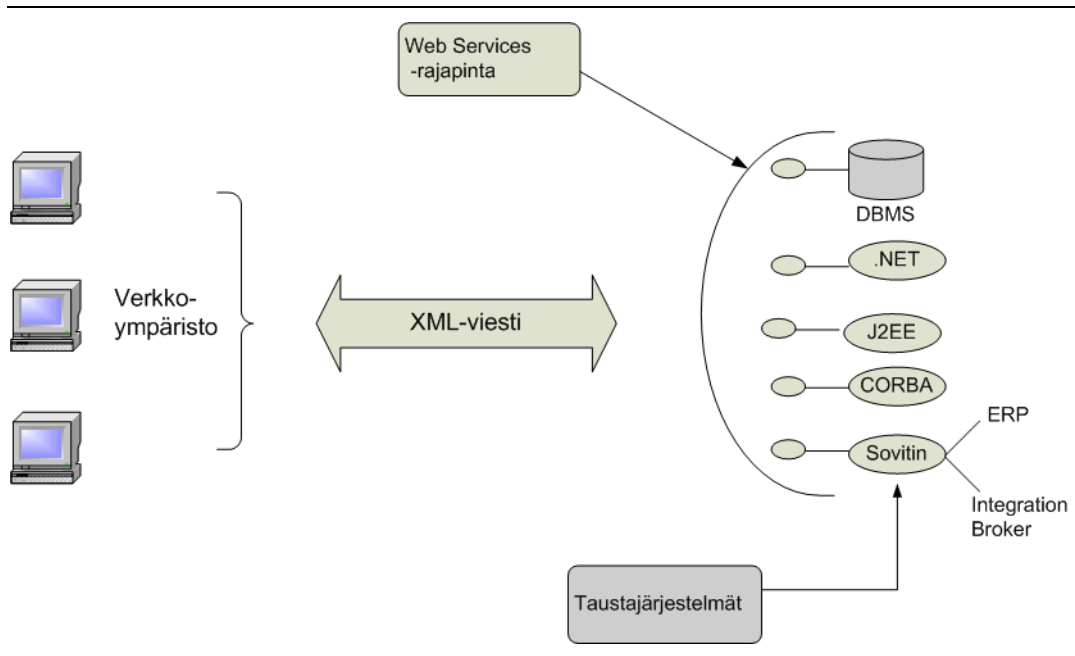


Kuva 1 Palveluarkkitehtuurin konseptuaalinen malli [Sommerville, 2004]

Arkkitehtuurin mukaisia palveluita voivat tarjota mitkä tahansa palvelun tarjoajat niin organisaation sisä- kuin ulkopuolella. Näin voivat organisaatiot luoda ohjelmistonsa yhdistelemällä palveluita joukolta palveluntarjoajia. Web Service on erityinen palvelutyyppejä, jota voidaan jakaa sekä käyttää hajautetun web-pohjaisen sovelluksen komponenttina. Ne kommunikoivat usein jo olemassa olevan taustajärjestelmän kuten esimerkiksi tilaus- tai asiakkuudenhallintajärjestelmän kanssa.

Aikaisempiin järjestelmiin verrattuna Web Services -teknologia tuo uuden abstraktiotason ohjelmistojen infrastruktuuriin ja sisältää semanttisen informaation liitettynä tietoon. Web Services ei pelkästään määrittele tietoa, vaan sisältää myös tiedon informaation prosessoinnista ja jakamisesta pohjalla oleviin ohjelmistosovelluksiin ja sieltä ulospäin. Web Services -teknologia on kuori tai kääre, joka voidaan kytkeä mihin tahansa sovellukseen, middleware-järjestelmään, tietokannan hallintajärjestelmään tai ohjelmistopakettiin. Web Services ovat ohjelmiin liitettyjä XML-sovelluksia. Ne lähettävät viestin muotoon luotuja XML-dokumentteja verkon yli ja mahdollisesti ottavat myös viestejä vastaan. Web Services -teknologia on uusi kerros, toisenlainen tapa tehdä asioita. Mutta se ei ole perustavaa laatua oleva muutos, joka poistaisi tarpeen käyttää nykyistä tietojärjestelmäinfrastruktuuria. Tämä uusi teknologiakerros esittää uuden toiminnon, uuden tavan työskennellä. Mutta kaikkein tärkeintä kuitenkin on, että se tarjoaa korkeammalla abstraktiotasolla määritellyn integrointimekanismin. [Newcomer, 2002]

Newcomerin [2002] mukaan Web Services voi toimia kuorena pohjalla toimivalle ohjelmistolle ja siten peittää näkyvistä palvelun toteuttavan tekniikan. Web Services -teknologia tarjoaa XML-kielen avulla määritellyn standardisoidun rajapinnan palvelulle, johon asiakkaat voivat liittyä löyhän sidonnan välityksellä. Paradigman etuna on, että tällöin ei asiakkaan edellytetä olevan jonkun tietyn tyyppinen. Kuvassa 2 havainnollistetaan, kuinka taustajärjestelmä voidaan kääriä Web Services -teknologiaa käyttäen ja tarjota verkkoympäristöön päin standardisoitu rajapinta kommunikointiin taustajärjestelmien kuten tietokannan hallintajärjestelmän, .NET:n, J2EE:n, CORBA:n ja muihin järjestelmiin liittävän sovitin kanssa. Web Services -rajapinta ottaa vastaan standardin XML-viestin verkkoympäristöstä, muuntaa viestin muotoon, jota kyseessä oleva taustajärjestelmä ymmärtää sekä mahdollisesti palauttaa vastausviestin. Web Servicen toteuttavan, taustalla toimivan ohjelmiston toteuttamiseen voidaan käyttää mitä tahansa ohjelmointikieltä, käyttöjärjestelmää tai middleware-ohjelmistoa.



Kuva 2 Web Services -rajapinta taustajärjestelmään [Newcomer, 2002]

Ohjelmille, jotka ovat vuorovaikutuksessa tietoverkon yli, täytyy olla mahdollista löytää toisensa, havaita tieto, joka mahdollistaa etäyhteyden ja kuvata, mitä odotetut vuorovaikutuksen muodot ovat – yksinkertainen kysely/vastaus vai monimutkaisempi prosessi. Sekä neuvotella sellaiset palvelun laatuksymykset kuten turvallisuus, viestien luotettavuus ja transaktioiden muodostus.

XML-kielen ja HTTP-protokollan lisäksi keskeisimmät Web Servicesin käyttämissä standardeista ovat Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) ja Universal Description, Discovery and Integration (UDDI). UDDI on avoin alustariippumaton kehys palveluiden kuvaamiseen, löytämiseen ja integrointiin internetin välityksellä. WSDL on XML-pohjainen kieli Web Services -teknologialla tuotettujen palvelujen määrittelyyn ja kuvaukseen. WSDL-kuvaus määrittelee palvelun rajapinnan eli määrittelee sen kuvauksen ja kutsutavan. SOAP on XML:ään perustuva protokolla jota käytetään tiedon siirtämiseen verkossa. Palvelun tarjoaja määrittelee tarjoamansa palvelun kuvauksen WSDL-kieltä käyttäen UDDI-rekisterissä. Palvelua tarvitseva asiakassovellus löytää tarvitsemansa palvelun kuvauksen UDDI-rekisteristä tai on sen muuten saanut tietoonsa. Asiakas kytkeytyy palveluntarjoajaan dynaamisesti ja kommunikoi palvelun kanssa verkon yli SOAP-sanomia käyttäen. [Newcomer, 2002]

3. Toiminnanohjausjärjestelmät

Toiminnanohjausjärjestelmät (Enterprise Resource Planning System ((ERP)), ovat suuria integroituja järjestelmiä, jotka on suunniteltu tukemaan yrityksen

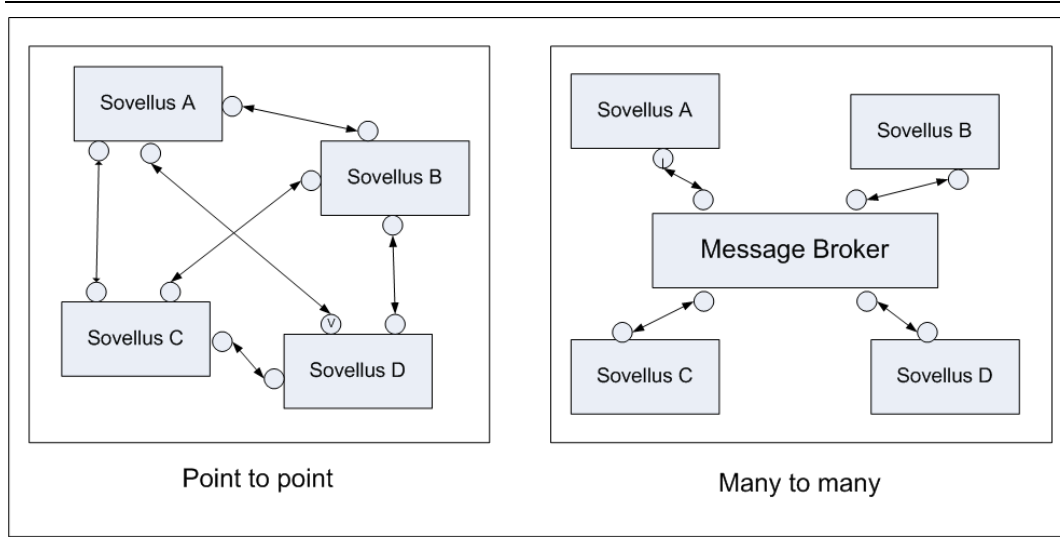
liiketoimintaprosesseja kuten tilausten käsittelyä, laskutusta, materiaalin hallintaa sekä tuotannon suunnittelua. ERP-järjestelmät toteutetaan yleensä valmisohjelmistoina, jotka sovitetaan organisaation tarpeisiin valitsemalla lukuisia asetuksia. ERP-ohjelmistot sisältävät liiketoimintaprosessien parhaat käytännöt, englanniksi *best practice*, usealta eri yritystoiminnan alalta. Koska ERP-ohjelmistojen katsotaan sisältävän parhaat käytännöt tavallisimmilta liiketoiminnan alueilta yritykset pyrkivät muokkaamaan toimintatapojaan usein siten, että ne ovat yhteensopivia ERP-ohjelmistojen kanssa. Näin uskotaan liiketoiminnan rationalisoituvan ja näin saavutettavan kilpailuetua muihin alan yrityksiin nähden taikka tämän nähdään olevan ainakin elinehto. [Cardoso et al., 2004]

Komponenttipohjaisesta ohjelmistokehityksestä on tullut pääasiallinen trendi yritysten ohjelmistoissa [Stal, 2002]. ERP-sovellukset toteutetaan valitsemalla yrityksen liiketoiminnan edellyttämät ohjelmistokomponentit ja asettamalla tyypillisesti tuhansia asetuksia jotta ohjelmisto saadaan toimimaan yrityksen toimintamallin mukaisesti. Arkkitehtuuriltaan toiminnanohjausjärjestelmät ovat viime aikoina olleet palvelinarkkitehtuurin mukaisia [Cardoso et al., 2004].

4. Tietojärjestelmäintegraatio

Tietojärjestelmäintegraatiolla (Enterprise Application Integration ((EAI)), tarkoitetaan yritysten eri tietojärjestelmien yhteensovittamista. Useat yritysten liiketoimintaprosesseista koskettavat useampaa yrityksessä käytettyä sovellusta. Näiden sovellusten tulisi pystyä kommunikoimaan keskenään reaaliajassa, jotta yritys pystyisi toimimaan tehokkaasti ja kommunikoimaan ulkomaailman kanssa. Tyypillisesti yrityksissä on vuosien saatossa syntynyt sovellussarekkeita jotka eivät pysty keskustelemaan keskenään. Uusien korkean tason sovellusten luominen edellyttää usein, että eri järjestelmiä joudutaan integroimaan keskenään.

Kuvassa 3 havainnollistetaan point-to-point ja Message Broker -arkkitehtuureita. Ensimmäiset integrointistrategiat perustuivat point-to-point -arkkitehtuuriin. Tällöin edellyttää jokainen uusi liittymä kahden ohjelman tai komponentin ohjelmointia olemassa oleviin sovelluksiin. Jos integroitavia sovelluksia on useita, tulee näiden sovellusten välisien yhteyksien määrästä ylivoimaisen suuri. Tällöin on point-to-point -tekniikka aikaa vievä ja kallis. Message broker on yksinkertaisempi arkkitehtuuri, kun keskenään kommunikoivia sovelluksia on useita. Message brokeria kutsutaan usein nimellä *middleware*. Perusideana on pienentää rajapintojen määrää ja jos joku sovellus muuttaa muotoaan täytyy vain yhtä rajapintaa muuttaa, nimittäin message brokeriin liittyvää. [Johanneson and Perjons, 2000]



Kuva 3 Point-to-point ja message broker -arkkitehtuurit [Johannesson and Perjons, 2000]

Stal [2002] toteaa, että useimmat saatavilla olevat EAI-sovellukset ovat patentoituja, vaikeahkoja käyttää eivätkä toimi kovin hyvin toistensa kanssa. EAI-teknologiat eivät voi itsenäisesti ratkaista integroimisongelmaa, koska ne yrittävät ratkaista ongelmaa joukolla patentoituja teknologioita. Tarkastellaan erimerkiksi tilannetta, jossa yritetään integroida järjestelmiä usean eri yrityksen välillä. Tällöin olisi usein tarve integroida järjestelmiä, jotka jo itsellään ovat integroitua käyttäen eri EAI-sovelluksia. Tällöin kohtaavat sovelluskehittäjät rekursiivisen ongelman integroida integroitua järjestelmiä.

5. Web Services toiminnanohjausjärjestelmissä

Suurimmat tietokoneohjelmistojen tuottajat tekevät suuria investointeja kehittääkseen ja edistääkseen Web Services -teknologian käyttöä ERP-järjestelmissä. Toteutuksilleen ovat yritykset antaneet useita eri nimiä. IBM kutsuu omaansa nimellä Web Services, Microsoft viittaa tuotteeseensa nimellä .NET, Oracle käyttää nimeä Network Services ja Sun puhuu tuotteesta nimellä Open Network Environment [Cardoso et al., 2004]. Kaikilta suurilta ERP-järjestelmien toimittajilta löytyy Web Services -rajapinnan tarjoava komponentti järjestelmään. ERP-järjestelmiä, jotka nykyisin vallitsevan käytännön mukaisesti perustuvat komponenttipohjaiseen ohjelmistokehitykseen, ei kuitenkaan ole yhdellä kertaa muutettu kokonaan Web Services -teknologian mukaisista komponenteista koostuviksi, vaan muutos halutaan toteuttaa vaiheittain. Perinteisten ERP-järjestelmien korvaamista kertarysäysmenetelmällä (big bang) on pidetty liian suurena riskinä. [Gillmann et al., 2002]

Yritykset ovat huomanneet, että heidän ERP-järjestelmät, joiden on tarkoituksena tehostaa ja taloudellistaa liiketoimintaa ovat itsessään erittäin suuria ja kalliita ohjelmistoja. Koska nykyisin käytettävät ERP-järjestelmät ovat valmiita suuria ohjelmistoja, ei niiden resursseja voida helposti supistaa. Eli käytännössä asiakkaat ostavat ja asentavat liiketoimintalogiikkoja, joita ne eivät koskaan käytä [Gillmann et al., 2002]. Monet yritykset hakeutuvatkin pois laajoista yhden toimittajan ERP-järjestelmistä kohti pienempiä ohjelmistoja, jotka tarjoavat suurempaa mukautuvuutta, toiminnallisuutta joka vastaa paremmin yrityksen tarpeita ja pienempiä kustannuksia [Cardoso et al., 2004].

Kun yritys rakentaa toiminnanohjausjärjestelmänsä käyttämällä Web Services -teknologiaa, sen ei tarvitse sitoutua vain yhden toimittajan järjestelmään vaan se voi valita jokaisen tarvitsemansa ohjelmistokomponentin parhaaksi katsomaltaan toimittajalta. Näin ollen ERP-järjestelmää valittaessa ei tarvitse tehdä kompromisseja järjestelmän ominaisuuksien suhteen valitsemalla vain yksi toimittaja, jonka järjestelmä on kokonaisuutena paras, vaan voidaan luoda lukuisten toimittajien komponenteista juuri omaan liiketoimintaan sopiva järjestelmä, jossa jokainen komponentti on juuri se paras vaihtoehto. Web Services -teknologiaa hyödynnettäessä ei tarvitse myöskään hankkia tarpeettomia ohjelmistokomponentteja. Komponentit ovat myös yksinkertaisempia vaihtaa ja lisätä siinä tapauksessa, jos yrityksen liiketoimintamalli muuttuu ja tietojärjestelmälle asetetaan uusia vaatimuksia.

6. Web Services ja tietojärjestelmäintegraatio

Elektroninen liiketoiminta synnyttää uusia toimintatapoja yhteistyölle ja yhteistoiminnalle niin eri yritysten välillä kuin yritysten ja kuluttajien välilläkin. Se mahdollistaa uusien liiketoimintamallien ja tuotteiden synnyn. 90-luvun lopulta lähtien on yritysten toiminta suuntautunut internetiin ja web-pohjaisten sovellusten käyttöön. Mutta yritykset ovat saaneet pettymykseen monesti huomata, etteivät niiden ERP-järjestelmät kykene jakamaan informaatiota liiketoimintapartnereiden kesken [Kezman and Rozman, 2002]. Monet yritykset kehittävät elektronista kaupankäyntiään kehittämällä business-to-business (B2B) -linkkejä hallitakseen arvoketjujaan paremmin. Organisaatioiden välisten prosessien automatisointi edellyttää järjestelmien saumatonta yhteistoimintaa [Cardoso et al., 2004]. Palveluarkkitehtuuria pidetään tällä hetkellä parhaana käytäntönä yrityksen tietojärjestelmiä kehitettäessä [Gorton and Liu, 2004].

Yrityksen tietojärjestelmien integroiminen vaatii kahden erilaisen integroinnin käyttämistä. Kun integroidaan yrityksen sisäisiä järjestelmiä käytetään EAI-sovelluksia tai integroidaan yritysten välisiä sovelluksia, tarvitaan B2B-järjestelmien integrointia. Jos voidaan käyttää samaa integrointitekniikkaa

molemmissa tapauksissa, täytyy sen johtaa kehitysprosessin nopeutumiseen ja taloudellistumiseen. [Chung et al., 2004]

Web Services -teknologia tarjoaa nämä edut. Web Services -ympäristössä kaikki järjestelmän osat, jotka ovat vuorovaikutuksessa keskenään, nähdään palveluina. Järjestelmälle ei ole mitään merkitystä missä nämä palvelut sijaitsevat. Siis ovatko ne organisaation sisäisiä vaiko ulkopuolisia.

7. Yhteenveto

Web Services -teknologian mukaiset ohjelmistot rakentuvat komponenteista, jotka kommunikoivat toistensa kanssa käyttäen yleisesti hyväksytyjä internet-teknikoita ja joiden toiminnallisuus kuvataan palveluina. Komponenttien julkiset rajapinnat ja sidokset kuvataan käyttämällä XML-kieltä, ja teknologia on sen ansiosta alusta- ja kieliriippumaton. Arkkitehtuuriltaan Web Services -teknologia toteuttaa palveluarkkitehtuurimallin, joka mahdollistaa löyhän kytkennän järjestelmien välille. Web Services -komponentit kommunikoivat usein jo olemassa olevan taustajärjestelmän kuten esimerkiksi tilaus- tai asiakkuudenhallintajärjestelmän kanssa. Hyödynnettäessä Web Services -teknologiaa tuo se uuden abstraktiotason ohjelmistojen infrastruktuuriin. Web Services -teknologia voi toimia kuorena tai kääreenä, joka voidaan kytkeä mihin tahansa sovellukseen. Tällöin tämä kuori peittää näkyvistä palvelun toteuttavassa taustajärjestelmässä käytetyn tekniikan. Keskeisimmät Web Services -teknologiassa käytetyistä standardeista XML:n ja HTTP:n lisäksi ovat SOAP, WSDL ja UDDI. Ollessaan vuorovaikutuksessa toistensa kanssa komponentit lähettävät viestin muotoon luotuja XML-dokumentteja verkon yli ja mahdollisesti ottavat myös vastaan viestejä.

Yritysten tietojärjestelmissä on komponenttipohjaisesta ohjelmistokehityksestä tullut pääasiallinen trendi. Toteutettaessa ERP-sovelluksia tehdään se usein koostamalla ohjelmisto valitsemalla tarjolla olevista ohjelmistokomponenteista yrityksen liiketoiminnan edellyttämät komponentit. Rakennettaessa tietojärjestelmiä Web Services -teknologiaa hyödyntäen voivat organisaatiot luoda ohjelmistonsa yhdistelemällä palveluita joukolta erillisiä palveluntarjoajia ja valita jokaisen tarvitsemansa ohjelmakomponentin parhaaksi katsomaltaan toimittajalta. Näin toimittaessa yritysten ei tarvitse sitoutua vain yhden toimittajan järjestelmään. Komponentit ovat myös yksinkertaisia vaihtaa ja lisätä siinä tapauksessa, että yrityksen liiketoimintamalli muuttuu. Trendinä vaikuttaa olevan, että yritykset hakeutuvat pois laajoista yhden toimittajan ERP-järjestelmistä kohti pienempiä ohjelmistoja, jotka tarjoavat suurempaa mukautuvuutta, toiminnallisuutta joka vastaa paremmin yrityksen tarpeita ja pienentävät kustannuksia

Nykyisin on yrityksissä tyypillisesti useita tietojärjestelmiä. Tehokkaasti toimiakseen tulisi näiden sovellusten pystyä kommunikoimaan keskenään reaaliajassa. Käytetyt EAI-teknologiat eivät kuitenkaan ole pystyneet ratkaisemaan integrointiongelmia käytössä olevien sovellusten välillä yrittäessään sovittaa yhteen joukkoa patentoituja teknologioita. Myös elektroninen liiketoiminta synnyttää koko ajan uusia toimintatapoja yhteistyölle ja yhteistoiminnalle yritysten ja asiakkaiden välille. Jos voidaan käyttää samaa integrointitekniikkaa, sekä yrityksen sisäisten järjestelmien integrointiin, että integrointiin yrityksen ulkopuolisiin järjestelmiin, pienentää se kustannuksia. Web Services -teknologia näyttää vastaavan näihin integrointivaatimuksiin, sillä siinä ei ole mitään merkitystä sijaitsevatko integroitavat palvelut organisaation sisä- vai ulkopuolella.

Viiteluettelo

- [Cardoso et al., 2004] Jorge Cardoso, Robert B. Bostrom, and Amit Sheth, Workflow management systems and ERP systems: differences, commonalities, and applications. *Information Technology and Management* **5**, 3-4(July 2004), 319-338.
- [Chung et al., 2004] Sam Chung, Lai Hong Tang, and Sergio Davalos, A web service oriented integration approach for enterprise and business-to-business applications. In: *Proc. of the Web Information Systems – WISE 2004: 5th International Conference on Web Information Systems Engineering*, Brisbane, Australia, (Nov 2004), *Lecture Notes in Computer Science* **3306**, Springer, 510-515.
- [Gillmann et al., 2002] Michael Gillmann, Joachim Hertel, Cristoph G. Jung, Günther Kaufmann, and Michael Wolber, Cooking the web-ERP. In: *Proc of Confederated International Conferences CoopIS, DOA, and ODBASE 2002* (2002), *Lecture Notes in Computer Science* **2519**, Springer, 602-617.
- [Gorton and Liu, 2004] Ian Gorton and Anna Liu, Architectures and technologies for enterprise application integration, In: *Proc. of the 27th International Conference on Software Engineering* 726-727.
- [Johannesson and Perjons, 2000] Paul Johannesson and Erik Perjons, Design principles for application integration. In: *Proc. of Advanced Information Systems Engineering: 12th International Conference, CAiSE 2000*, Stockholm, Sweden, (June 2000), *Lecture Notes in Computer Science* **1789**, Springer, 212-231.

- [Kezman and Rozman, 2002] Bostjan Kezman and Ivan Rozman, Web services in ERP solutions: A managerial perspective, In: *Metainformatics: International Symposium, MIS 2002*, Esbjerg, Denmark, (Aug 2002), *Lecture Notes in Computer Science* **2641**, 177-179.
- [Newcomer, 2002] Eric Newcomer, *Understanding Web Services: XML, WSDL, SOAP, and UDDI*. Addison-Wesley, 2002
- [Sommerville, 2004] Ian Sommerville, *Software Engineering Seventh Edition*. Addison-Wesley, 2004.
- [Stal, 2002] Michael Stal, Web services: Beyond component-based computing, *Communications of the ACM* **45**, 10 (Oct. 2002), 71-76.
- [W3C, 2004] Web services architecture requirements. February 11, 2004. Available as <http://www.w3.org/TR/wsa-reqs/#RFC2396>.

Shakkimoottorin pelipuun perusteita

Jon Sahlberg

Tiivistelmä

Shakissa parhaan mahdollisen siirron löytäminen on tietokoneelle erittäin työlästä. Siinä missä hyvä ihmispelaaja vain silmäilee shakkilautaa ja valitsee siirtonsa muutaman parhaan siirron joukosta, joutuu shakkitietokone tekemään miljoonia laskutoimituksia pelkästään selvittämään mitkä siirrot ovat pelin sääntöjen mukaisia. Hyvä ihmispelaaja harkitsee jokaisessa pelipuun solmussa vain muutamaa siirtovaihtoehtoa ja pystyy näin arvioimaan pelitilanteen mahdollista kehittymistä erittäin monen siirron päähän pelipuun juuresta. Tietokone joka joutuu laskemaan useita siirtovaihtoehtoja jokaisessa pelipuun haarassa laskee lyhyessäkin pelipuussa jopa miljoonakertaisen määrän siirtovaihtoehtoja ihmispelaajaan verrattuna. Onneksi tietokoneenkaan ei tarvitse laskea kaikkia pelipuun haaroja, sillä on olemassa menetelmiä, joilla osa pelipuun haaroista voidaan jättää tutkimatta päätyen silti samaan lopputulokseen. Tämä tutkielma keskittyy shakkimoottorin algoritmeihin. Tutkielmassa en halua ottaa kantaa siihen miten tietokone voidaan opettaa pelaamaan shakkia tai mitkä tiedot ovat olennaisia pelitilanteen arvioimisen kannalta.

Avainsanat ja -sanonnat: tietokoneshakki, pelipuu.

CR-luokat: C.4

1. Johdanto

Tietokoneshakissa parhaan siirron etsiminen kaikkien mahdollisten siirtojen joukosta on erittäin työlästä. Levyn [1984] mukaan shakin pelipuun haarassa on keskimäärin noin 37 mahdollista siirtovaihtoehtoa. Tämä tarkoittaa sitä, että kahden siirtovuoron eli kahden plyn jälkeen pelipuu on haarautunut jo yli tuhanteen haaraan. Neljän plyn jälkeen haaroja on jo melkein kaksi miljoonaa. Pinnallisetkin pelipuun haut voivat olla erittäin aikaa vieviä, sillä tutkittavien solmujen määrä kasvaa eksponentiaalista vauhtia.

Yksinkertaiset hakualgoritmit kuten mini-max käyvät läpi pelipuun jokaisen solmun. Helppoissa peleissä kuten ristinollassa, jossa pelipuu on suhteellisen pieni, koko pelipuun läpikäyminen sujuu nopeasti ja parhaan siirron löytäminen ei ole ongelma hitaallakaan algoritmilla. Monimutkaisemmissa peleissä, kuten shakissa, jossa pelipuun voidaan sanoa olevan loputtoman pitkä ja kasvavan eksponentiaalista vauhtia on kiinnitettävä huomiota algoritmiin, jolla pelipuuta tutkitaan. Liian yksinkertaisella algoritmilla voi lyhyenkin pelipuun tai sen osan läpikäymiseen shakissa kulua jopa tunteja.

Koska shakin pelipuu on liian laaja tutkittavaksi kokonaan, on erittäin todennäköistä, että hakualgoritmi joutuu keskeyttämään etsintänsä jossain vaiheessa vaikka tutkittavaa pelipuun haaraa ei olisikkaan vielä käyty loppuun. Tästä johtuen osa siirroista, jotka myöhemmässä pelipuun vaiheessa osoittautuvat hyväiksi, jää tutkimatta. Tällaista tapahtumaa kutsutaan horisonttiefektiksi.

Pienessäkin osassa shakin pelipuuta miltei kaikki pelitilanteet esiintyvät useita kertoja. Jos tietokoneen käyttämä pelipuu rakennetaan siten, että tietokoneen ei tarvitse laskea samojen pelitilanteiden uusia esiintymiä tai niiden jälkeen muodostuvia pelipuita moneen kertaan shakin pelipuun kasvunopeus voisi oman arvioni mukaan jopa puolittua. Pelipuun kasvunopeuden puolittamisella tarkoitan sitä, että jokaisessa pelipuun haarassa on tarpeellista laskea keskimäärin vain puolet kaikista mahdollisista siirtovaihtoehdoista. Tämä vähentäisi shakkimoottorin laskemisen tarvetta eksponentiaalisesti alkuperäiseen verrattuna. Pelipuun kasvunopeus shakissa on kuitenkin niin nopeaa, että kasvunopeuden puolittamisen jälkeenkään kaikkia siirtovaihtoehtoja ei aina ehditä tutkia realistisessa ajassa.

2. Menetelmiä pelipuun läpikäymiseen

2.1. Mini-max

Yksinkertaisuudestaan huolimatta mini-max-algoritmi on erittäin käyttökelpoinen. Mini-max-algoritmi etsii pelipuusta siirtoja, jotka ovat mahdollisimman hyviä molemmille pelaajille. [Akl and Newborn, 1977] Algoritmissa kutsutaan, yleensä rekursiivisesti, vuorotellen funktioita, joista toinen suorittaa valkoisen pelaajan siirrot ja yrittää saada pelin tuloksen mahdollisimman suureksi ja toinen suorittaa mustan pelaajan siirrot yrittäen saada pelin tuloksen mahdollisimman pieneksi. Algoritmissa käydään syvyysuuntaisesti jälkijärjestyksessä läpi kaikki kussakin pelitilanteessa mahdolliset siirrot tiettyyn pelipuun syvyyteen asti. Kun yhden pelitilanteen kaikki lailliset siirrot on käyty läpi, funktiot palauttavat parhaan löytämänsä siirron tai sen tuloksen itseään kutsuneelle funktiolle.

2.2. Alpha-beta

Onneksi kaikkien siirtovaihtoehtojen läpikäyminen osoittautui turhaksi, sillä 60-luvun alkupuolella tekoälytutkijat keksivät alpha-beta-algoritmin jonka avulla suuri osa pelipuun haaroista voidaan jättää laskematta päätyen silti samaan lopputulokseen mini-max-algoritmin kanssa. [Levy, 1984] Alpha-beta-algoritmissa pelipuun haaran tutkiminen keskeytetään, jos vastustaja pystyy pakottamaan tilanteen itselleen paremmaksi kuin mihin pelaajat päätyisivät

jossain toisessa aikaisemmin tutkitussa pelipuun haarassa. Itse asiassa menetelmällä ei ole juuri muita eroja. Ehkä juuri siksi alpha-betaa pidetäänkin mini-max-menetelmän uutena versiona tai parannuksena.

Ainakin shakissa mini-max-algoritmin korvaaminen alpha-betalla ainoastaan nopeuttaa pelipuun hakutoimintoa. Molemmilla algoritmeilla samaan pelipuun syvyyteen tehdyt haut palauttavat aina saman tuloksen, alpha-beta vain palauttaa tuloksen yleensä monta kertaa nopeammin, sillä se käy läpi vain pienen osan siirroista mini-maxiin verrattuna. [Akl and Newborn, 1977] Alpha-beta-algoritmia käyttämällä on mahdollista jättää jopa yli 99,5 % pelipuun solmuista laskematta. [Levy, 1984] On myös tapauksia joissa alpha-beta-algoritmia käyttävä tietokone joutuu laskemaan kaikki pelipuun siirtovaihtoehdot. Tällaisissa harvinaisissa tapauksissa alpha-beta on kuitenkin vain yhtä hidas kuin mini-max-algoritmi.

2.3. Horisonttiefekti

Sekä mini-max- että alpha-beta-algoritmia koskeva ongelma shakissa on, että kummallakaan algoritmilla ei voida tietää miten peli kehittyy haetun pelipuun alueen ulkopuolella. Koska molemmat algoritmit etsivät parasta siirtoa yleensä tiettyyn pelipuun syvyyteen asti on mahdollista, että esimerkiksi heti seuraava siirto tarkastelun ulkopuolella johtaisi vastustajalle erityisen suotuisaan asemaan. Tällaista horisonttiefektiä voi siirtää vain pidemmälle pelipuussa, mutta siitä ei käytännössä voi koskaan päästä kokonaan eroon. On kuitenkin mahdollista yrittää arvioida tilanteen vaarallisuutta niin sanottua turbulenssia pelipuun lehtisolmuissa jotta saataisiin arvio voiko shakkietokone lopettaa kyseisen haaran tutkimisen nykyiseen syvyyteen, vai pitäisikö sen vielä jatkaa laskemista muutama puolisiirto pidemmälle. Pelitilanteen turbulenssia voi arvioida esimerkiksi sillä miten moni pelinappula on vastustajan uhan alla. Kaikkia mahdollisia peräkkäisiä uhkaussiiroja ei kuitenkaan voi tutkia, sillä useissa tapauksissa se aiheuttaa ketjureaktion, joka kasvattaa pelipuun liian suureksi tutkittavaksi kokonaan.

3. Muistin ja ajan kulutus sekä niiden hallinta

3.1. Shakin pelipuun ominaisuuksia

Shakin pelipuu kasvaa eksponentiaalista vauhtia. Pelin alussa molemmilla pelaajilla on 20 siirtovaihtoehtoa. Pelin edetessä yhdessä pelipuun haarassa voi olla jopa yli 40 siirtovaihtoehtoa. Koska siirtovaihtoehtojen määrä kasvattaa pelipuun kokoa eksponentiaalisesti, voidaan todeta, että kaikkia siirtoja ei ajan ja muistin riittämättömyyden vuoksi voi eikä kannata laskea tai tallentaa tietokoneen muistiin.

3.2. Pelitilanteen muistin kulutus

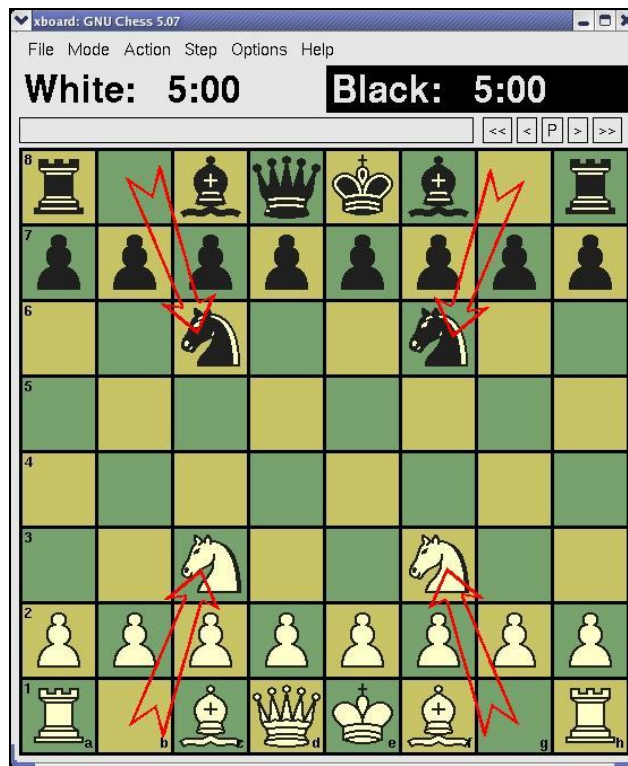
Yhden pelitilanteen tallentamiseen shakissa tarvitaan tieto nappuloista laudalla, siirtovuorosta, tornitusmahdollisuuksista ja ohestalyöntimahdollisuudesta. Lisäksi pelitilanteen täydellinen tallentaminen vaatii myös tietoja aikaisemmista siirroista ja lukumäärän miten monta puolisiirtoa on kulunut edellisestä sotilaan siirrosta tai nappulan lyömisestä.

Koska molemmilla pelaajalla on yhteensä kuusi erilaista pelinappulaa, tarvitaan yhden nappulan tallentamiseen vähintään neljä bittiä muistia. Siirtovuoroja on kaksi, mikä pystytään merkitsemään yhdellä bitillä. Tornitusmahdollisuuksia on molemmilla pelaajilla kaksi erilaista, joten kaikkien mahdollisten tornitusmahdollisuuksien kombinaatioiden tallentamiseen tarvitaan vähintään neljä bittiä. Ohestalyöntimahdollisuuden tallentamisen voi tehdä monella tavalla. Yksi tapa on tallentaa edellinen siirto muistiin, jolloin ohjelma voi päätellä, onko ohestalyönti mahdollinen. Yhden siirron tallentaminen onnistuu tallentamalla siirron lähtö- ja määränpääruudut. Tämä voidaan tehdä yhteensä 12 bitillä. On myös mahdollista, että ohjelma voi hakea edellisen siirron pelin siirtohistoriasta jolloin pelitilanteen ei tarvitse erikseen tallentaa edellistä siirtoa.

Laskennallisesti yhden pelitilanteen tallentaminen onnistuu ainakin 273 bitillä. Päästäksemme tasan 35 tavuun voimme tallentaa pelin edellisten siirtojen lukumäärän jäljelle jäävillä seitsemällä bitillä. Yhden pelitilanteen tallentamiseen kuluu melko vähän muistia, mutta jo viidestä kuuteen siirtoa syvän pelipuun tallentamiseen voi kulua muistia useita satoja megatavuja.

3.3. Päällekkäinen laskeminen

Muutamaa poikkeusta lukuun ottamatta lopputilanteen kannalta shakissa ei ole merkitystä, missä järjestyksessä kyseiseen tilanteeseen päätyneet siirrot on tehty. [Moreland, 2001] Voidaan olettaa, että valkealla pelaajalla on pelissä mahdollisuus tehdä siirrot A ja C ja vastaavasti musta pelaaja voi omalla vuorollaan tehdä siirrot B ja D. Kaikki edellä mainitut siirrot A, B, C ja D ovat laillisia siirtoja pelaajien omilla vuoroillaan. Siirrot on mahdollista järjestää neljään eri järjestykseen, joista jokaisella järjestyksellä päädytään samaan lopputilanteeseen. Esimerkkinä mainittakoon, että shakin alku tilanteesta valkea pelaaja voi tehdä siirrot Rb1-c3 sekä Rg1-f3 ja musta pelaaja siirrot Rb8-c6 sekä Rg8-f6. Nämä siirrot on esitetty kuvassa 1.



Kuva 1. Esimerkki tilanteesta joka johtaa päällekkäiseen laskemiseen pelipuussa.

Kuvassa 1 pelitilanne on sama riippumatta siitä, missä järjestyksessä siirrot on tehty. Siirtojen jälkeen ohjelman tarvitsee tutkia vain yksi neljästä pelipuun haarasta, sillä kaikki neljä siirtojen eri haaraa ovat päätyneet samaan tilanteeseen. Shakissa on olemassa muutamia poikkeuksia, joissa siirtoja ei voi tehdä missä tahansa järjestyksessä. Tällaisia tilanteita voi syntyä muun muassa shakkauksen ja tornituksen yhteydessä.

3.4. Hajautustaulut

Yksi tapa säästää muistia ja aikaa on säilöä pelitilanteita tai niiden tietoja hajautustauluun. Pelitilanteen paikka hajautustaulussa voidaan laskea pelitilanteesta generoitavalla satunnaisella avaimella. Koska pelitilanteita voi olla paljon enemmän kuin hajautustaulun avaimia, on mahdollista, että kahdelle eri pelitilanteelle joudutaan antamaan sama avain. Tällaisia tapauksia kutsutaan avainten yhteentörmäyksiksi ja niitä pyritään välttämään, sillä ne voivat sekoittaa koko shakkimoottorin toiminnan. Morelandin [2001] mukaan hajautustaulun avainten yhteentörmäysten välttämiseksi olisi hyvä käyttää mahdollisimman pitkiä hajautustaulun avaimia. Nykytietokoneilla 64 bittistä pitkä hajautustaulun avainta voidaan pitää riittävän pitkinä, sillä voidaan laskea, että tietokoneen muisti loppuu kesken ennen kuin hajautustaulu tulee liian täyteen ja avainten yhteentörmäysten todennäköisyys kasvaa liian suureksi. Joissain

tapauksissa saattaa riittää jopa 32-bittinen avain, riippuen siitä miten monta pelitilannetta hajautustauluun tarvitsee tallentaa ja miten suuri riski avainten yhteentörmäyksen mahdollisuudelle halutaan antaa.

Esimerkiksi 32 bittisellä avaimella ja 128 megatavun kokoisella muistilla avaimen yhteentörmäyksen todennäköisyys on

$$\left(\frac{128Mt}{35t}\right) \div 2^{32} = \frac{1}{1120}.$$

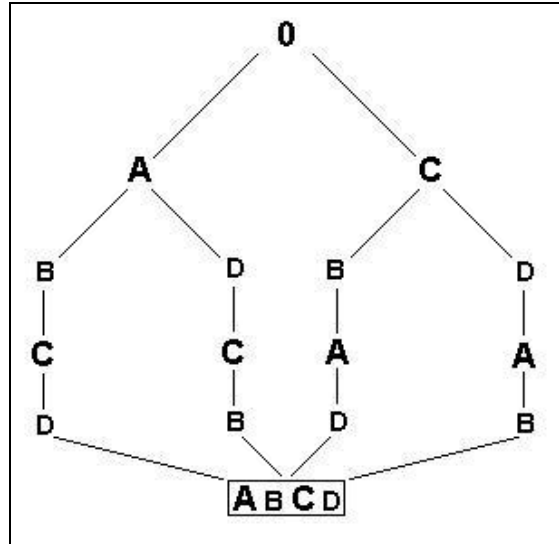
Puolestaan 64 bittisellä avaimella ja 1024 megatavun kokoisella pelipuulla kahden avaimen yhteentörmäyksen todennäköisyys on vain

$$\left(\frac{1024Mt}{35t}\right) \div 2^{64} = \frac{1}{601295421440}.$$

Törmäysten todennäköisyys on tietenkin sitä pienempi mitä vähemmän hajautustaulun avaimia on otettu käyttöön.

Kaikissa tilanteissa ei ole tarpeellista tallentaa koko pelitilannetta hajautustauluun. Joissain tapauksissa riittää kun hajautustauluun tallennetaan vain muutamia tietoja pelitilanteesta myöhempää käyttöä varten. Hyödyllisiä tietoja voisivat ovat esimerkiksi kyseisen pelitilanteen arvo, pelitilanteen paikka tai syvyysaste pelipuussa ja paras löytynyt seuraava siirto kyseisestä pelitilanteesta. Hajautustauluun tallennettuja tietoja voidaan käyttää esimerkiksi alpha-beta-menetelmässä nopeuttamaan parhaan siirron etsintää. [Moreland, 2001]

Hajautustaulun käyttäminen voi kuluttaa erittäin suuren osan tietokoneen muistista. Muistia ja suoritusaikaa kuitenkin säästyy, kun suuri osa jo lasketuista tilanteista tai vaikka alpha-beta-algoritmin alkuarvot voidaan hakea suoraan hajautustaulusta uudelleen laskemisen ja tallentamisen sijasta. Yksi hajautustaulujen käyttösovellus onkin pitää muistissa mitkä pelipuun tilanteet on jo laskettu, jolloin niitä ei tarvitse laskea uudelleen. Kuvassa 2 esitettävässä kaaviossa kuvitteelliset siirrot A, B, C ja D jakavat pelipuun neljään haaraan jotka kaikki johtavat lopulta samaan pelitilanteeseen. Hajautustaulua käyttämällä on mahdollista, että näiden neljän siirron jälkeen algoritmin tarvitsee laskea vain neljäsosa pelipuun kaikista siirtovaihtoehtojen kombinaatioista.



Kuva 2. Usein samaan pelitilanteeseen voidaan päästään monella eri tavalla.

3.5. Ajankäyttö siirtojen etsintään

Yksi vaikea ongelma shakkimoottorissa on ajankäyttö ja sen hallinta. Aikaisemmissa kappaleissa on keskitytty laskemisen tarpeen vähentämiseen pelipuun tutkimisen yhteydessä. Koska shakin pelipuu on erittäin syvä, pelipuun haku joudutaan usein katkaisemaan jossain vaiheessa ennen pelin päättymistä. Miten paljon aikaa voidaan käyttää yhden tilanteen etsintään ja miten syvän etsinnän tässä ajassa ehditään tehdä?

Levyn [1984] mukaan yksi erittäin yksinkertainen tapa hallita ajankäyttöä shakin pelipuun hakutoiminnossa on tehdä aina uusia, syvempiä etsintöjä alpha-beta-algoritmeilla kunnes yhden siirron etsimiseen varattu aika loppuu. Tällainen haku toimii erittäin tehokkaasti, jos alpha-beta-algoritmi aina ensimmäisenä tarkistaa edellisellä haulilla kyseisestä solmusta löytyneen parhaan siirron. Koska shakin pelipuu kasvaa eksponentiaalista vauhtia, voidaan olettaa, että seuraava haku vie aikaa vähintään yhtä kauan kuin edellinen. Kannattaa kuitenkin varautua myös siihen, että uusi syvemmälle ulottuva haku voi viedä moninkertaisesti aikaa edelliseen hakuun verrattuna.

Yksi mahdollisuus ajan kulutuksen arviointiin on laskea algoritmin suorituksen aikana montako siirtovaihtoehtoa pelipuun eri syvyytasoilla on. Laskurin tietojen perusteella algoritmi voisi arvioida seuraavan syvyytason siirtovaihtoehtojen määrää. Tällaisen arvion avulla algoritmi voisi päätellä, onko sillä vielä riittävästi aikaa suorittaa vielä yhtä tasoa syvempi etsintä pelipuussa.

4. Yhteenveto

Shakkimoottorin pelipuussa voi lyhyelläkin etsinnällä kuluttaa suuria määriä muistia ja aikaa. On olemassa monia tapoja nopeuttaa pelipuun hakualgoritmin toimintaa tai vähentää muistin kulutusta. Shakin pelipuu kasvaa kuitenkin niin räjähdysmäisesti, että parhaan siirron etsiminen pelipuusta on yleensä aikapulan takia katkaistava ennen kuin etsintä on päässyt pelipuun tai sen haaran loppuun saakka. Useassa tapauksessa on mahdollista nopeuttaa hakualgoritmin toimintaa käyttämällä osa tietokoneen muistista esimerkiksi hyvien tai vastaavasti huonojen siirtojen muistamiseen. Usein shakkimoottorin toiminta perustuukin hyvän suhteen löytämiseen algoritmin nopeuden, tarkkuuden ja muistin kulutuksen välille.

Viiteluettelo

- [Akl and Newborn, 1977] Selim G. Akl and Monroe M. Newborn, The principal continuation and the killer heuristic. In: *Proceedings of the 1977 Annual Conference of the ACM*, 466 - 473.
- [Levy, 1984] David N. Levy, *The Chess Computer Handbook*. B. T. Batsfort Ltd, 1984.
- [Moreland, 2001] Bruce Moreland, Programming Topics. <http://www.seanet.com/~brucemo/topics/topics.htm>.