

Hypertekstirakenteessa navigoiminen visualisoinnin avulla

Ilkka Lehtinen

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Tietojenkäsittelyoppi
Pro gradu -tutkielma
Maaliskuu 2004

Tampereen yliopisto

Tietojenkäsittelytieteiden laitos

Ilkka Lehtinen: Hypertekstirakenteessa navigoiminen visualisoinnin avulla

Pro gradu -tutkielma, 69 sivua, 6 liitesivua

Maaliskuu 2004

Tässä tutkielmassa selvitetään hypertekstirakenteessa navigoimisen ongelmallisuutta ja sen syitä. Navigoinnin suurimpana ongelmana on koettu olevan käyttäjien eksyminen hypertekstisivujen muodostamaan verkkoon. Eksymisen johdosta käyttäjät turhautuvat ja kokevat tiedonhaun vaivalloiseksi.

Tutkielmassa tarkastellaan mahdollisia ratkaisuja navigointiongelmaan, keskittyen erityisesti ongelman ratkaisemiseen visualisointien avulla. Tarkemmin tarkastellaan navigointihistorian visualisoinnista ja tämän hyödyllisyyttä käyttäjän kannalta. Visualisointien toimivuuden arvioimiseksi käydään läpi eri mahdollisuudet liittää nykyiseen WWW-ympäristöön visualisoinnin luova sovellus ja toteutetaan käytännössä navigointihistoriaa visualisoiva sovellus.

Sovelluksen toimivuutta testattiin laboratoriossa järjestetyillä testeillä. Testien tarkoituksena oli saada tietoa siitä, miten hyvin toteutettu visualisointi toimii käytännössä. Tutkimuksen tuloksena todetaan, että uusi sovellus ei poikkea merkittävästi nykyisistä vastaavista sovelluksista. Saatujen tulosten pohjalta hahmotellaan vaatimuksia navigointihistoriaa visualisoiville sovelluksille ja tältä pohjalta esitetään yksi mahdollinen käyttöliittymä.

Avainsanat ja -sanonnat: hyperteksti, navigointi, tiedon visualisointi, internet, navigointiongelma, WWW.

Sisällys

1.	Johdanto.....	1
2.	Navigointi hypertekstirakenteessa.....	3
2.1.	Navigoinnista yleensä.....	3
2.2.	Hypertekstiympäristö ja käyttäjän sisäinen malli.....	5
2.3.	Navigointiongelma hypertekstiympäristössä.....	7
2.4.	Navigoinnin tuki nykyisissä selaimissa.....	8
2.5.	Havaitut navigointimallit hypertekstirakenteessa.....	10
2.6.	Navigointimallit ajallisesti tarkasteltuina.....	12
3.	Tiedon visualisointi.....	13
3.1.	Tiedon visualisointi prosessina.....	14
3.2.	Visualisoinnin vuorovaikutteisuus.....	16
3.3.	Tiedon mahtuminen visualisointiin.....	18
3.4.	Tiedon esittäminen ja käytettävyys visualisoinneissa.....	21
3.5.	Visualisoinnin näyttäminen hypertekstiympäristössä.....	24
4.	Ratkaisumalleja navigointiongelmaan.....	26
4.1.	Yleiskuvaan perustuvat ratkaisut.....	27
4.2.	Käyttäjän toimet huomioon ottavat ratkaisut.....	29
4.3.	Sivujen esittäminen navigointityökaluissa.....	30
4.4.	Vaihtoehtoisia esitystapoja.....	32
4.5.	Sivujen järjestäminen navigointityökaluissa.....	33
4.6.	Uusi tapoja sivujen järjestämiseksi.....	35
5.	Käyttäjän selailuhistorian visualisoiva sovellus.....	37
5.1.	WWW-ympäristön toiminta käyttäjän navigoidessa.....	37
5.2.	Visualisoinnin luominen hajautetusti.....	38
5.3.	Visualisoinnin liittäminen WWW-selaimen.....	39
5.4.	NavigationHelper-sovelluksen ympäristö.....	40
5.5.	NavigationHelper-sovelman rakenne.....	42
5.6.	NavigationHelper-sovelluksen käyttöliittymä.....	43
6.	Käyttöliittymän testaaminen ja arviointi.....	49
6.1.	Testien järjestelyt.....	49
6.2.	Koehenkilöiden taustatiedot.....	50
6.3.	Käyttöliittymien käytössä havaitut erot.....	51
6.4.	Koehenkilöiden mielikuvat testistä ja käyttöliittymistä.....	55
6.5.	Testien arviointia.....	57
7.	Testin arviointi selainten jatkokehityksen kannalta.....	59
7.1.	Selailuhistorian jäsentäminen.....	59
7.2.	Selailuhistorian esittäminen.....	61

7.3. Kolmiosaisen selailuhistorian hahmotelma	63
8. Päätelmät	65
Viiteluettelo	67
Liite 1	70
Liite 2	71
Liite 3	72

1. Johdanto

Vaikka hyperteksti on ideana suhteellisen vanha, on se tullut ihmisten tietoisuuteen ja osaksi jokapäiväistä elämää laajemmin vasta 1990-luvulla. Varsinkin WWW:n (World Wide Web) keksimisen ja kehittymisen ansiosta hyperteksti tuli yleiseen käyttöön. Tänä päivänä hypertekstiä käyttääkin päivittäin suuri joukko ihmisiä.

Hypertekstin on havaittu olevan monessa suhteessa mainio keksintö, mutta ei täysin ongelmaton. Hypertekstin parhaina puolina on pidetty sitä, että sen avulla teksti voidaan esittää myös muussa muodossa kuin periteisessä lineaarisessa järjestyksessä. Hypertekstin avulla esitettyä tekstiä käyttäjä voi lukea haluamassaan järjestyksessä toisin kuin esimerkiksi perinteistä kirjaa.

Jo varhain hypertekstin käytössä havaittiin ongelmia, joista suurimmaksi muodostui käyttäjien eksyminen hypertekstisivujen muodostamaan verkkoon. Nykyään ongelma on vielä laajempi, koska varsinkin WWW:n sisältämä hypertekstisivujen määrä on jatkuvassa kasvussa. Erilaisia ratkaisuja ongelmaan on yritetty keksiä jo pitkään, mutta mittava tutkimus hypertekstin ympärillä käynnistyi vasta 1990-luvun keskivaiheilla, kun WWW tuli suuren yleisön käyttöön.

Tutkimusten avulla hypertekstiin perustuvien järjestelmien käyttöä on pyritty saamaan käyttäjäystävällisemmiksi. Koska nykyään hyvin suuri määrä tietoa on hypertekstinä, on luonnollista pyrkiä tekemään tiedon hakeminen tästä muodosta entistä helpommaksi ja sujuvammaksi. Uusien ratkaisuiden turvin ihmisten olisi helppo olla vuorovaikutuksessa hypertekstiin perustuvien järjestelmien kanssa ja hyötyä niistä nykyistä enemmän.

Kuten aiemmin todettiin, hypertekstissä navigoimisen ongelmallisuutena pidetään sitä, että käyttäjä eksyy liian helposti hypertekstisivujen muodostamaan verkkoon. Tämän ongelman ratkaisemiseen on monia eri vaihtoehtoja. Yleiskuvan näyttäminen hypertekstiympäristöstä on yksi eniten pohdituista ratkaisuista. Käyttäjän navigointihistorian näyttäminen on toinen ehdotettu ratkaisu. Tämän lisäksi on pohdittu mahdollisuutta kehittää hypertekstin selailemiseen käytettyjä sovelluksia, hypertekstin kuvailuun käytettyä HTML-kuvailukieltä sekä itse sivujen suunnittelua.

Tässä tutkielmassa käydään läpi erilaisia navigointia helpottamaan kehitettyjä ratkaisuita. Tarkastelu keskittyy hypertekstin selailemiseen tarkoitettujen työkalujen puutteisiin ja erityisesti kiinnitetään huomiota navigointihistorian hyödyntämiseen käyttäjän apuna. Kaikki osa-alueet hypertekstiympäristössä ovat osaltaan vaikuttamassa navigointiongelman syntyyn ja sen takia mitään yksittäistä osa-aluetta ei voi jättää tarkastelematta.

Aiheen käsittely on rajattu siten, että ratkaisuja yritetään hakea mahdollisimman läheltä nykyisiä toteutuksia. Pyrkimyksenä ei ole pohtia kokonaan uudenlaisen järjestelmän kehittämistä.

Tutkielma alkaa ihmisen navigoinnin tutkimisella eri ympäristöissä ja erityisesti hypertextiympäristössä. Navigointia tarkastellaan Spencen [1999] esittämän viitekehyksen mukaisesti ja tämän pohjalta hahmotellaan navigoinnin luonnetta ja tarkastellaan, miten sitä voidaan helpottaa. Luvun kaksi tarkoituksena on saada käsitys siitä, minkälaisena tapahtumana navigointi tulisi nähdä.

Kolmannessa luvussa tarkastellaan tiedon visualisointia, joka on yksi tapa lähestyä navigoinnin ongelmaa. Luvussa käydään läpi visualisointiprosessi lähtöaineistosta vuorovaikutteisen visualisoinnin näyttämiseen. Luvun lopussa tiedon visualisointia sovelletaan WWW-ympäristöön. Tiedon visualisointia käsitellään tarkemmin, koska se tarjoaa vaihtoehtoja navigoinnin helpottamiseksi.

Neljännessä luvussa käsitellään navigointiongelman ratkaisemista visualisointien avulla. Huomiota kiinnitetään siihen, miten visualisoinnin avulla pystytään tukemaan navigointia ja minkälainen suhde visualisoinnilla tulisi olla itse navigointiin. Tämän luvun tarkoituksena on koota yhteen toisen ja kolmannen luvun havainnot ja pyrkiä soveltamaan niitä navigointiongelman ratkaisemiseksi.

Koottujen havaintojen pohjalta on toteutettu sovellus, jonka avulla uusien ideoiden toimivuutta voidaan testata käytännössä. Viidennessä luvussa käydään läpi sovelluksen toteutus ja toiminta. Luvun alussa käydään läpi WWW:n rakennetta ja toimintaa, koska se muodostaa toteutetun sovelluksen käyttöympäristön ja samalla selviää ympäristön sovellukselle asettamat rajoitukset.

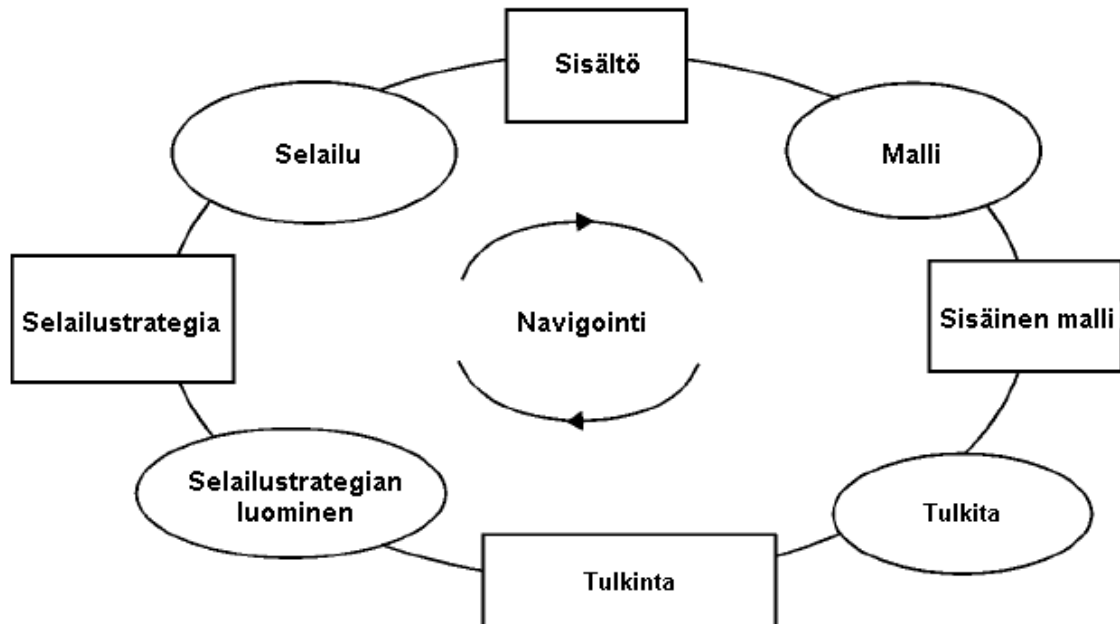
Kuudennessa luvussa kerrotaan testeistä, jotka tehtiin toteutetun sovelluksen arvioimiseksi. Sovellusta verrattiin nykyisten selainten tarjoamaan käyttöliittymään. Luvussa esitellään testeistä saadut tulokset ja arvioidaan toteutuksen hyödyllisyyttä ja soveltuvuutta tehtävänsä. Testien tuloksena saatiin paljon tietoa sovelluksen käytöstä ja käyttäjien mielipiteitä sovelluksesta. Seitsemännessä luvussa pohditaan saatujen tulosten perusteella, miten toteutettua sovellusta voitaisiin kehittää jatkossa nykyistä paremmaksi.

2. Navigointi hypertekstirakenteessa

Tässä luvussa käsitellään navigointia yleensä ja erityisesti navigointia hypertekstisivuilla. Ensin esitellään navigoinnin periaate yleisellä tasolla ja tarkastellaan navigointia toimintana. Tämän jälkeen tarkastellaan navigointia hypertekstisivuilla. Neljännessä ja viidennessä kohdassa analysoidaan nykyisten selaimien tarjoamia välineitä hypertekstisivuilla liikkumiseen. Lopuksi tarkastellaan käyttäjien liikkumista hypertekstisivuilla sekä rakenteellisesti että ajallisesti. Tämän tarkastelun perusteella ehdotetaan työkaluja navigoimisen helpottamiseksi.

2.1. Navigoinnista yleensä

Ihmiset ovat navigoineet kautta aikojen. reaali maailmassa navigoinnissa apuna ovat esimerkiksi maamerkit, majakat ja polut. Navigointi hypertekstirakenteessa tai muissa uuden median sovelluksissa poikkeaa kuitenkin huomattavasti tästä fyysisessä maailmassa tapahtuvasta navigoinnista. Vaikka navigointi on tavallisessa ja virtuaalisessa maailmassa huomattavasti erilaista, on kuitenkin löydettävissä myös samankaltaisuuksia.



Kuva 2.1. Kehys navigointitapahtumalle [Spence, 1999].

Navigointi on *sisäisen mallin* (internal model) luomista ja tulkitsemista [Spence, 1999]. Navigoidessaan ihminen havainnoi ympäristöään ja rakentaa siitä samalla sisäistä mallia. Tämän mallin avulla ihminen pystyy myöhemmin

liikkumaan ympäristössä ja etsimään sieltä kohteita mielekkäällä tavalla. Sisäistä mallia navigoinnin yhteydessä kutsutaan myös *kognitiiviseksi kartaksi* (cognitive map).

Spence [1999] ehdottaa navigoinnille eräänlaista kehystä (Kuva 2.1). Tässä navigointi nähdään prosessina, jolla on tietyt vaiheet. Navigoiminen alkaa *selailustrategian* (browsing strategy) muodostamisella. Tässä vaiheessa päätetään, millä tavalla ympäristössä lähdetään kulkemaan. Strategian muodostamisen jälkeen tapahtuu itse ympäristön *selaileminen* (browsing). Selailemisen tuloksena pystytään muodostamaan selailusta *sisällöstä* (content) sisäinen malli. Tämän jälkeen pystytään hyödyntämään muodostunutta sisäistä mallia, sekä tarkentamaan ja lisäämään siihen osia suorittamalla myöhempiä selailuja ympäristössä.

Sisäisen mallin rakentumista on kuvattu myös toisella tavalla, jossa mallin rakentamisessa voidaan erottaa kolme eri tekijää: (1) *maamerkit* (landmarks), (2) *reitti* (route) sekä (3) *yleiskatsaus* (survey) [Chen, 1998]. Tässä sisäisen mallin rakentuminen alkaa maamerkkien havainnoimisella ympäristöstä. Tämän jälkeen alkaa reittien hahmottaminen eri kohteiden välillä. Maamerkit toimivat reittitiedon hankinnan tukena auttamalla hahmottamaan eri kohteiden sijainteja toisiinsa nähden. Viimeisessä vaiheessa sisäiseen malliin muodostuu yleiskatsaus ympäristöstä, jolloin ympäristössä liikkuja ei ole enää niin riippuvainen maamerkeistä ja reiteistä vaan voi sisäiseen malliin tukeutuen hahmottaa uusia reittejä kohteiden välillä.

Spencen ja Chenin malleissa sisäisen mallin rakentuminen nähdään hyvin samankaltaisena prosessina. Selailustrategian luominen tapahtuu usein maamerkkeihin tukeutuen ja itse selailutapahtuman edetessä hahmottuvat ympäristössä olevat reitit. Kun navigoinnin viitekehyksen mukaisia läpikäyntejä ympäristön hahmottamiseksi on tehty tarpeeksi monta, muodostuu ympäristöstä yleiskuva.

Sisäisen mallin muodostamista helpottaa huomattavasti, jos omaksuttava asia on jo ennalta ilmaistu sisäisen mallin rakentamisen kannalta edullisella tavalla. Tästä hyvä esimerkki on Harry Beckin suunnittelema Lontoon metrokartta. Metrossa kulkijan kannalta on oleellista ainoastaan se, missä metroasemat sijaitsevat sekä missä eri metrolinjat risteävät. Metrossa kulkijan ei tarvitse välttämättä tietää metron sijaintia kaupungin kartalla tietyissä matkan vaiheissa. On tärkeämpää tietää, mihin asemalle metro seuraavaksi pysähtyy ja mille toiselle linjalle on mahdollista vaihtaa kyseisellä asemalla. Kartta korostaa onnistuneesti ympäristön maamerkkejä (asemat) ja reittejä (metrolinjat) sekä yleiskuvan luomista näyttämällä eri metrolinjojen suhteet toisiinsa. Kartta on siten käyttäjän sisäisen mallin luomisen kannalta mahdollisimman edullinen.

2.2. Hypertekstiympäristö ja käyttäjän sisäinen malli

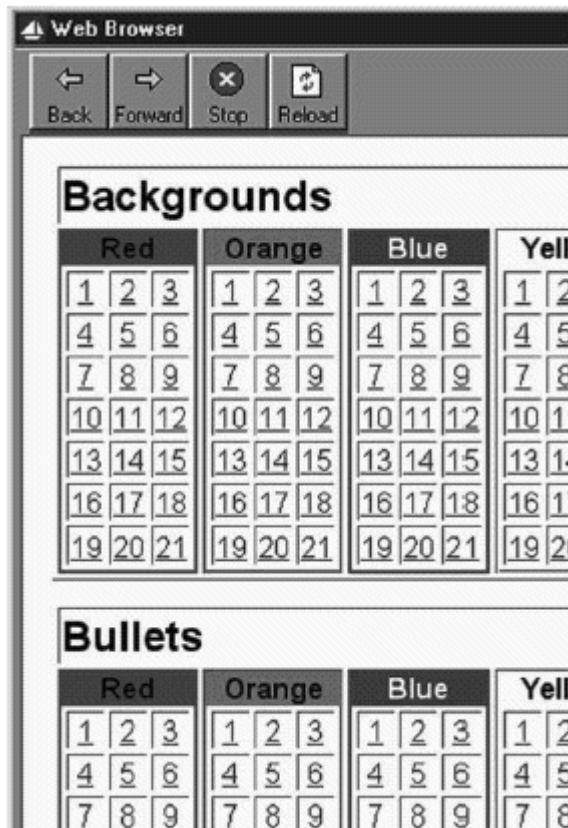
Kuten aiemmin todettiin, on onnistuneen navigoinnin taustalla käyttäjän muodostama oikeanlainen sisäinen malli ympäristöstä, jossa navigoiminen tapahtuu. Käyttäjien sisäisiä malleja hypertekstiympäristöistä on tutkittu suhteellisen vähän. Suurin osa navigoinnin tueksi tuotetuista työkaluista esittävät hypertekstiympäristön siten, että käyttäjän sisäisessä mallissa sen on ajateltu olevan *tilallisesti* (spatial) järjestynyt. Navigoinnin tueksi tuotetuissa työkaluissa hypertekstiympäristön elementit siis esitetään verkon muodossa, jossa eri elementeillä on tilallinen suhde toisiinsa.

Farris et al. [2002] ovat tutkineet käyttäjien sisäisiä malleja hypertekstiympäristöstä. Testissä käytettiin tietokoneella luotuja kuvia, jotka oli järjestetty luokkiin värin ja tyyppin mukaan. Kuvien esittämiseen luottiin neljä erilaista hypertekstisivustoa, joilla jokaisella oli esitettynä 1365 kuvaa. Erona sivustoissa oli niiden pohjalla käytetty hierarkkinen järjestys. Yhden tason hierarkkisessa järjestyksessä kaikki hypertekstisivut olivat yhteydessä toisiinsa. Tässä versiossa käyttäjälle näytettävissä hypertekstisivuissa oli jokaisella viisi taulua, yksi jokaista kuvatyypistä kohden. Lisäksi taulujen sisällä kuvat oli lajiteltu värien mukaan. Jokaista samaa väriä ja tyyppiä olevaa kuvaa oli 21 kappaletta (Kuva 2.2). Kahden tason hierarkiassa jokainen kuva esitettiin omalla sivullaan. Etusivu kahden tason esityksessä oli samanlainen kuin yhden tason esityksessä. Kolmen tason hierarkiassa ensimmäisellä tasolla käyttäjä valitsi kuvan kategorian. Toisella sivulla kuvat näytettiin järjestettyinä tauluihin värien perusteella, joista käyttäjä pääsi tarkastelemaan kutakin kuvaa. Neljän tason hierarkiassa käyttäjä valitsi ylimmällä hierarkian tasolla olevan kuvan kategorian, toisella kuvan värin ja kolmannella kuvan numeron, josta sitten pääsee neljännelle tasolle tarkastelemaan kuvaa.

Testin 40 koehenkilöä jaettiin kymmenen hengen ryhmiin. Kukin ryhmä laitettiin käyttämään aina yhtä versiota hypertekstisivustosta. Käyttäjille annettiin viisi minuuttia aikaa tutkia sivustoa, jonka jälkeen heitä pyydettiin piirtämään kuvio sivuston muodostavien sivujen välisistä suhteista. Riippumatta siitä, mitä hypertekstisivuston versiota käyttäjät olivat käyttäneet, kaikki käyttäjät piirsivät sivuston pohjalta samanlaisen kuvan.

Farris et al. [2002] päättelevät tulosten pohjalta, että käyttäjät eivät muodosta sisäistä mallia hypertekstisivustosta sen sivujen välisten suhteiden pohjalta vaan sivuilla esitettävän sisällön perusteella. Käyttäjien sisäiset mallit hypertekstisivustoista muodostuvat siis sivuilla olevien käsitteiden perusteella, eikä niinkään sivujen välisten linkitysten pohjalta. Testiä voidaan kritisoida siitä, että hypertekstisivuston perustana olevat kuvat ja niiden järjestäminen

käsitteellisesti eri luokkiin on liian keinotekoinen eikä vastaa oikeilta hypertekstisivustoilta löytyviä käsitteitä ja niiden välisiä suhteita.



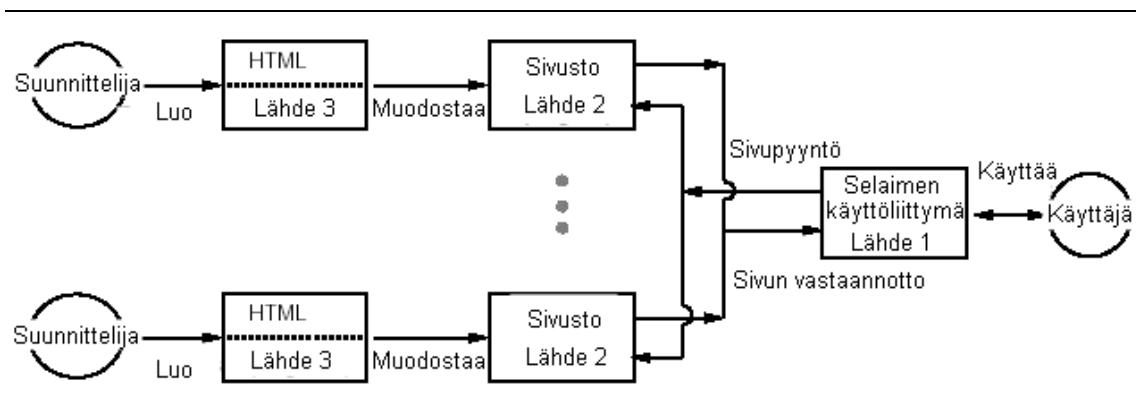
Kuva 2.2. Kuvat järjestettynä yhden hierarkian rakenteeseen [Farris et al. 2002]

Käyttäjien sisäisiä malleja hypertekstisivustoista on tutkinut myös Modjeska ja Marsh [1997], jotka selvittivät käyttäjien navigointia ja myöhempiä mielikuvia eri hypertekstisivustoista. Tutkimuksessa käytettiin neljää erilaista hypertekstisivustoa: kahta laajaa sivustoa, joista toinen oli vahvasti hierarkkinen ja toinen heikosti hierarkkinen, sekä kahta suppeaa sivustoa, jotka myös edustivat kahta hierarkkisuuden ääripäätä. Tutkimuksen tuloksena havaittiin, että hierarkkiset sivustot koetaan pienemmiksi kuin heikosti hierarkkiset sivut. Hierarkkisilla sivustoilla käyttäjät myös vierailivat useammilla sivuilla, kuin heikosti hierarkkisilla sivuilla. Tutkimuksessa kävi myös ilmi, että käyttäjät kokivat sivustot vahvasti hierarkkisiksi riippumatta siitä, oliko niiden taustalla oleva linkkirakenne luonteeltaan vahva hierarkia vai heikko hierarkia. Farrisin et al. [2002] tutkimuksen tulosten kaltaisesti myös tämä tutkimus puhuu sen puolesta, että sisäinen malli sivustosta muodostetaan sivuston sivuilla olevien käsitteiden pohjata eikä niinkään linkkirakenteen pohjalta.

Hypertekstiympäristössä käyttäjän sisäisen mallin muodostuminen on monimutkainen prosessi. Edellä esitetyt tulokset viittaavat siihen, että sisäinen malli ei muodostu sivujen välisten linkkien mukaan vaan käsitteisiin perustuen. Tällä tiedolla on merkitystä siihen, minkälainen kartta hypertekstiympäristöstä kannattaa näyttää käyttäjälle. Sisäisen mallin muodostamiseksi käyttäjä liikkuu ympäristössä. Sisäisen mallin muodostumiseen vaikuttaa osaltaan myös ne keinot, joilla käyttäjä voi ympäristöä tutkia ja liikkua. Seuraavassa kohdassa tarkastellaan käyttäjälle tarjottuja välineitä hypertekstiympäristössä liikkumiseen ja myöhemmin tutkitaan miten käyttäjät hypertekstiympäristössä liikkuvat.

2.3. Navigointiongelma hypertekstiympäristössä

Käyttäjän eksyminen dokumenttien muodostamaan verkkoon johtuu siitä, että käyttäjä pystyy kulkemaan verkossa lukemattomia eri reittejä pitkin. Ympäristö itsessään ei tarjoa navigointia helpottavia maamerkkejä, yksiselitteisiä reittejä tai yleiskuvaa, jotka helpottaisivat sisäisen mallin muodostamista ja siten myös navigointia.



Kuva 2.3. Navigointiongelmien lähteet. [Cockburn and Jones, 1997].

Cockburn ja Jones [1997] ovat jaotelleet navigoinnin ongelmallisuuden lähteet kolmeen eri luokkaan (Kuva 2.3). Ensimmäinen ongelmia tuottava osa-alue on selainten käyttöliittymät. Useat käyttäjät eivät ole tietoisia selainten käyttöliittymien toimintalogiikasta, mikä aiheuttaa ongelmia. Erityisesti eteenpäin- ja taaksepäin-painikkeiden, sekä historia-ominaisuuden käytössä on ilmennyt ongelmia. Toinen ongelmien lähde on hypertekstisivujen suunnittelu. Vaikka käyttäjä osaisikin käyttää selaimen käyttöliittymää sujuvasti, saattaa sivusto olla niin huonosti suunniteltu, että navigointi vaikeutuu [Cockburn and Jones, 1997]. Kolmannen osaongelman tuottaa sivujen kuvailuun käytetty HTML-kuvailukieli, joka on tehtävänsä monessa suhteessa riittämätön

[Cockburn and Jones, 1997]. Seuraavassa kohdassa käydään tarkemmin läpi ensimmäistä osa-aluetta, eli selainten käyttöliittymien tuomia ongelmia.

2.4. Navigoinnin tuki nykyisissä selaimissa

Yleisesti voidaan erotella kahdeksan eri tapaa, joilla käyttäjä pystyy valitsemaan seuraavan sivun selaimessa (Taulukko 2.1). Tutkimuksessaan Byrne et al. [1999] tutkivat selainten käyttöä nauhoittamalla käyttäjien toimia videolle ja analysoimalla näitä myöhemmin. Tutkimuksessa eri toimista luotiin taksonomia, jossa sivujen välillä siirtymiseen löydettiin samat kahdeksan keinoa kuin Cockburnin ja Jonesin [1996] tutkimuksessa.

Numero	Sivun valintametodi
1	Hyperlinkin valitseminen www-sivulta
2	Taakse-painikkeen painaminen
3	Selaimen sisäänrakennetut painikkeet
4	URL-osoitteen kirjoittaminen
5	Eteen-painikkeen painaminen
6	Kirjamerkin valitseminen
7	Selailuhistorian käyttö
8	Sivun uudelleenlataus

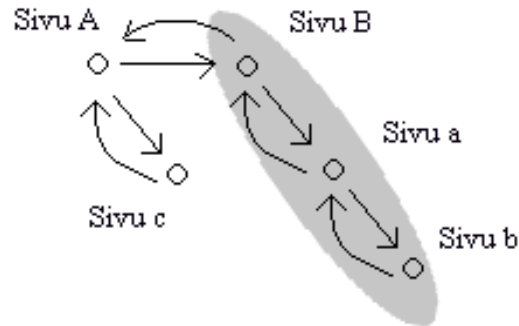
Taulukko 2.1. Nykyiset selainten tarjoamat keinot sivujen välillä siirtymiseen.

[Cocburn and Jones 1996]

Taakse-painike on eniten käytetty selainten tarjoama käyttöliittymä-komponentti. Selailun aikana käyttäjän suorittamista tapahtumista 42% saattaa olla taakse-painikkeen painalluksia [Catledge and Pitkow, 1995]. Taakse-painikkeen käyttö ja hyperlinkkien seuraaminen muodostavat yhteensä 93% kaikista selaimen käyttöliittymän tapahtumista navigoinnin aikana. Selainten käyttöliittymiä voidaan pitää huonosti suunniteltuina, koska suuri osa selainten käyttöliittymän tarjoamista toiminnoista on käyttämättöminä selailu-tapahtuman aikana.

Koska taakse-painiketta käytetään paljon, on sen käyttöä myös tutkittu tarkemmin. Suurimalla osalla selainten käyttäjistä on vääränlainen mielikuva sen toiminnasta [Cockburn and Jones, 1996]. Greenberg ja Cockburn [1999] ovat selvittäneet miten taakse-painike toimii ja miten sen toimintaa voisi parantaa. Sekä Internet Explorerin että Netscape Navigatorin taakse-painikkeet hyödyntävät pino-periaatetta. Tämä tarkoittaa sitä, että selain pitää erillistä pinoa, jossa on lueteltuna aiemmin vierailut sivut. Tämän mallin ongelmaksi

on havaittu se, että pinosta putoaa välillä pois sivuja, joissa käyttäjä on vierailut.



Kuva 2.4. Esimerkki navigoinnista sivujen välillä.

Pinosta putoavat sivut ovat sellaisia sivustohaaroja, joissa käyttäjä käy, palaa sitten takaisin lähtösivulle ja lähtee toiseen haaraan. Tätä navigointia kutsutaan *säteittäisnavigoinniksi* (hub and spoke navigation). Säteittäisnavigoinnista on esimerkki kuvassa 2.4. Kuvassa sivustojen pääsivut ovat merkitty isoilla kirjaimilla ja alisivut pienillä kirjaimilla. Esimerkin tapauksessa käyttäjä aloittaa navigoinnin sivulta A jatkaen edelleen sivulle B ja sieltä sivuille a ja b, palaten takaisin sivulle A, josta jatkaa edelleen sivulle c (A -> B -> a -> b -> a -> B -> A -> c). Tässä tapauksessa käyttäjä ei enää pysty palaamaan sivulle B käyttäen taakse-painiketta, koska reitti B -> a -> b on pudonnut pois pinosta kokonaan.

Greenberg ja Cockburn [1999] ovat ehdottaneet uutta mallia taakse-painikkeen toiminnalle. Painikkeen uudessa toimintamallissa sivuja ei pudotettaisi pois pinosta, vaan kaikki vierailut sivut jäisivät muistiin. Cockburn et al. [2002] ovat ehdottaneet, että tämä uusi malli ei kuitenkaan johda parantuneeseen käyttöliittymään. Kummallakin mallilla testihenkilöt suoriutuivat suurin piirtein samassa ajassa ja samalla toimintojen määrällä koetehtävistä.

Tauscher ja Greenberg [1997] ovat saaneet hyviä tuloksia testeissään uudelleen suunnitelluilla historia-listoilla. Pinoon perustuvan taakse-painikkeen sisältämä lista viimeksi vierailuista sivuista sisältää parhaimmillaan 40% todennäköisyydellä seuraavan vierailtavan sivun. Jos kaikki sivut säilytetään listassa siten, että kahteen kertaan vierailuista sivuista säilytetään listassa vain viimeisin vierailu, päästään yli 50% todennäköisyyteen.

Taakse-painikkeen käyttö on yksinkertaista ja useasti tapahtuvaa nykyisissä selaimissa. Sen käytössä on myös ongelmansa. Tästä esimerkkinä on edellä

mainittu käyttäjien vääränlainen sisäinen malli sen toiminnasta. Yksi mahdollisuus parantaa selainten tarjoamien työkalujen käytettävyyttä on tutkia, minkälaista käyttöä niiden tulisi tukea. Seuraavassa kohdassa käydään läpi navigointimalleja hypertekstirakenteessa. Navigointimallien tutkiminen on hyödyllistä siinä mielessä, että ne antavat viitteitä selaimilta vaadittavista ominaisuuksista.

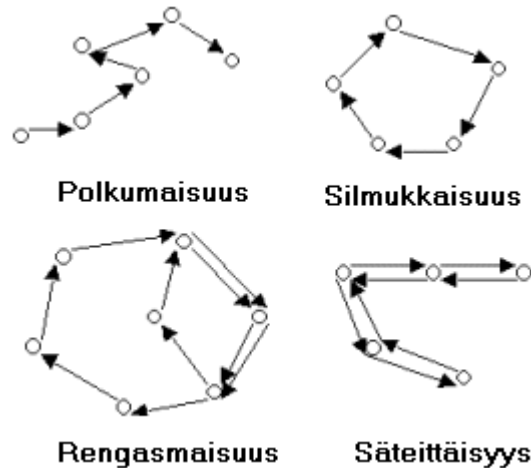
2.5. Havaitut navigointimallit hypertekstirakenteessa

Käyttäjien navigointimalleja ja navigointikäyttäytymistä on tutkittu hypertekstin alkuajoista lähtien paljon. Suuri osa alkuperäisistä tutkimuksista on vielä tänäkin päivänä varteenotettavia, vaikka hypertekstin selailuun tarkoitettut työkalut ovat kehittyneet samoin kuin myös HTML-kieli. Kehityksestä huolimatta monet käyttäjien selailukaavat ja käyttäytymismallit hypertekstiympäristössä ovat säilyneet muuttumattomina.

Käyttäjien navigointimallien tutkimus on tapahtunut pääasiassa kahdella eri tavalla [Byrne et al., 1999]. Käyttäjien navigointia on havainnoitu tutkimalla selainten tuottamia lokeja, joista selviää heidän eri selaimella suorittamat toiminnot ja niiden ajankohdat. Toisaalta on pyritty tutkimaan korkeammalla tasolla niitä tehtäviä ja päämääriä, joita käyttäjillä on hypertekstirakenteessa kulkiessaan. Päämäärien selvittämiseen on käytetty joko kyselylomakkeita tai kontrolloitua koetilannetta, jossa tutkija on tehnyt havainnot käyttäjien toimista [Cockburn ja McKenzie 2000]. Uudemmissa tutkimuksissa on painotettu enemmän jälkimmäistä tutkimusmetodia ja siirretty tutkimuksen painopistettä siten, että on pyritty löytämään ja ymmärtämään käyttäjien muodostamia sisäisiä malleja selainten toimintalogiikasta sekä hypertekstirakenteista.

Mullier et al. [2002] ovat tutkineet käyttäjien selailureittejä hypertekstisivustoilla. Selailureittien tutkimusta motivoi ajatus siitä, että mikäli pystytään löytämään yhteys selailumallien muodon ja selaajan käsillä olevan tehtävän välillä, voidaan käyttäjää auttaa selailutapahtumassa. Selailumallien selvittämiseen Mullier et al. [2002] ovat kehittäneet ohjelman nimeltä Hypernet.

Navigointimallien taustalta on löydetty neljä erilaista peruskuviota (Kuva 2.5): (1) polkumaisuus, (2) rengasmaisuus, (3) silmukkamaisuus sekä (4) säteittäisyys. Polulle ominaista on se, että kukin hypertekstisivu tulee käytyä läpi vain kerran. Rengasmaisessa rakenteessa sivuja käydään läpi siten, että ne muodostavat rengasmaisia rakenteita, jotka voivat olla myös sisäkkäin. Silmukka muodostaa renkaan, joka ei sisällä sisäkkäisiä renkaita. Neljäntenä kuviona on säteittäisesti tapahtuva navigointi.



Kuva 2.5. Peruskuviot navigointimallien taustalla.

Näistä jo aiemmin mainittu säteittäinen navigointi on havaittu myös muissa tutkimuksissa yleiseksi [Catledge and Pitkow, 1995]. Säteittäisnavigoinnin yleisyys on seurausta siitä, että nykyisten selainten taakse-painikkeeseen perustuva navigointi on usein juuri säteittäistä. Tällä menetelmällä selailemalla käyttäjä löytää eräänlaisen keskuksen eli sivun, josta on mahdollista lähteä kulkemaan moneen eri suuntaan. Jos kiinnostavia vaihtoehtoja on useita, voi käyttäjä käydä säteitä luoden tarkastamassa ne kaikki, joiden välissä hän aina palaa takaisin keskuksen lähteäkseen tarkastamaan vielä käymättömiä reittejä. Erityisesti kokemattomat käyttäjät käyttävät navigointimallinaan useammin säteittäisnavigointia kuin kokeneemmat käyttäjät [Mullier et al., 2002]. Kokeneemmilla käyttäjillä yleisempi navigointimalli on rengasmaiset kuviot.

Mullier et al. [2002] ovat löytäneet myös tehtävien suorittamisen taustalta erilaisia selailurakenteita, jotka jakautuvat viiteen osaan: (1) *skannaus* (scanning), (2) *selailu* (browsing), (3) *hakeminen* (searching), (4) *tutkiminen* (exploring) sekä (5) *vaeltelu* (wandering). Skannauksessa käyttäjä navigoi säteittäisesti ja käyttää myös pieniä silmukoita. Tällaisen selailun tuloksena hän saa yleiskuvan selaamastaan alueesta. Yleiselle selailemiselle on puolestaan tyypillistä pitkät silmukat ja isot renkaat. Haettaessa tietoa rakenteeksi muodostuu säteittäisyys sekä pienet silmukat. Tutkiminen tuottaa pitkiä polkuja ja vaeltelu keskikokoisia renkaita.

Tulevaisuudessa selaimet voisivat tukea navigointia paremmin ottamalla nämä erilaiset navigointimallit huomioon. On kuitenkin huomattava, että nämä navigointimallit ovat osaltaan seurausta nykyisten selainten tarjoamista käyttöliittymistä.

2.6. Navigointimallit ajallisesti tarkasteltuina

Navigointimallien hahmottamisen lisäksi navigointia voidaan tarkastella myös ajallisesti. Ajallinen tarkastelu auttaa ymmärtämään, kuinka navigointimallit suhtautuvat aikaan ja kuinka ne toistuvat navigoinnin edetessä [Catledge ja Pitkow, 1995].

Tauscher ja Greenberg [1997] tutkivat 23 käyttäjän selailutapahtumia kuuden viikon ajalta. Tutkimuksessa havaittiin, että 58% käyttäjien vierailemista sivuista oli uudelleenvierailuja. Kunkin käyttäjän vierailmien URL-osoitteiden määrän havaittiin nousevan tasaisesti ajan kuluessa. Cockburnin ja McKenzien [2000] vastaavassa tutkimuksessa analysoitiin seitsemäntoista käyttäjän lokitietoja neljän kuukauden ajalta. Analysoinnin tuloksena havaittiin, että 81% sivuilla käynneistä oli uudelleenvierailuja.

Tauscher ja Greenberg tutkivat myös tarkemmin uudelleenvierailuiden laatua. Tärkein tulos näissä oli se, että seuraavan vierailtavan sivun todennäköisyys löytyä juuri aiemmin vierailtujen sivujen joukosta on suuri. Esimerkiksi kuuden viimeksi vierailun sivun joukosta löytyy seuraavaksi vierailtava sivu 39% todennäköisyydellä. Seuraavan sivun löytymisen todennäköisyys viimeisen viidentoista vieraillun sivun joukosta on 46%.

Tulosten perusteella tutkimuksessa tutkittiin myös historia-listojen ja uudelleenvierailuiden vuorovaikutusta. Yhteensä tutkimuksessa testattiin kahdeksaa eri tavalla toimivaa historia-listaa. Yleisenä tuloksena oli, että 40-50% todennäköisyydellä seuraavaksi vierailtava sivu löytyy viimeisen kymmenen vieraillun sivun joukosta. Parhaimmillaan todennäköisyys löytää seuraavaksi vierailtava sivu viidentoista viimeksi vieraillun sivun joukosta oli 55%. Tulosten perusteella on selvää, että ajallisesti tarkasteltuna navigointi on toistuvaa. Käyttäjän navigointihistorian avulla pystytään suhteellisen suurella todennäköisyydellä ennustamaan käyttäjän seuraavat navigointitapahtumat. Ottamalla huomioon käyttäjien navigointimallit ja sivujen ajallinen järjestäytyminen selailuhistoriassa voidaan käyttäjien liikkumista hypertekstisivuilla pitää kaavamaisena ja samaa kuviota toistavana tapahtumana.

Koska uudelleenvierailujen määrä sivuilla on melko suuri ja navigointitapahtuman luonne toistuva, ei navigointihistoriaa ylläpitävää työkalua välttämättä voida pitää navigointihistoriana sanan varsinaisessa merkityksessä. Useat tällaisella listalla olevat sivut tulevat käyttäjän eteen myös myöhemmin navigoinnin aikana ja saattavat kertautua useaan kertaan.

3. Tiedon visualisointi

Tässä luvussa käydään läpi tiedon visualisointi yksityiskohtaisesti, koska suurin osa navigoinnin tukemiseen tuotetuista sovelluksista perustuu tiedon visualisointiin. Myös tämän tutkielman osana toteutetussa sovelluksessa tiedon visualisoinnilla on keskeinen osa. Ensin määritellään tiedon visualisointi tieteenalana ja selvitetään sen sisältö. Tämän jälkeen visualisoinnin luominen esitetään vaiheittaisena prosessina. Visualisoinnin tarkastelu prosessina auttaa myös ymmärtämään paremmin, mistä visualisoinnissa on kysymys. Myöhemmin tiedon visualisointiin liittyviä asioita käydään läpi tämän prosessimallin avulla. Viimeisenä selvitetään WWW-ympäristön visualisoinneille asettamia rajoituksia.

Tiedon visualisoinnin kannalta on tärkeää tehdä selkeä ero datan ja informaation välillä. Data on jotain ilmiötä kuvaava kokoelma tietoa. Dataa tutkimalla ihminen voi ymmärtää datan kuvaamaa ilmiötä ja näin saada uutta informaatiota ilmiöstä. Informaatio siis syntyy datan pohjalta eli johdetaan datasta [Spence, 2001, pp. 4-5].

Tiedon visualisoinnin tehtävänä on auttaa ihmisiä näkemään ja löytämään oleellinen informaatio datasta. Ymmärryksen helpottamiseksi data visualisoidaan ihmiselle sellaiseen muotoon, jonka ajatellaan olevan helpommin tulkittavissa kuin data sellaisenaan. Visualisoinnin avulla siis autetaan ihmistä datan ymmärtämisessä. Nykyään ihmisiä ympäröi suuri määrä erilaista dataa ja sen määrä kasvaa jatkuvasti. Datan määrän kasvaminen tekee oleellisen informaation löytämisestä entistä vaikeampaa. Tämän takia tiedon visualisoinnin merkitys kasvaa tulevaisuudessa entisestään.

Tiedon visualisointi täytyy erottaa tieteellisestä visualisoinnista. Tieteellisessä visualisoinnissa pyritään kuvaamaan reaalimaailman ilmiötä mahdollisimman tarkasti. Tiedon visualisointi sen sijaan ei ole tällä tavoin sidottu reaalimaailman ilmiöihin, vaan se keskittyy visualisointien luomiseen abstraktien käsitteiden pohjalta.

Karkealla tasolla visualisoinnit voidaan jakaa kahteen eri luokkaan: (1) tekniikoihin, joiden avulla esitetään datan ominaisuuksia ja arvoja sekä (2) tekniikoihin, joiden avulla ilmaistaan datan rakennetta ja suhteita [Freitas et al., 2002]. Suurin osa tiedon visualisoinnin kehittämistä tekniikoista on keskittynyt jälkimmäiseen luokkaan. Myös tässä tutkielmassa tarkastellaan pääasiassa menetelmiä, joiden avulla datan rakennetta ja suhteita olisi järkevä visualisoida.

Ensimmäiseen luokkaan kuuluvat esimerkiksi tilastotieteessä paljon käytetyt dataa kuvaavat esitykset. Tällaisia esityksiä ovat muun muassa histogrammit, viivadiagrammit sekä muut kahden tai useamman muuttujan

suhdetta kuvaavat esitykset. Datan rakennetta ja suhteita kuvaavat visualisoinnit ovat esimerkiksi ympäristöä kuvaavat kartat ja muut tilallisesti järjestetyt esitykset.

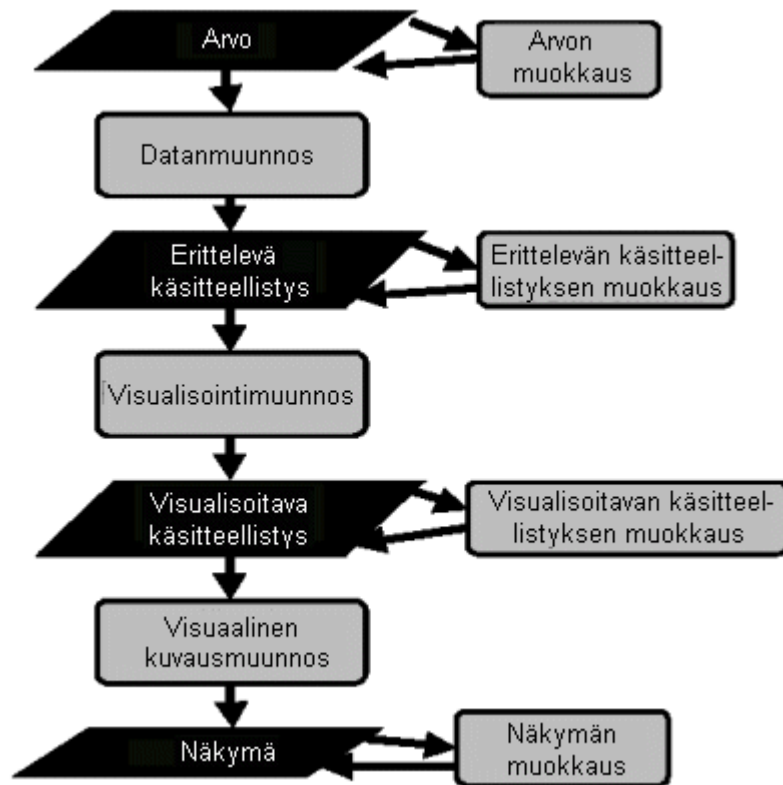
Toiseen luokkaan kuuluvat visualisoinnit ovat olleet perinteisesti yhteen erityiseen tehtävään räätälöityjä, eivätkä niiden toimintaperiaatteet ole useasti olleet vertailukelpoisia. Visualisointien taustalta on kuitenkin pyritty löytämään kaikille visualisoinneille yhteisiä periaatteita eli on pyritty luomaan taksonomia visualisointimenetelmille. Yksi tällainen ehdotettu taksonomia on *datatilojen referenssimalli* (data state reference model) (Kuva 3.1) [Chi and Riedl, 1998]. Seuraavassa kohdassa tarkastellaan tätä mallia. Mallin pohjalta käsitellään myöhemmissä kohdissa myös visualisointien eri osa-alueita.

3.1. Tiedon visualisointi prosessina

Tiedon visualisointi on nähty yleensä prosessina, jossa lähtökohtana on raakadata, josta eri vaiheiden kautta pyritään muodostamaan hyödyllinen visualisointi eli näkymä käyttäjälle. Yleensä visualisointiprosessi on kuvattu neliportaiseksi (Kuva 3.1). Kunkin vaiheen välissä tietoa muunnetaan tietyllä tavalla.

Tilamalleja on kehitelty auttamaan visualisointien suunnittelijoita sekä käyttäjiä hahmottamaan visualisointiprosessi paremmin [Chi and Riedl, 1998]. Mallien avulla visualisoinnit voidaan eritellä selkeästi eri komponentteihin ja vaiheisiin. Tilamalleilla on kaksi erillistä käyttötarkoitusta. Toisaalta ne auttavat visualisointien suunnittelijoita hahmottamaan visualisoinnin tekemisen *linjastona* (pipeline), jonka eri vaiheissa eri ohjelmistokomponentit tekevät tietyt operaatiot syötteenä saamalleen datalle. Toisaalta näillä malleilla voidaan tarkastella visualisointia käyttäjän kannalta. Ollessaan vuorovaikutuksessa visualisoinnin kanssa käyttäjä voi mahdollisesti muuttaa eri vaiheissa olevien komponenttien toimintaa ja näin vaikuttaa itse näytettävään visualisointiin.

Visualisointiprosessin ensimmäisessä vaiheessa tieto on raakadatan muodossa. Ensimmäinen muunnos on *datamuunnos* (data transformation), jonka tuloksena alkuperäiseen dataan on lisätty metadataa. Alkuperäisestä datasta on myös voitu poistaa visualisoinnin kannalta epäoleellista tietoa. Lähteestä riippuen muunnoksen lopputulosta kutsutaan *eritteleväksi käsitteellistykseksi* (analytical abstraction) tai vaihtoehtoisesti *datatauluiksi* (data tables). Tässä muodossa alkuperäinen data on yleensä järjestetty haluttuun tietorakenteeseen.



Kuva 3.1. Datatilojen referenssimalli. [Chi, 2000]

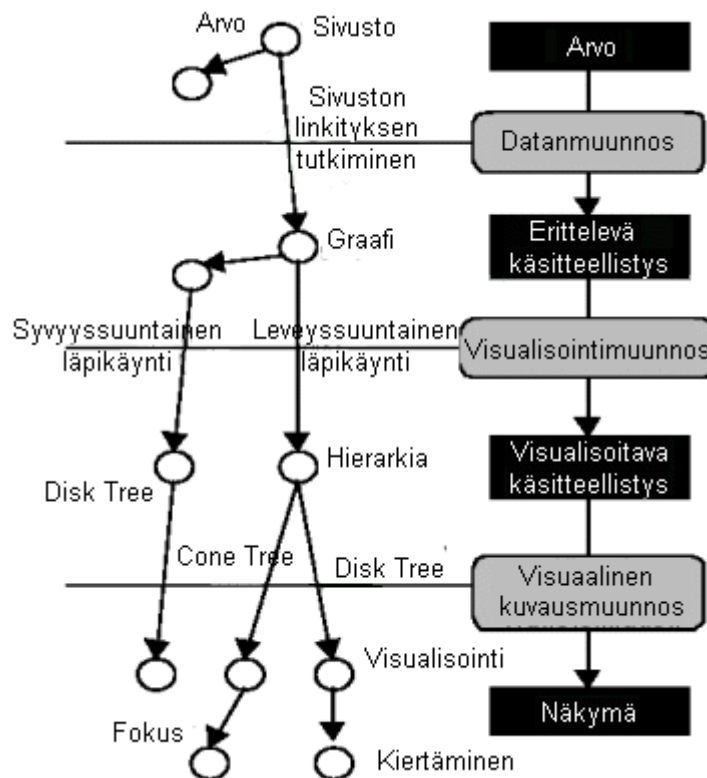
Mallia voidaan soveltaa esimerkiksi visualisoinnin tekemiseen hypertekstisivustoista (Kuva 3.2). Tällöin data koostuu hypertekstisivujen sisällöstä. Datanmuunnoksessa yksittäisten sivujen väliset linkit voitaisiin ottaa huomioon ja järjestää sivut tietorakenteeksi linkkien mukaan. Erittelevä käsitteellisyys olisi tässä tapauksessa graafi sivuston sivuista. Tässä vaiheessa ei oteta kantaa siihen, miten sivut ja niiden väliset suhteet tulisi esittää käyttäjälle vaan pyritään ainoastaan kuvaamaan alkuperäinen data rakenteisesti.

Hypertekstiympäristöstä visualisointia tehdessä päädytään useasti tietorakenteeseen, joka on graafi. Tämä johtuu hypertekstin verkkomaisesta rakenteesta. Myös siinä tapauksessa, että visualisoidaan käyttäjän liikkeitä hypertekstisivuilla, päädytään samanlaiseen tietorakenteeseen. Tässä tapauksessa graafi on hypertekstiympäristön sivuja kuvaavan graafin aligraafi, koska monesti käyttäjä käy vain osalla kaikista hypertekstiympäristön sivuista.

Ensimmäisen muunnoksen jälkeen tehdään *visualisointimuunnos* (visualization transformation), jonka tuloksena syntyy *visualisoitava käsitteellisyys* (visualization abstraction). Tässä vaiheessa alkuperäinen data on järjestetty sellaiseen muotoon, että se pystytään näyttämään käyttäjälle. Visualisoitaessa WWW-sivustoa tässä vaiheessa päätettäisiin, miten edellisen vaiheen tuloksena syntynyttä graafia käydään läpi. tämä synnyttää graafin

solmujen välille tiettyjä suhteita. Graafi voidaan esimerkiksi käydä läpi niin, että solmut muodostavat hierarkkisen järjestyksen. Tämä hierarkkinen rakenne on jo mahdollista piirtää, koska solmujen väliset suhteet toisiinsa ovat tiedossa.

Lopullisen näkymän tuottamiseksi tehdään vielä *visuaalinen kuvausmuunnos* (visual mapping transformation). Kuvausmuunnos kuvaa edellisessä vaiheessa saadun tietorakenteen näytettävälle pinnalle. Tämän prosessin tuloksena syntynyttä kuvaa käyttäjä voi tarkastella ja tulkita.



Kuva 3.2. Datatilojen referenssimalli sovellettuina WWW-sivustoihin [Chi, 2000]

3.2. Visualisoinnin vuorovaikuttisuus

Jotta visualisointi olisi mahdollisimman hyödyllinen käyttäjän kannalta, tarjotaan käyttäjälle yleensä myös mahdollisuus olla vuorovaikutuksessa visualisoinnin kanssa. Vuorovaikuttisuus voi kohdistua mihin tahansa edellä esitetyn datatilojen referenssimallin tiloihin. Keskeinen jako tehdään siinä, kohdistuuko vuorovaikuttisuus visualisoitavaan dataan vai tämän datan perusteella tehtyyn näkymään [Chi and Riedl, 1998]. Ensimmäisessä vaihtoehdossa käyttäjä käsittelee visualisoinnin lähteenä olevaa dataa. Vuorovaikutus voi kohdistua datan referenssimallin kolmeen ensimmäiseen vaiheeseen eli alkuperäiseen dataan, datanmuunnokseen tai erittelevään

käsitteellistykseen. Toinen vaihtoehto on, että käyttäjä on visualisoitavan datan sijasta vuorovaikutuksessa näkymään liittyvien vaiheiden toiminnan kanssa.

Jos käyttäjä on vuorovaikutuksessa alkuperäisen datan kanssa, hän saattaa rajata joitain datan osia kokonaan pois tai muuten käsitellä alkuperäistä dataa. Esimerkiksi hypertekstisivujen perusteella tehtävässä visualisoinnissa käyttäjä voisi vuorovaikutteisesti valita, mitkä hypertekstisivut tulisivat mukaan ja mitkä eivät. Käyttäjän vuorovaikutus voi kohdistua myös datanmuunnokseen. Tällöin käyttäjä vaikuttaa siihen, miten data käsitellään tietorakenteen muotoon. Hypertekstisivut voidaan esimerkiksi järjestää tietorakenteeksi linkkien mukaan tai niiden hakemistojen mukaan, joissa ne sijaitsevat. Erittelevän käsitteellistykseen vaiheeseen kohdistuvan vuorovaikutuksen avulla käyttäjä voi muuttaa datanmuunnoksessa syntynyttä tietorakennetta.

Jos käyttäjän vuorovaikutus kohdistuu näkymän muodostamiseen liittyvien vaiheiden toimintaan eikä dataan, itse näkymän lähteenä oleva data ei muutu, mutta datan esittäminen näkymässä muuttuu. Esimerkiksi hierarkkinen tietorakenne pystytään näyttämään käyttäjälle monella eri tavalla muuttamatta hierarkian lähteenä olevaa dataa. Seuraavassa käydään läpi datatilojen referenssimallin vaiheet, joissa käyttäjä voi muuttaa näkymää visualisoitavaan dataan. Nämä vaiheet ovat visualisaatiomuunnos, visualisoitava käsitteellistys, visuaalinen kuvausmuunnos ja lopullinen näkymä.

Visualisaatiomuunnokseen kohdistuvalla vuorovaikutuksella voidaan vaikuttaa menetelmään, jonka avulla erittelevä käsitteellistys muunnetaan visualisoitavaksi käsitteellistykseksi. Käyttäjä voi esimerkiksi valita menetelmän, jolla tietorakenne käydään läpi visualisoitavan tietorakenteen luomiseksi [Chi and Riedl, 1998]. Visualisoitavaan käsitteellistykseen kohdistuvan vuorovaikutuksen avulla käyttäjä pystyy muokkaamaan näytettävää tietorakennetta. Jos visualisoitava tietorakenne on esimerkiksi puu, voi käyttäjä tässä vaiheessa vaikuttaa muun muassa siihen, kuinka monta puun hierarkian tasoa näytetään [Chi and Riedl, 1998].

Viimeisessä muunnoksessa eli visuaalisessa kuvausmuunnoksessa visualisoitava käsitteellistys kuvataan lopulliseen näkymään. Tässä vaiheessa käyttäjä voi esimerkiksi valita sen tekniikan, jonka mukaan muunnos tehdään. Esimerkiksi puu voidaan piirtää näkymään monella eri tavalla. Käyttäjä voisi esimerkiksi päättää, piirretäänkö puu siten, että sen juuri on näkymän ylälaidassa vai alalaidassa. Lopullisessa visualisoinnissa käyttäjä pystyy vaikuttamaan näytettävän kuvan piirteisiin. Tässä käyttäjä voisi esimerkiksi järjestellä graafin solmuja näkymässä ja valita miten graafin solmut näytetään. Mitä lähemmäs mallissa mennään alkuperäistä raakadatan vaihetta, sitä vaikeammaksi vuorovaikutteisuuden toteuttaminen tulee [Chi and Riedl, 1998].

Helpointa vuorovaikutteisuuden toteuttaminen on viimeisessä vaiheessa, jolloin käyttäjä muuttaa vain jotain näkymään liittyvää parametria.

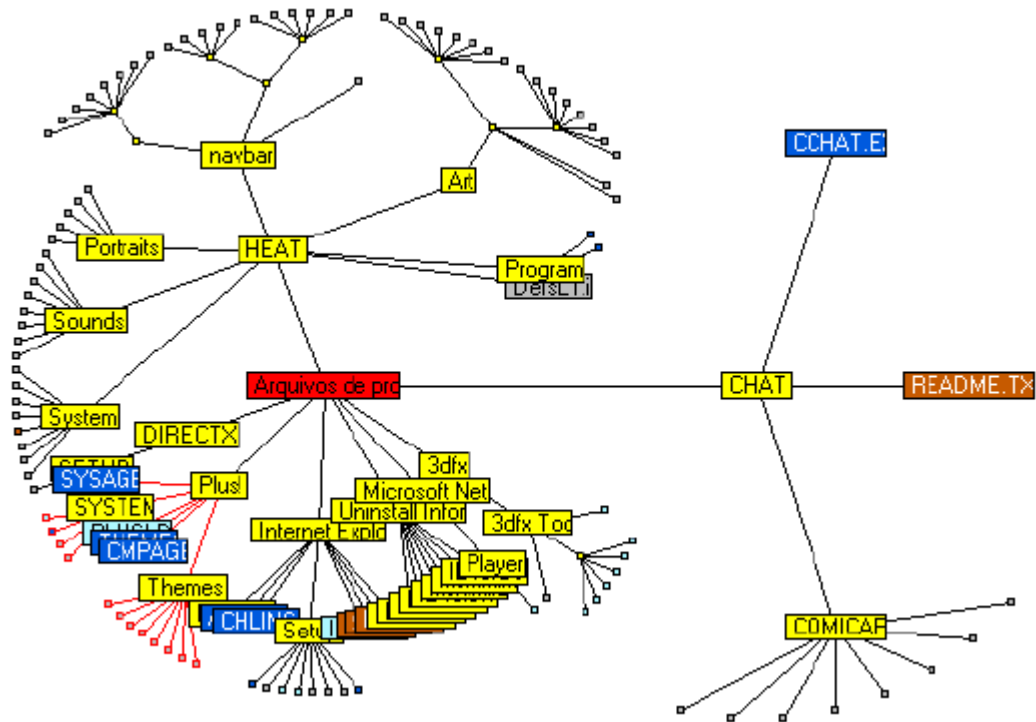
Visualisoinnin vuorovaikutteisuus ja tiedon mahtuminen visualisointiin on läheisessä suhteessa toisiinsa. Vuorovaikutteisen visualisoinnin avulla on mahdollista näyttää enemmän tietoa. Seuraavassa kohdassa käydään läpi joitain menetelmiä, joiden avulla visualisointiin saadaan mahtumaan enemmän tietoa.

3.3. Tiedon mahtuminen visualisointiin

Visualisoinnit, jotka ilmaisevat visualisoitavan datan rakennetta ja suhteita vaativat paljon näyttötilaa. Tätä ongelmaa on pohdittu pitkään ja erilaisia ehdotuksia ratkaisuksi on useita. Mikään ratkaisu ei kuitenkaan ole osoittautunut täydelliseksi. Kaikissa vaihtoehtoissa joudutaan joustamaan jonkun tekijän suhteen.

Spence [2001, pp. 116-133] jakaa tilaongelman ratkaisuun esitetyt tekniikat kuuteen eri luokkaan: (1) *focus+context*, (2) *piilottaminen* (suppression), (3) *magic lenses*, (4) *selailu* (browsing), (5) *effective view*, sekä (6) *zoomaus ja panorointi* (zoom and pan). Jokainen näistä kuudesta tekniikasta tarjoaa oman ratkaisun siihen, miten mahdollisimman suuri määrä tietoa saataisiin esitettyä käyttäjälle mahdollisimman tehokkaasti. Seuraavassa käydään tarkemmin läpi näitä kuutta eri menetelmää tarkastelemalla niitä rinnan puurakenteen visualisointiin käytetyn *kaksitaittoisen puun* (bifocal tree) kanssa (Kuva 3.3).

Focus+context -tekniikka ratkaisee esittämiseen liittyvän ongelman siten, että tarkasteltavasta ympäristöstä näytetään käyttäjälle tarkennettuna vain osa. Tästä on se hyöty, että käyttäjä pystyy tarkastelemaan esitystä yksityiskohtaisesti ilman, että se vaatisi kohtuuttoman suuren näyttöpinnan. Ne alueet, joista käyttäjä on kiinnostunut, näytetään fokusalueella ja ympäröivät alueet pienennettynä tämän ympärillä. Focus+context -tekniikka on yksi esimerkki yleisesti käytetystä *vääristämisestä* (distortion). Siinä näkymää vääristetään siten, että tärkeät osat pysyvät selkeästi näkyvillä ja epäoleelliset asiat näytetään pienemmällä tarkkuudella.



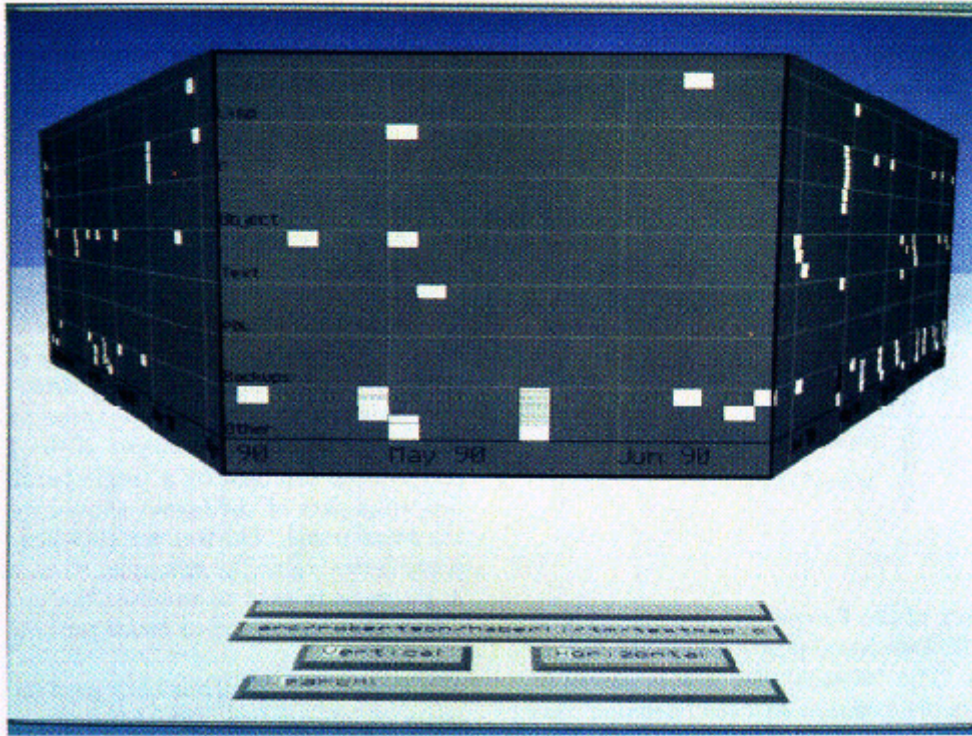
Kuva 3.3. Puurakenteen kaksitaittoinen visualisointi [Cava et al., 2002]

Perspektiivinen seinä (perspective wall) [Mackinlay et al. 1991] on yksi esimerkki focus+context -tekniikan käytöstä (Kuva 3.4). Menetelmän avulla pystytään tehokkaasti esittämään lineaariseen järjestykseen asetettua lähdedataa. Käyttäjä pystyy tarkastelemaan koko lähdedataa liikuttelemalla fokuksen sijaintia seinällä. Vaikka käyttäjä näkeekin hetkellisesti tarkennettuna vain osan datasta, pystyy hän helposti muodostamaan kokonaiskuvan koko datasta.

Yleiskuva ja tarkennus (overview+detail) on focus+context -tekniikan lähisukulainen. Nämä kaksi eroavat siinä, että fokus ja konteksti näytetään samassa näkymässä, kun yleiskuva ja tarkennus ovat kaksi eri näkymää. Yleiskuvassa näytetään ympäristö siten, että käyttäjän on mahdollista suhteuttaa kulloinkin sijaintinsa muun ympäristön suhteen. Tarkennuksessa näytetään ympäristö tarkennettuna sijainnin mukaan. Käyttäessään yleiskuva ja tarkennus -tekniikalla tehtyä esitystä käyttäjän siis täytyy vaihtaa kahden eri näkymän välillä. Focus+context sen sijaan näytetään samassa näkymässä, jolloin käyttäjä liikuttelee fokusaluetta nähdäkseen tarkemmin haluamansa alueet.

Myös puurakenteen kaksitaittoinen visualisointi käyttää focus+context -tekniikkaa ja on osaltaan myös yleiskuva ja tarkennus -tekniikan sovellutus. Tässä fokuksia on yhden sijasta kaksi, mutta periaate on sama kuin yhtä

fokusta käytettäessä. Voidaan ajatella, että vasemmanpuoleinen fokus näyttää ympäristön laajemmin eli toimii yleiskuvana ja oikeanpuoleinen näyttää tarkennuksen. Yleensä kuitenkin yleiskuva ja tarkennus -tekniikkaan perustuvissa toteutuksissa yleiskuva on erillinen kokonaisuus tarkennuksesta, mikä ei edellä mainitussa kaksitaittoisessa visualisoinnissa toteudu.



Kuva 3.4. Perspektiivinen seinä. [Mackinlay et al., 1991]

Piilottamisessa käyttäjän työtä helpotetaan esittämällä vain hänen kannaltaan oleelliset asiat ympäristöstä. Esimerkiksi kartasta voidaan esittää käyttäjälle eri tarkkuudella eri osat sen mukaan, mikä on käyttäjän kannalta hyödyllistä. Ongelmana tässä tekniikassa on se, miten määritellään käyttäjän kannalta oleellinen tieto kullakin hetkellä. Kaksitaittoisen puun kohdalla piilottaminen tapahtuu laskemalla kunkin solmun suhde tarkastelun alla olevaan solmuun. Näin saadaan luku, joka ilmoittaa käyttäjän *kiinnostuksen asteen* (degree of interest (DOI)) kyseiseen solmuun. Jos aste on pieni, näytetään solmusta vähemmän tietoa kuin suuremman asteen omaavista solmuista.

Magic lenses -tekniikan lähestymistapa on sikäli erilainen, että käyttäjä ei tarkastele visualisointia vaan tavallista dataa, esimerkiksi tekstidokumenttia. Käyttäjä voi ottaa menetelmän eri hetkillä käyttöön, jolloin se voisi näyttää tarkoitukseen sopivaa tietoa käyttäjälle. Tekstidokumentissa tämä voisi tarkoittaa vaikkapa sanan merkityksen esittämistä [Spence, 2001, pp. 124-125].

Kaksitaittoisen puun kohdalla tämän tekniikan sovellutus voisi olla lisäinformaation näyttäminen piilotetusta solmusta taikalinsillä solmuun osoitettaessa.

Selailu on myös hyödyllinen tapa esittää käyttäjälle tietoa ympäristöstä. Se on hyödyllinen siinä tapauksessa että käyttäjä haluaa muodostaa kokonaiskuvaa ympäristöstään, kiinnittämättä liikaa huomiota yksityiskohtiin. Data voidaan esittää käyttäjälle monella eri tavalla siten, että se täyttää selailemisen vaatimukset. Käyttäjälle voidaan tarjota mahdollisuus itse selata aineistoa läpi, tai aineisto voidaan näyttää käyttäjälle tietyssä järjestyksessä automaattisesti. Selailun jälkeen käyttäjällä on mielikuva siitä, mitä aineisto pitää sisällään. Kaksitaittoiseen puuhun sovellettuna selailu voisi tarkoittaa sitä, että käyttäjälle näytetään pikaisesti jonkun puun haaran sisältämät solmut. Käyttäjän ei tarvitsisi navigoida puun haarassa vaan hän saisi kuvan haarassa olevista solmuista pikaisen selailun ansiosta.

Näkymässä liikkumisen sujuvuuteen on mietitty ratkaisua effective view -tekniikassa. Keskeisenä tutkimusongelmana on se, miten näkymä tulisi järjestää, että se olisi liikkumisen suhteen mahdollisimman tehokas. Pienellä näyttöalalla voidaan esimerkiksi saada esitettyä paljon tietoa, jos näkymä on järjestetty siten, että käyttäjä pystyy liikkumaan siinä sujuvasti ja löytämään tarvitsemansa kohteet ongelmitta. Tässäkin tekniikassa toteutus on riippuvainen esitettävästä lähdedatasta.

Kuudes ja viimeinen tekniikka on zoomaus ja panorointi. Suurta kuvaa voidaan tarkastella lähemmin zoomaamalla ja kulkea alueella liikuttelemalla zoomausta (panorointi). Näillä kahdella menetelmällä pystytään tarkastelemaan isoakin aluetta tarkasti pienellä näyttöalueella.

Effective view ja zoomaus eivät ole sinällään sovellettavissa kaksitaittoiseen puuhun, koska focus+context -tekniikan käyttö takaa sen, että käyttäjälle on kullakin hetkellä tarjolla tarpeeksi informaatiota ympäröivästä ympäristöstä. Effective view ja zoomaus ovat varteenotettavia vaihtoehtoja silloin, kun näyttöpinta-ala on pieni eikä näytettäviä kohteita voi vääristää, ts. näyttää pienemmällä tarkkuudella. Esimerkki tällaisesta tapauksesta on kohteiden esittäminen visualisoinnissa tekstinä.

3.4. Tiedon esittäminen ja käytettävyys visualisoinneissa

Visualisoitavan datan käsittely visualisoitaviksi rakenteiksi ja tämän rakenteen piirtäminen näkymään olivat edellisten kohtien aiheita. Tässä kohdassa esitetään eri visualisointivaihtoehtoja kohteiden esittämiseksi käyttäjälle etenkin visualisoinnin käytettävyyden kannalta. Kohteiden esittämiseen visualisoinneissa ei ole olemassa mitään erityistä teoriaa tai viitekehystä. Eri

esitystapoja käytetään visualisoinneissa sen mukaan, miten onnistuneesti niiden on havaittu toimivan eri yhteyksissä.

Kohteiden esittämiseksi visualisoinneissa käytetään yleensä hyväksi ihmisten näköaistin hyvää erottelukykä tiettyjen kohteiden ominaisuuksien kohdalla. Ihmiset pystyvät helposti erottamaan kokoeroja, pituuseroja ja värieroja. Erojen avulla visualisoinnissa voidaan erottaa kohteita toisistaan ja ilmentää niiden ominaisuuksia. Ihmiset ovat myös hyviä tunnistamaan kasvoja, ikoneita ja kuvioita, sekä hahmottamaan tilallisesti järjestettyjen kohteiden suhteita toisiinsa [Spence, 2001].

Vaikka mitään yleistä ohjetta ei olekaan esitystavan valitsemiseksi, auttaa esitystavan valinnassa ja suunnittelussa käytettävyyden arviointiin kehitetyt heuristiikat. Yksi tällainen heuristiikka on Jacob Nielsenin [1994] kehittämä kymmenen kohdan menetelmä heuristisen analyysin tekemiseksi. Soveltamalla näitä heuristiikkoja pystytään suuntaa antavasti määrittelemään esitystapojen ominaisuuksia.

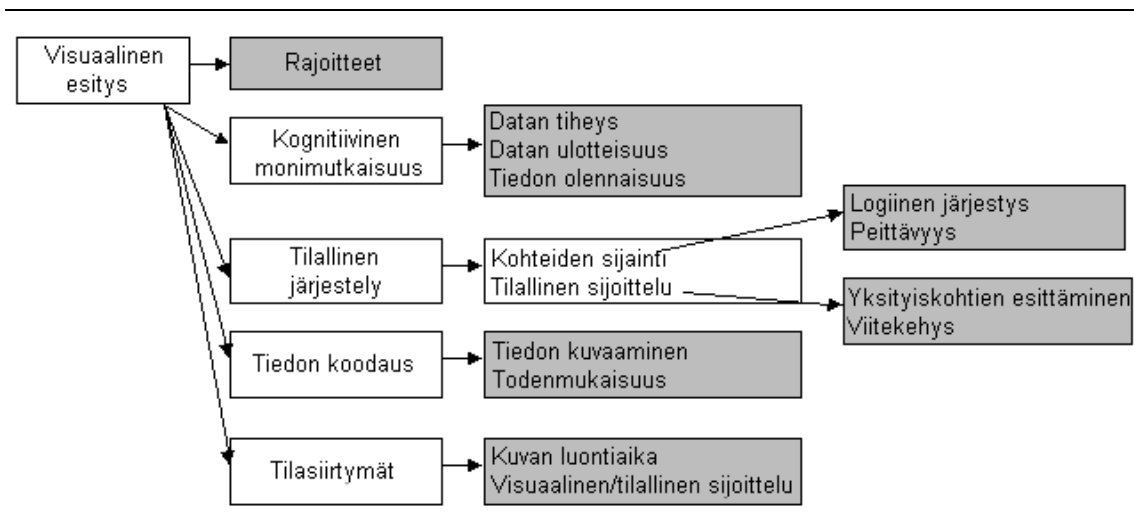
Edellä mainituista kymmenestä heuristiikasta neljä on suoraan sovellettavissa kohteiden esittämiseen visualisoinnissa. Nämä ovat: (1) *esittämisen yhdistäminen reaali maailmaan* (match between system and the real world), (2) *yhdenmukaisuus ja standardinmukaisuus* (consistency and standards), (3) *tunnistaminen muistamisen sijaan* (recognition rather than recall), sekä (4) *esteettinen ja minimalistinen suunnittelu* (aesthetic and minimalist design).

Kohteiden esitys visualisoinnissa tulisi vastata mahdollisimman hyvin reaali maailmaa. Tämä tarkoittaa sitä, että ihmisten valmiita mielikuvia ja käsityksiä reaali maailman kohteista tulisi hyödyntää esitystä suunniteltaessa. Kohteiden tulisi myös olla yhdenmukaisia koko käyttöliittymän suhteen, sekä noudattaa standardin mukaista esitystä, mikäli sellainen on olemassa. Lisäksi kohteen tulisi esittää visualisoinnissa siten, että ne tukevat tunnistamista muistamisen sijaan. Käyttäjän ei siis tarvitsisi muistaa visualisoinnissa olevien esitystapojen merkityksiä. Lisäksi visualisoinnissa tulisi noudattaa esteettistä ja minimalistista esitystapaa kohteita esitettäessä. Käyttöliittymässä kohteet pitäisi esittää ilman suurempia yksityiskohtia, jotka saattavat häiritä käyttäjää tunnistamasta oleellisia asioita.

Freitas et al. [2002] ovat puolestaan luoneet luokituksen, jonka avulla visualisointien käytettävyyttä voidaan arvioida. Luokitus on tarkoitettu erityisesti hierarkisia rakenteita sisältävien visualisointien tarkasteluun. Luokituksen avulla on myös mahdollista erottaa kohteiden esittämislle asetettuja vaatimuksia. Korkeimmalla tasolla visualisoinnin käytettävyyden arviointi jakautuu kolmeen osa-alueeseen: käyttäjälle näytettävän näkymän, visualisoinnin tarjoamien interaktiivisuusmekanismien sekä datan

käytettävyyden arviointiin. Seuraavassa keskitytään näkymän arviointiin käytettyihin kohtiin.

Näkymän arviointi jakautuu luokituksessa viiteen osa-alueeseen. Näkymästä arvioidaan sen *rajoituksia* (limitations), *kognitiivista monimutkaisuutta* (cognitive complexity), *tilallista järjestelyä* (spatial organization), *tiedon koodausta* (information coding) sekä *tilasiirtymiä* (state transitions) (Kuva 3.5). Nämä viisi osa-aluetta sisältävät kukin vielä tarkempia arviointikohteita.



Kuva 3.5. Visualisoinnin arviointiperusteet. [Freitas et al., 2002]

Näistä tiedon koodaus visualisoinnissa vastaa Nielsenin mainitsemaa yhteyttä reaali maailmaan. Visualisoinnin käytettävyyttä voidaan parantaa, mikäli visualisoitava informaatio on esitetty käyttöliittymässä sellaisella tavalla, että käyttäjien on se helppo ymmärtää. Yksi tapa on käyttää suoraa yhteyttä reaali maailman kohteisiin. Lisäksi tiedon koodaus voidaan yhdistää Nielsenin mainitsemaan yhdenmukaiseen ja standardinmukaiseen esitykseen. Standardin mukainen esitys tulee ottaa myös huomioon suunniteltaessa kohteiden tilallista järjestämistä visualisoinnissa. Yleensä esimerkiksi hierarkioiden esittämiseen on olemassa vakiintuneet tavat ja niistä poikkeaminen saattaa harhaanjohtaa käyttäjiä.

Kognitiivinen monimutkaisuus on yhteydessä esityksen esteettiseen ja minimalistiseen suunnitteluun. Käyttöliittymässä tulisi pyrkiä välttämään turhaa monimutkaisuutta ja esittää kohteet siten, ettei niiden tunnistaminen vaadi käyttäjältä ylimääräistä huomiota. Kognitiivinen monimutkaisuus vaikuttaa myös osaltaan siihen, suosiiko käyttöliittymä kohteiden tunnistamista esityksessä vai täytyykö käyttäjän muistaa eri kohteita esittävät symbolit.

Varsinkin vuorovaikutteisissa visualisoinneissa törmätään useasti siihen, että kohteiden sijainnit muuttuvat näkymässä. Tilasiirtymien arvioinnilla pyritään selvittämään, toimiiko käyttöliittymä siten, että käyttäjän on mahdollista säilyttää mielikuva kohteiden liikkeistä näkymässä. Tilasiirtymien suunnittelussa tulee noudattaa yhdenmukaisuutta, jolloin käyttäjä pystyy helpommin ymmärtämään siirtymiä.

3.5. Visualisoinnin näyttäminen hypertekstiympäristössä

Nykyiset hypertekstiympäristöt asettavat visualisoinnin näyttämiseksi paljon rajoitteita. Erityisesti WWW-ympäristö monine kilpailevine tekniikoineen ja sovelluksineen on ongelmallinen sekä visualisoitavan datan että visualisoinnin näyttämisen kannalta.

Tässä tutkielmassa tarkasteltavat visualisoinnit hypertekstirakenteessa navigoimisen helpottamiseksi ovat miltei poikkeuksetta yleiskuva ja tarkennus-menetelmään tukeutuvia. Tarkennuksessa oleva osa on yleensä normaali yksittäinen hypertekstisivu ja yleiskuvassa esitetään itse navigoinnin helpottamiseen tarkoitettu visualisointi. On olemassa myös visualisointeja, joissa käyttäjä ei näe alkuperäisiä hypertekstisivuja navigoidessaan, vaan hänen on ne erikseen avattava suoritettuaan navigoinnin visualisoinnin avulla.

Yleiskuva ja tarkennus -tekniikan käyttäminen on sikäli ongelmallista, että sen käyttämisen on todettu lisäävän jo itsessään jonkin verran käyttäjän kognitiivista kuormitusta, koska käyttäjän pitää tekniikkaa hyödyntäessään siirtää huomiotaan yleiskuvan ja tarkennuksen välillä. Ongelman lievittämiseksi voidaan yleiskuvan ja tarkennuksen toimiminen suunnitella siten, ettei niiden vuorovaikutus ole käyttäjän tehtävänä. Tällaisessa ratkaisussa käyttäjä voi keskittyä kerrallaan toiseen näkymään kiinnittämättä huomiota toiseen.

On luonnollista tehdä tarkennuksen ja yleiskuvan välisestä toiminnasta sellainen, ettei käyttäjän tarvitse seurata kumpaakin yhtä aikaa. On huomattava, että toisin kuin monissa muissa visualisoinneissa käyttäjän vuorovaikutus yleiskuvassa näytettävän visualisoinnin kanssa tulee olla mahdollisimman vähäistä. Tämä poikkeaa perinteisestä mallista, jossa käyttäjän saama hyöty visualisoinnista koetaan lisääntyvän, jos hän saa olla sen kanssa suorassa vuorovaikutuksessa. Useasti navigointia tukeva visualisointi järjestyy käyttäjän navigointitapahtumien mukaisesti eli perinteinen suora vuorovaikutus visualisoinnin kanssa korvataan tässä tapauksessa navigoimalla tarkennuksessa, joka otetaan yleiskuvassa huomioon.

Yksi tärkeimmistä ympäristön asettamista vaatimuksista on visualisoinnin nopeus ja tehokkuus. Esimerkiksi yhdeksi WWW-ympäristön pahimmaksi käytettävyysongelmien lähteeksi on havaittu ympäristön hitaus käyttäjän

kannalta [Nielsen, 1999]. Ympäristössä, jossa nopeus on jo sinällään ongelma, tulee välttää ylimääräisen hitauden aiheuttamista. Visualisoinnin näyttämiseen käytetyissä asiakaskoneissa on yleensä käytettävissä suhteellisen paljon laskentatehoa, joten itse näytettävän visualisoinnin tuottaminen ei ole ongelma. Ongelmaksi muodostuu visualisoitavan datan hankkiminen, mikäli se täytyy siirtää verkon yli tai on muuten hitaan yhteyden päässä.

4. Ratkaisumalleja navigointiongelmaan

Tässä luvussa käydään läpi eri vaihtoehtoja, joiden avulla voidaan ratkaista navigointiongelma tai lievittää sitä. Lähtökohtana on reaali maailman kaltainen ratkaisu, jossa navigoinnin helpottamiseksi on otettu käyttöön karttoja. Kartan luominen hypertextiympäristöstä on visualisointien tuottamista. Toisin, kuin reaali maailmassa kartan pohjaksi on käytettävissä lukematon määrä erilaista informaatiota ympäristöstä. Tämän luvun tarkoituksena on selvittää, mitä informaatiota ympäristö tarjoaa ja miten sen perusteella tulisi luoda edellä mainittu kartta.

Hypertextirakenteessa navigoimisen ongelman nähdään johtuvan siitä, että nykyinen hypertextiympäristö ja sen tarkastelemiseen käytetyt työkalut eivät tarjoa tarpeeksi navigointitukea käyttäjälle. Kuten luvussa 2 mainittiin, ongelman aiheuttajat voidaan eritellä kolmeen kategoriaan: (1) seilailuun käytettyjen työkalujen huono sopivuus tehtäväänsä, (2) hypertextisivujen huono suunnittelu sekä (3) HTML-kielen puutteellisuus [Cockburn and Jones, 1997]. Näistä kolmesta HTML-kieli on kehittynyt eteenpäin jonkin verran ja sen oheen on tullut tukevia tekniikoita. Myös hypertextisivujen suunnitteluun on kiinnitetty enemmän huomiota. Selainten käyttöliittymät ovat kehittyneet vähiten, joten niihin on vastaavasti alettu kiinnittämään entistä enemmän huomiota.

Tässä luvussa keskitytään erityisesti ensimmäiseen kategoriaan liittyviin ongelma-alueisiin, tutkimuksiin ja kehitysehdotuksiin. Kahteen viimeksi mainittuun kategoriaan kuuluvat kehitysehdotukset ovat siitä ongelmallisia, että niiden kulkeutuminen laajaan käyttöön vie huomattavasti aikaa. HTML-kielen ja paremman suunnittelun avulla voidaan osaltaan helpottaa hypertextiympäristössä liikkumista, mutta niiden avulla ei voida täysin ratkaista navigointiongelmaa.

Sovellukset navigoinnin helpottamiseksi voivat olla kahdenlaisia. Keskeinen kahtiajako sovelluksissa tapahtuu sen suhteen, toimivatko ne dynaamisesti vai staattisesti [Cockburn and Jones, 1997]. Tässä jako dynaamisiin ja staattisiin perustuu siihen, missä vaiheessa sovellusta ajetaan suhteessa käyttäjän navigointitapahtumaan. Dynaamiset sovellukset hankkivat tietonsa käyttäjän navigointitapahtuman aikana. Staattisia ohjelmia ajetaan pääasiassa ennen navigointitapahtumaa. Jako staattisiin ja dynaamisiin sovelluksiin voitaisiin tehdä myös käyttöliittymän vuorovaikutteisuuden mukaan.

Staattisesti toimivat sovellukset saavat käyttöönsä enemmän tietoa ympäristöstä kuin dynaamisesti toimivat, koska ne pystyvät käymään läpi

hypertekstiympäristön sivustoja jo ennen kuin käyttäjä navigoi ympäristössä. Yleensä staattiset ohjelmat tuottavat yleiskuvia hypertekstiympäristöstä, kun taas dynaamiset sovellukset pyrkivät keskittymään käyttäjän navigoinnin havainnointiin.

Dynaamisesti toimivat sovellukset ovat siinä mielessä rajoittuneempia, että ne ovat riippuvaisia käyttäjän navigoinnista ja käyttäjän ympärillä olevasta ympäristöstä kullakin hetkellä. Nopeusvaatimukset käyttäjän suunnalta rajoittavat sovelluksen käytettävissä olevaa laskentatehon määrää ja aikaa. Toimiakseen siedettävällä nopeudella dynaaminen sovellus ei voi esimerkiksi tehdä kovin laajaa linkkirakenteeseen perustuvaa analyysiä käyttäjää ympäröivästä hypertekstisivustosta.

Seuraavassa kohdassa tarkastellaan staattisia, yleiskuvan luontiin perustuvia ratkaisuja navigointiongelmaan. Tämän jälkeen tarkastellaan lähemmin dynaamisesti toimivia ratkaisuita, jotka muodostavat valtaosan tässä tutkimuksessa tarkasteltavista ratkaisuksista.

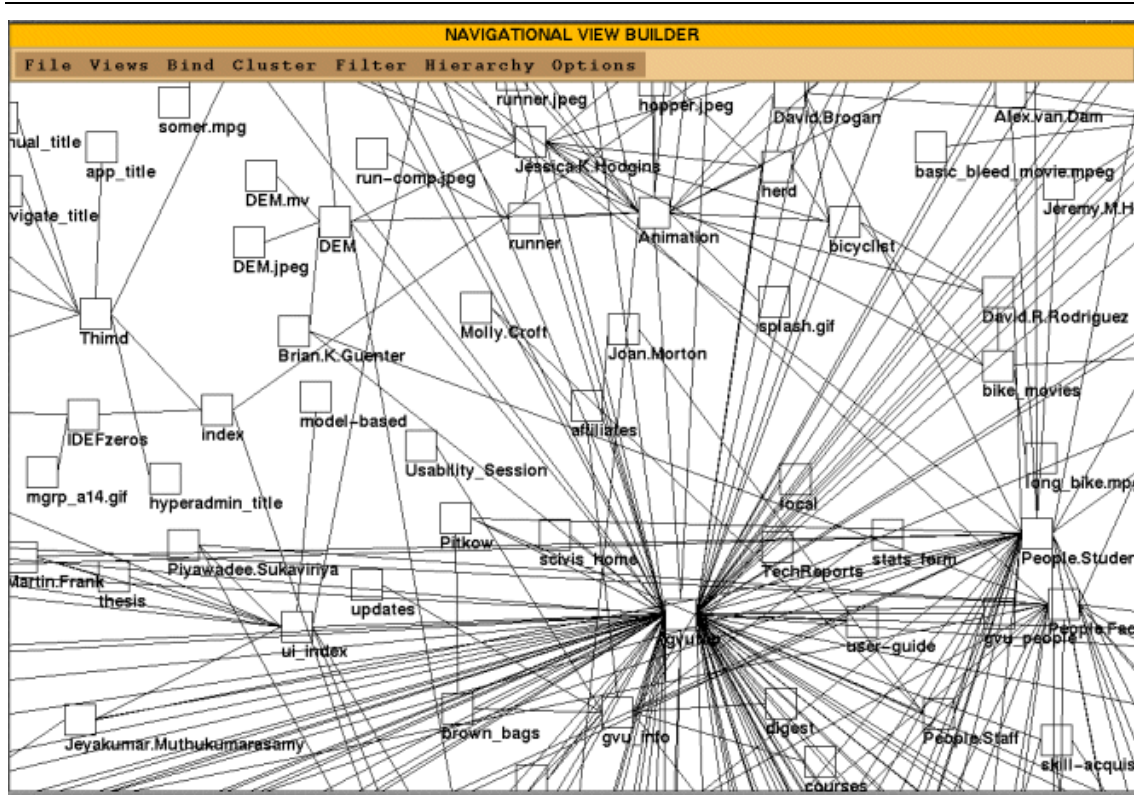
4.1. Yleiskuvaan perustuvat ratkaisut

Kuten toisessa luvussa mainittiin, hyvin suunniteltu yleiskuva ympäristöstä auttaa käyttäjää muodostamaan oikeanlaisen sisäisen mallin siitä. Suuri osa hypertekstiympäristössä navigointia tukemaan suunnitelluista sovelluksista perustuvatkin idealle, että käyttäjälle tulisi näyttää yleiskuva navigoitavasta ympäristöstä.

Ongelma näissä sovelluksissa on, että ne luovat yleiskuvan hypertekstisivustojen linkkirakenteen pohjalta. Toisessa luvussa mainitut tutkimukset käyttäjien sisäisistä malleista hypertekstisivustoista puoltavat sitä, että ei ole järkevää perustaa hypertekstirakenteesta muodostettua yleiskuvaa pelkän linkkirakenteen varaan. Yleiskuva ei tue tällöin oikeanlaisen sisäisen mallin muodostumista.

Modjeskan ja Marshin [1997] mukaan esimerkiksi Mukherjean ja Haran [1997] algoritmi, joka etsii hypertekstirakenteesta maamerkit linkkirakenteen perusteella, ei löydä niitä maamerkkejä, jotka käyttäjät kokevat maamerkeiksi. Tähän vaikuttaa osaltaan myös se, miten hyvin sivuston linkkirakenne vastaa sivuston käsitteellistä rakennetta. Tällä tavoin tuotetut visualisoinnit voivat olla käyttäjän kannalta harhaanjohtavia.

Esimerkiksi kuvasta 4.1 voi selkeästi nähdä sivut, jotka ovat keskussivuja ja yhteydessä moniin muihin sivuihin. Nämä sivut eivät välttämättä kuitenkaan ole käyttäjän kannalta maamerkeiksi luettavia sivuja ja siten oleellisia navigoinnin kannalta.



Kuva 4.1. Jäsentämätön graafi WWW-sivustosta. [Mukherjea and Foley, 1995].

Käsitekartan luonti automaattisesti hypertekstisivuston pohjalta saattaisi olla käyttäjän sisäisen mallin tukemisen kannalta edullisempaa. McDonald ja Stevenson [1998] ovat tutkineet käsitekarttojen, sekä oppimisen välistä suhdetta hypertekstiympäristössä. Tutkimuksessa selvitettiin, miten erityisalaan perehtymättömien ja hyvin perehtyneiden ihmisten oppimiseen vaikutti se, oliko käytettävissä käsitekartta opittavasta alasta.

Tutkimuksen tuloksena alaan perehtymättömät saivat testin lopussa järjestetystä kokeesta yhtä hyvän tuloksen kuin hyvin alaan perehtyneet, jos käytettävissä oli käsitekartta. Sen sijaan pelkkää hypertekstiympäristöä käyttäneet alaan perehtymättömät ihmiset menestyivät kokeessa selkeästi huonommin.

Yleiskuvan näyttäminen käyttäjälle esimerkiksi käsitekartan muodossa ei kuitenkaan aina johda parantuneisiin tuloksiin navigoinnissa. Nilsson ja Mayer [2002] tutkivat, miten yleiskuvan näyttäminen vaikutti navigointiin hierarkkisesti järjestetyllä hypertekstisivustolla. Testi jakautui kahteen osaan: ensimmäisessä vaiheessa koehenkilöt opettelivat sivuston rakennetta ja toisessa vaiheessa he suorittivat varsinaista navigointia mittaavan testin.

Tuloksena havaittiin, että käsitekartan käyttäjät olivat navigoinnissa opetteluvaiheen aikana tehokkaampia, mutta koevaiheessa ilman käsitekarttaa olleet käyttäjät ylsivät yhtä hyvin tuloksiin käsitekartan käyttäjien kanssa.

Nilsson ja Mayer päättävät, että kartan käyttäjät eivät opetteluvaiheessa sisäistä yhtä aktiivisesti hypertekstisivuston rakennetta, koska rakenne on annettu valmiina. Tämän takia kartan käyttäjät saattavat jäädä oppimistuloksissa ilman karttaa navigoivien jälkeen.

Yleiskuvien käytöstä on edellä olevien tutkimusten mukaisesti saatu ristiriitaisia tuloksia. Yleiskuvat saattavat auttaa käyttäjää jossain tilanteissa, mutta eivät aina. Yleiskuvien käytännöllisyyttä heikentää myös se, että ne joudutaan generoimaan kustakin hypertekstisivustosta erikseen. Hypertekstisivut sisältävät lisäksi harvoin järjestelmällisesti ilmaistua metatietoa, joka on edellytyksenä yleiskuvan automaattiselle tuottamiselle. Tämän takia yleiskuvia hedelmällisempiä vaihtoehtoja ovat dynaamisesti luodut visualisoinnit, jotka pystyvät ottamaan myös käyttäjän navigoinnin huomioon.

4.2. Käyttäjän toimet huomioon ottavat ratkaisut

Navigoinnin helpottamiseksi on ensin tunnistettava ne vaiheet, jotka käyttäjä käy läpi liikkuessaan hypertekstiympäristössä. Liikkumisen tutkiminen vaiheittainen auttaa löytämään ne kohdat, joissa käyttäjää voidaan mahdollisesti avustaa.

Head et al. [2000] esittävät viiden kohdan mallin hypertekstiympäristössä liikkumisen vaiheiksi: (1) *tunnistus* (identify), (2) *haku* (search), (3) *selaus* (browse), (4) *järjestäminen* (organize), sekä (5) *uudelleenlöytäminen* (rediscover). Liikkuminen aloitetaan tietotarpeiden tunnistamisella, jonka jälkeen suoritetaan haku sopivien sivustojen löytämiseksi. Tämän jälkeen löydettyjen sivujen sisältöä selaillaan. On myös mahdollista, että haku jätetään kokonaan pois, jolloin tiedon hakeminen tapahtuu pelkästään selailemalla hypertekstisivuja.

Selailun jälkeen mielenkiintoiset sivut järjestetään siten, että ne ovat löydettävissä myöhemmin. Viimeisenä vaiheena on aiemmin löydetyn tiedon uudelleenhaku. Tässä käyttäjä vierailee jo käymillään sivuilla hyödyntäen edellisessä vaiheessa järjestämiään sivuja. Tätä prosessia toistetaan, kunnes vaadittavat tiedot on hankittu.

Hypertekstiympäristössä kulkemiseen liittyy edellisistä kohta 3 eli selailu. Selailutapahtuman edetessä käyttäjän on hyödyllistä tietää: (1) sijainti tällä hetkellä, (2) missä on käynyt aiemmin, sekä (3) missä ei vielä ole käyty. Nämä kolme kohtaa ovat sovellettavissa myös koko navigointitapahtumaan. Tästä esimerkkinä voidaan tutkia kokonaisen selailutapahtuman kulkua. Aloittaessaan selailun käyttäjän päämääränä voi olla esimerkiksi halvan hotellin löytäminen haluamastaan paikasta. Hakemisen ja selaamisen tuloksena käyttäjä löytää kahden hotelleja kyseiseltä alueelta välittävän yrityksen WWW-sivut, jotka hän merkitsee muistiin (järjestäminen) uudelleenkäyntiä varten. Edelleen käyttäjä suorittaa mainitun haun, selauksen ja järjestämisen

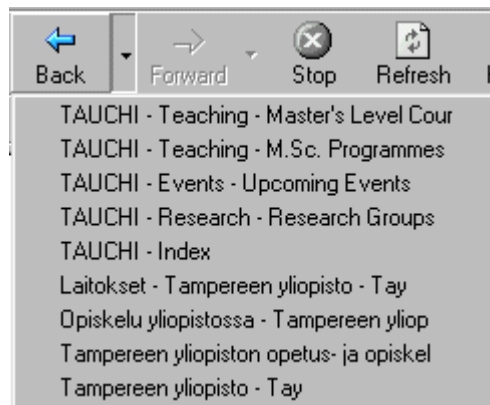
kummallakin näistä kahdesta löytämästään yrityksen WWW-sivuista, löytääkseen lopulta haluamansa hotellin.

Käyttäjän toimia huomioivat ratkaisut pyrkivät auttamaan häntä toistuvan navigointitapahtuman jäsentämisessä ja organisoimisessa. Erilaiset ratkaisut poikkeavat sen suhteen, miten laaja kattavuus niillä on koko selailutapahtumassa. Navigoinnin helpottamiseen tarkoitetun työkalun tulee huomioida kolme viimeistä vaihetta: selailua, järjestämistä sekä uudelleenlöytämistä.

4.3. Sivujen esittäminen navigointityökaluissa

Kaikissa hypertekstiympäristöistä tehdyissä visualisoinneissa on ongelmana sivujen esittäminen. Yhden hypertekstisivun esittämiseen on monta eri vaihtoehtoa, mutta mikään niistä ei ole selvästi paras. Kohteen esittämistä rajoittaa myös yhdelle kohteelle varattu näyttöpinta-ala.

Nykyisin yleisin tapa esittää pienessä tilassa yksi hypertekstisivu on käyttää tekstimuotoista esitystapaa. Yhdestä hypertekstisivusta pystytään erottamaan monta erilaista tekstimuotoista esitystä. Näistä tärkeimmät ovat (1) sivun osoite hypertekstiympäristössä, (2) sivulle annettu otsikko title-tunnisteen avulla, (3) jokin sivun sisältämän tekstielementin käyttö sekä (4) sivun esittäminen kuvakkeena tai muunlaisena graafisena esityksenä [Cockburn and Greenberg, 1999].



Kuva 4.2. Hypertekstisivujen tekstimuotoinen esitys.

Sivun URL-osoitteen käyttäminen sivun tunnisteena on ongelmallista, koska osoite on aloitteleville käyttäjille mitään sanomaton. Kokeneemmille käyttäjille osoite on hyödyksi vain siitä tilanteesta, että se on informatiivinen. Toisin sanoen sivuston luoja on nimennyt sivut siten, että ne ovat käyttäjän kannalta merkityksellisiä. Sivujen osoitteiden on havaittu olevan erityisesti

kokeneempien käyttäjien lisäapu navigoinnissa, mutta sen ei pitäisi olla varsinainen keino sivujen tunnistamiseksi.

Sivujen title-tunnisteen sisällä olevan tekstin näyttäminen on toinen vaihtoehto. Tässä ongelma on sama kuin osoitteen esittämisessä. On täysin sivuston luojan käsissä, onko tunnisteen sisällä oleva teksti sivun tunnistamisen kannalta hyödyllinen (Kuva 4.2). Yhden sivustokokonaisuuden muodostavilla sivuilla saattaa olla kaikilla samanniminen otsikko, tai ei otsikkoa ollenkaan. On myös mahdollista, että otsikon teksti ei vastaa sivua niin, että käyttäjä pystyisi yhdistämään käymänsä sivun ja sen tekstimuotoisen otsikon toisiinsa.

Kolmas vaihtoehto on etsiä sivulta löytyviä tekstejä ja näyttää näistä joku käyttäjälle. Sivun sisältämä ensimmäinen lause tai joku muu sivun sisältämä tekstiosa voidaan esimerkiksi näyttää käyttäjälle. Yksi vaihtoehto on myös näyttää sivulle tuoneen linkin käyttämä teksti linkin kohteesta. Tässä haittapuolena on se, että linkit on yleensä nimetty suhteessa sivujen muuhun sisältöön, mikä rajoittaa linkin tekstin hyödyllisyyttä käyttäjän kannalta itse visualisoinnissa.

Sen lisäksi, että sivuilta on vaikea löytää kuvaavia tekstimuotoisia esityksiä, on ongelmana myös tekstin näyttäminen käyttäjälle. Teksti kaikissa muodoissaan vaatii suhteellisen paljon näyttötilaa, kun otetaan huomioon se, että käyttäjän on pystyttävä lukemaan teksti vaivattomasti.

Tekstimuotoisen esittämisen ongelmallisuuden vuoksi on mietitty vaihtoehtoisia esitystapoja. Yksi näistä on hypertekstisivun esittäminen *kuvakkeena* (thumbnail) [Cockburn et al., 1999; Cockburn and Greenberg, 1999]. Sivun esittäminen kuvakkeena tapahtuu siten, että hypertekstisivun luonnollisen kokoisesta esityksestä tehdään pienempi versio ja käytetään tätä kuvaketta esittämään sivua (Kuva 4.3). Kuvakkeet eivät kuitenkaan voi olla kovin yksityiskohtaisia, mikä johtaa siihen, että useita eri sivuja kuvaavat kuvakkeet voivat näyttää täysin samanlaisilta. Hypertekstisivujen *sommittelu* (layout) on myös usein samanlainen, mikä osaltaan myös tekee kuvakkeista samannäköisiä toisiinsa verrattuna [Cockburn and Greenberg, 1999].



Kuva 4.3. WWW-sivuja esittäviä kuvakkeita. [Cockburn and Greenberg, 1999]

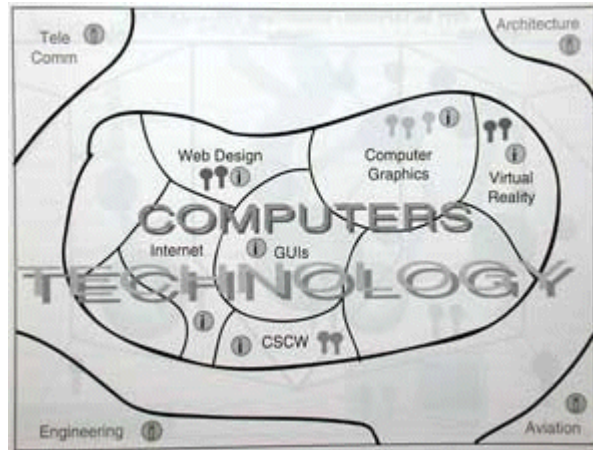
Yksittäin nämä esitystavat voivat johtaa heikkoihin tuloksiin, mutta eri esitystapojen käyttäminen rinnakkain voi olla hyödyllinen vaihtoehto. Esimerkiksi WebView [Cockburn et al., 1999] sovellus tukeutuu tällaiseen hybridiratkaisuun, jossa sivun otsikko ja sivun pohjalta luotu kuvake näytetään rinnakkain.

Hypertekstisivujen ympäristö tarjoaa myös paljon hyödyllisiä informaation lähteitä, joita voidaan hyödyntää visualisoinnissa. Yksi vähemmän käytetty keino on liittää sivua esittävän otsikon tai kuvakkeen oheen myös tieto siitä, minkälainen hypertekstisivu on kyseessä. Onko sivu mahdollisesti sivuston pääsivu vai alasivu? Kuinka monta kertaa käyttäjä on käynyt sivulla verrattuna muihin sivuihin?. Myös nämä tiedot auttavat käyttäjää erottelemaan sivuja toisistaan, vaikka ne muuten olisivat otsikoiltaan ja ulkoasultaan samanlaisia.

4.4. Vaihtoehtoisia esitystapoja

Käyttöliittymän elementtien ja kuvakkeiden suunnittelussa hyödynnetään usein käyttäjien reaali maailmasta saamaa kokemustietoa. Tästä esimerkkinä on yleinen käytäntö esittää käyttöjärjestelmien levyjärjestelmän hakemistot kansio-kuvakkeen avulla. Ratkaisussa hyödynnetään käyttäjien kokemustietoa reaali maailmasta, jossa asiakirjat on useasti järjestetty kansioihin. Käyttöliittymässä voidaan hyödyntää myös laajemmin asioiden ja esineiden järjestystä reaali maailmassa ja luoda vertauskuvallinen yhteys käyttöliittymän komponenttien ja reaali maailman välille.

Esimerkiksi Dieberger ja Frank [1998] esittävät, että monimutkaisten informaatioavaruuksien näyttämiseen käyttäjälle voidaan käyttää kaupunki-metforaa. Tässä mallissa käytetään hyödyksi ihmisten entuudestaan luomaa sisäistä mallia kaupungeista ja niissä sijaitsevista elementeistä. Informaatio-avaruuteen sijoitetut esineet ja niiden väliset suhteet voidaan esittää vertauskuvallisesti kaupungista löytyvillä elementeillä. Tällöin ihmisillä on jo valmiiksi muodostunut kuva siitä, miten ympäristön voi hahmottaa.



Kuva 4.4. Kaupunki-metaforan sovellutus [Dieberger and Frank, 1998].

Esimerkiksi käsittekartan luomisessa voidaan soveltaa kaupunki-metaforaa (Kuva 4.4). Tässä tietojenkäsittelyyn liittyvät käsitteet on sijoitettu saarelle ja näille käsitteille läheiset mutta kaukaisemmat käsitteet saaren ulkopuolelle.

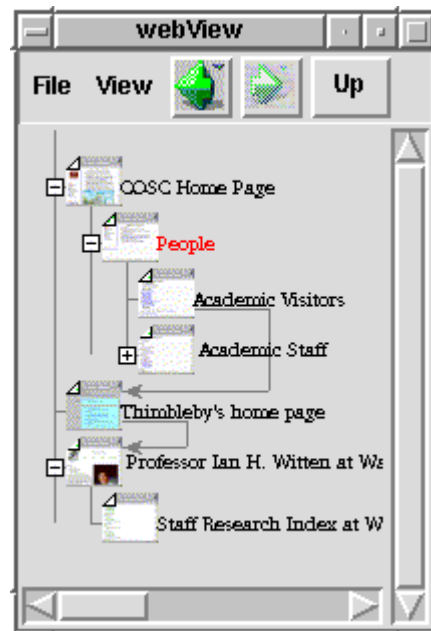
Vertauskuvallisuuden käyttäminen reaali maailman kohteisiin voi jossain tapauksissa myös koitua haittaavaksi tekijäksi. Näin voi käydä etenkin siinä tapauksessa, että reaali maailman elementit eivät toimi käyttöliittymässä samalla tavalla kuin reaali maailmassa. Toisin sanoen niiden toiminta ei herätä oikeanlaista yhteyttä ihmisten mielissä [Preece et al., 2002, pp. 90-91].

Myös hyperteksti ympäristössä voidaan soveltaa samankaltaista lähestymistapaa ja jäsentää esitys ympäristöstä vertauskuvallisesti reaali maailmaan viittaavaksi tai aiemmin kehitettyjen käyttöliittymien elementteihin viittaavaksi. Esimerkiksi hyperteksti ympäristöstä tehtävässä esityksessä voidaan esittää kaupungeina eri sivustokokonaisuudet, taloina keskeiset sivut ja edelleen huoneina yksittäiset sivut. Se, miten konkreettisesti yhtenevyys esitetään, on osin sovelluksesta riippuvainen. Preece et al. [2002, pp. 90-91] pitävätkin tärkeänä, että sopivanlainen ero täysin yhteneväisen ja hieman abstraktisuutta soveltavan esityksen välillä otetaan huomioon.

4.5. Sivujen järjestäminen navigointityökaluissa

Sen lisäksi, että sivut täytyy pystyä näyttämään visualisoinnissa, täytyy visualisoinnissa olla myös joku menetelmä sivujen järjestämiseen näkymässä. Tässä kohdassa keskitytään erityisesti hypertekstisivujen järjestämiseen työkaluissa, jotka näyttävät käyttäjän selailuhistorian. Cockburn ja Greenberg [1999] esittävät yleisellä tasolla neljä eri vaihtoehtoa sivujen järjestämiseen: (1) *säteittäiset dynaamiset puut* (hub-and-spoke dynamic trees), (2) tilallinen järjestäminen, (3) sivustokartat sekä (4) ajallinen järjestäminen.

Ensimmäisessä vaihtoehdossa sivut järjestetään sen mukaan, miten käyttäjä on käynyt ne läpi. Säteittäinen navigointi on yksi yleisimpiä navigointistrategioita hypertekstiympäristössä ja järjestäminen tässä tapauksessa tarkoittaa sivujen järjestämistä tämän läpikäyntimallin mukaan. Luonnollisesti navigointimallien mukaisessa järjestämisessä tulee ottaa huomioon myös muut kuin säteittäinen navigointimalli. Koska hypertekstisivut muodostavat graafin, muodostuu visualisoinnista tässä tapauksessa puumainen, hierarkkinen esitys (Kuva 4.5). Kuten kolmannessa luvussa mainittiin, on tällaisella esitystavalla ongelmansa. Suurin ongelma on graafin viemä suuri tila sen suurentuessa. Aiemmin mainitussa WebView-sovelluksessa on sovellettu tätä järjestystapaa ja tilaongelma on ratkaistu siten, että käyttäjä pystyy piilottamaan puun haaroja tarvittaessa.



Kuva 4.5. Sivujen säteittäinen järjestäminen. [Cockburn et al., 1999]

Tilallinen järjestäminen on toinen vaihtoehto, mutta siinä on myös ongelmansa (ks. kohta 4.2). Myös käyttäjä voi tehdä tilallisen järjestämisen, mikä toisaalta aiheuttaa käyttäjälle lisää kognitiivista kuormitusta [Cockburn and Greenberg, 1999]. Myös sivustokartat ovat yksi vaihtoehto. Sivustokartat joudutaan luomaan staattisesti tai käsin, mikä rajoittaa niiden käyttöä yleisesti navigoinnin tukena.

Neljäntenä vaihtoehtona on ajallinen järjestys. Tämä järjestys on yleisin käytetty järjestys ja vastaa nykyisten selainten taakse- ja eteen-painikkeiden sekä historia-toiminnon käyttämää järjestystä. Ajallinen järjestys on käyttäjän sisäisen mallin kannalta helppokäyttöinen, koska navigointi on helppo jäsentää

ajallisesti. Käyttäjällä on kuva niistä paikoista, jossa hän on aiemmin käynyt, vaikka täsmällinen ajallinen järjestys paikkojen välillä ei olisikaan selvä. Ajallisesti järjestettyyn navigointihistoriaan tukeutuen käyttäjä osaa suunnilleen osoittaa hakunsa oikeaan kohtaan listassa.

Eri menetelmissä on omat puutteensa, eivätkä niitä hyödyntävät sovellukset ole johtaneet oleellisesti nykyistä tehokkaampaan käyttöliittymään. Oleellinen ongelma hyvän järjestyksen löytämisessä on se, että sen pitäisi toisaalta olla ajallisesti järjestetty, mutta myös ilmentää navigointitapahtuman toistuvaa ja rakenteellista luonnetta.

Joissakin toteutuksissa nämä kaksi eri järjestystä on otettu huomioon samanaikaisesti. Tauscher ja Greenberg [1997] ovat esittäneet rakenteisen, ajallisesti järjestetyn historia-listan seitsemän muun eri toteutuksen lomassa. Toteutukset arvioitiin sen perusteella, miten suurella todennäköisyydellä ne sisälsivät seuraavaksi vierailtavan sivun. Näistä kahdeksasta toteutuksesta Tauscherin ja Greenbergin lista saavutti parhaan tuloksen. On kuitenkin huomattava, että vaikka tämä ratkaisu saavutti parhaan tuloksen, vaati se myös käyttäjältä enemmän huomiota [Tauscher and Greenberg, 1997].

4.6. Uusi tapoja sivujen järjestämiseksi

Toisessa luvussa esitettiin neljä navigointimallia, jotka on löydetty tutkittaessa hypertekstiympäristössä liikkuvien käyttäjien navigointia. Yhteistä kaikille näille eri navigointimalleille on niiden muodostama yhdyssivu, joka muodostaa säteittäisen navigoinnin keskuksen ja silmukkamaisen navigoinnin silmukan aloituskohdan, kuin myös rengasmaisen navigointimallin silmukoiden aloituskohdat. Yksi mahdollinen tapa järjestää navigointihistoria on tehdä se näiden keskussivujen varaan. WebView-sovelluksessa tällaista mallia on käytetty ja siitä on saatu alustavasti hyviä tuloksia.

Navigointihistorian järjestäminen keskussivujen mukaisesti aiheuttaa puumaisen visualisoinnin, jonka suuruus kasvaa oleellisesti navigointitapahtuman edetessä. WebView-sovelluksessa estetään puun kasvaminen äärettömän korkeaksi aloittamalla hierarkian ensimmäiseltä tasolta uusi alipuu käyttäjän siirtyessä toiselle sivustolle. Cockburn et al. [1999] pitävät tätä huonona ratkaisuna siinä mielessä, että samassa asiayhteydessä olevat keskussivujen alisivut joutuvat puun eri haaroihin. Tämä on ratkaistu lisäämällä visualisointiin eri sivustojen muodostamien alipuiden välisten alisivujen välisiä linkkejä (Kuva 4.5).

Toinen tapa vähentää puiden muodostumista liian korkeaksi on hajottaa puun rakenne jollain toisella menetelmällä. Yksi vaihtoehto on erottaa jokainen keskussivu omaksi haarakseen, jolloin puu pysyy matalana rakenteena. Tästä

on myös se hyöty, että keskussivujen muodostama järjestys on helpommin mielletävissä ajallisesti järjestetyksi.

Hypertekstiympäristön kulloisenkin käyttötarkoituksen ja ratkaistavan tehtävän mukaan käyttäjillä on erilaisia navigointimalleja. Tietyn sivun löytäminen on yksi esimerkki navigoinnista, jossa haun aloitussivun ja ketjussa viimeisenä olevan löydetyn sivun väliset sivut eivät ole ollenkaan merkityksellisiä. Tällöin ei ole käyttäjän kannalta oleellista pitää yllä tietoa niistä sivuista, joiden läpi käyttäjä kulki pyrkiessään hakemalleen sivulle eikä niitä ole kaikissa tapauksissa järkevää lisätä käyttäjän navigointihistoriaan.

Tilanne muuttuu toiseksi, jos käyttäjä haluaa luoda yleiskuvan jostain hypertekstisivujen muodostamasta kokonaisuudesta ja sen jälkeen haluaa tutkia jotain aluetta tarkemmin. Tällöin navigointihistoriassa ovat oleellisia kaikki sivut. Navigointihistorian luomisen kannalta olisikin tärkeätä tietää, mitkä sivut ovat käyttäjän kannalta kulloinkin merkityksellisiä ja mitkä eivät. Tämän tiedon avulla navigointihistorian voisi myös pitää pienempänä ja se toimisi tehokkaammin käyttäjän apuna. Tulevaisuudessa navigoinnin tukemiseen tarkoitetut työkalut voisivat tukea paremmin käyttäjän toimia, jos niillä olisi mahdollisuus tietää, mikä käyttäjän kannalta on kulloinkin oleellista.

5. Käyttäjän selailuhistorian visualisoiva sovellus

Tutkielman osana toteutettiin sovellus, joka visualisoi navigointihistorian auttaa käyttäjää navigoimaan paremmin hypertekstiympäristössä. Sovelluksen avulla voidaan tutkia tarjoaako uudenlainen navigointihistorian esittäminen käyttäjälle paremmat mahdollisuudet navigointiin sekä tutkia, miten hypertekstiympäristössä liikkuja jäsentää navigointitapahtuman.

Tässä luvussa käydään ensin läpi sitä ympäristöä, johon sovellus on toteutettu. Ympäristön tarkastelu selventää sovellukselle asetettuja rajoituksia ja mahdollisuuksia. Sovelluksen toteutusympäristöksi valittiin WWW-ympäristö (*World Wide Web*) sen laajan suosion vuoksi. WWW-ympäristön tarkastelu aloitetaan esittämällä sen toimintaa yleisesti. Tämän jälkeen kerrotaan tarkemmin, mistä osista ympäristö koostuu ja miten tähän ympäristöön on mahdollista liittää uusia sovelluksia. Ympäristön tarkastelun jälkeen esitellään itse sovellus.

5.1. WWW-ympäristön toiminta käyttäjän navigoidessa

WWW-ympäristö on pääperiaatteeltaan kaksitasoiseen arkkitehtuuriin perustuva. Se koostuu WWW-palvelimista ja näihin yhteyttä ottavista WWW-asiakasohjelmista. Yleisimpiä asiakasohjelmia ovat WWW-selaimet, jotka hakevat palvelimilta hypertekstisivuja. Tämän perustoiminnallisuuden ohien on kehittynyt myös monia muita tekniikoita ja arkkitehtuureita.

Tässä kohdassa kuvataan WWW-ympäristön kaksitasoisen arkkitehtuurin toiminta käyttäjän navigoidessa. Vuorovaikutuksen tarkastelun perusteella selviää visualisoinnin luomiselle asetetut rajoitukset ja mahdollisuudet. Ensinnäkin kuvataan vuorovaikutusprosessi palvelimen ja selaimen välillä. Vuorovaikutusta ei kuvata protokollatasolla, koska se ei ole oleellista itse visualisoinnin kannalta.

WWW-sivut sijaitsevat palvelimella, josta ne haetaan WWW-selaimella ja näytetään käyttäjälle. WWW-sivujen haku palvelimelta tapahtuu URL-osoitteen perusteella. URL-osoite sisältää tiedon yhteyskäytännöstä, palvelimesta ja haettavan sivun sijainnista palvelimella. Esimerkiksi URL-osoite `<http://www.uta.fi/index.html/>` viittaa *index.html*-nimiseen WWW-sivuun *www.uta.fi*-palvelimella ja määrittää yhteyskäytännöksi *http*-protokollan. Asiakas lähettää palvelimelle *HTTP-pyyntöjä* (request), jotka sisältävät tiedot haettavasta sivusta. Yleisin pyyntö on GET-metodin kutsu, joka palauttaa asiakkaalle asiakkaan pyytämän WWW-sivun.

Aina WWW-sivut eivät sijaitse tiedostoina WWW-palvelimella vaan ne saatetaan esimerkiksi luoda dynaamisesti käyttäjän suorittaessa sivupyynnön.

Sivujen luonti dynaamisesti tapahtuu useasti CGI-tekniikka (Common Gateway Interface) hyödyntäen. Tässä ratkaisussa asiakkaan sivupyynnöksi kohdistuu WWW-palvelimella sijaitsevaan prosessiin, jonka kutsu tuottaa tuloksena WWW-sivun. Tällaisten prosessien toimintaa voidaan varioida liittämällä kutsuihin parametreja. Parametrien siirto onnistuu joko edellä kuvatun GET-metodin avulla, jolloin parametrin siirtyvät osana URL-osoitetta tai vaihtoehtoisesti POST-metodin avulla, jolloin parametrit siirtyvät tietovirrassa.

Yhteys asiakkaan ja palvelimen välillä ei ole *pysyvä* (persistent) [Gunnarsson, 1996] vaan uuden sivun hakemiseksi palvelimelta avataan aina yhteys uudelleen, siirretään tarvittavat tiedot ja suljetaan yhteys. Mikäli palvelimen puolella halutaan seurata yksittäisen käyttäjän liikkeitä, on asiakkaan tunnistetiedot lähetettävä aina palvelimelle jokaisen uuden yhteyden avaamisen yhteydessä. Tunnistetietojen lähetys tapahtuu GET- tai POST-metodia käyttäen edellä kuvatun parametrinvälityksen avulla.

Navigointitapahtuman alkaessa käyttäjä pyytää selainta hakemaan tietoa tietyistä URL-osoitteista. Selain hakee dokumentin kyseisestä URL-osoitteesta käyttämällä GET- tai POST-metodia ja näyttää käyttäjälle kyseisen dokumentin. Tämän jälkeen käyttäjä siirtyy seuraavalle WWW-sivulle käyttäen selainten tarjoamia keinoja (ks. taulukko 1), jolloin selain suorittaa uuden sivupyynnön palvelimelle. On myös mahdollista, että sivu on selaimen muistissa, jolloin uutta sivupyynnön tekemistä ei välttämättä tarvita. Tätä tapahtumasarjaa toistetaan koko navigoinnin ajan.

5.2. Visualisoinnin luominen hajautetusti

Neljännessä luvussa mainitut staattiset visualisoinnit luodaan yleensä etukäteen, jolloin ne ovat käyttäjien saatavilla tarvittaessa. Ne voidaan luoda automaattisesti WWW-palvelimilla. Dynaamiset visualisoinnit eroavat staattisista siinä, että ne ovat riippuvaisia käyttäjän liikkeistä WWW-sivustolla. Tästä johtuen visualisointia täytyy rakentaa sitä mukaa, kun käyttäjä navigoi WWW-sivustolla. Visualisoinnin kannalta oleellista dataa saattaa sijaita niin WWW-palvelimilla kuin myös asiakkaan puolella, jossa visualisointi näytetään. Tämän takia WWW-ympäristöön tehtävät visualisoinnit ovat yleensä hajautettuun arkkitehtuuriin perustuvia kuten WWW-ympäristön sovellukset muutenkin.

Käyttäjän navigoidessa WWW-sivustoilla joitain tietoja käydyistä sivuista siirtyy automaattisesti asiakkaan puolelle. Esimerkiksi URL-osoitteet ja sivujen sisältämä data on käytössä asiakkaan puolella joka tapauksessa. Näitä tietoja ei siis tarvitse erikseen hakea WWW-palvelimelta visualisointia ajatellen. Yleensä visualisoinnin näyttävä sovellus pystyy noutamaan WWW-ympäristöstä myös muuta tietoa, mutta se ei ole kaikissa tapauksissa järkevää. Esimerkiksi WWW-

sivulla olevien linkkien tarkistaminen ei ole järkevää asiakkaan puolella, mikäli se pystytään tekemään palvelimien puolella nopeammin. Palvelinten hyödyntäminen visualisoinnin rakentamisessa onnistuu käyttämällä joko *välityspalvelinta* (proxy) tai muita palvelimille rakennettuja sovelluksia.

Hyödynnettäessä välityspalvelinta asiakkaan sivupyynnöt eivät menekään suoraan palvelimelle, vaan ne ohjataan sinne välityspalvelimen kautta. Välityspalvelimen saadessa sivupyynnön se lähettää sen edelleen palvelimelle. Palvelin lähettää vastauksen välityspalvelimelle ja välityspalvelin lähettää edelleen saamansa vastauksen pyynnön lähettäneelle asiakkaalle. Välityspalvelimeen perustuvissa ratkaisussa visualisoinnin luonti tai sitä tukevat toimet tapahtuvat välityspalvelimella. Välityspalvelimen avulla voidaan luoda visualisointeja ilman, että asiakkaan puolelle täytyy asentaa mitään lisäohjelmia.

Esimerkki välityspalvelimen käytöstä on Weinreichin ja Lamersdorfin [2000] kehittämä ratkaisu, jossa käyttäjälle näytetään ylimääräistä tietoa WWW-sivun sisältämistä linkeistä. Tässä ratkaisussa asiakas lähettää välityspalvelimelle pyynnön hakea tietyssä URL-osoitteessa oleva dokumentti. Tämän jälkeen välityspalvelin hakee pyydetyn dokumentin palvelimelta ja muokkaa sitä ennen kuin palauttaa sen edelleen asiakkaalle. Välityspalvelin esimerkiksi tutkii dokumentin sisältämät linkit ja lisää dokumenttiin muun muassa tiedot siitä, onko linkki rikki, avautuuko linkin kohde uuteen ikkunaan tai onko linkin kohteena oleva sivu samalla palvelimella.

Välityspalvelimen käyttö on mahdollista ja järkevää vain joissain tapauksissa. Monikäyttöisempi ratkaisu on tehdä palvelimelle sovellus, jonka kanssa asiakkaan sovellus voi keskustella suoraan. Sovellusten keskustelu verkon yli onnistuu hyödyntämällä hajautettujen ohjelmistojen tekemiseen tarkoitettuja tekniikoita. Näistä kolme yleisintä ovat CORBA (Common Object Request Broker Architecture), DCOM (Distributed COM), tai Java RMI (Remote Method Invocation). Näiden hajautettujen ratkaisuiden avulla asiakaskoneella oleva sovellus pystyy keskustelemaan palvelimilla ajettavien ohjelmien kanssa ja hyödyntämään näiden tarjoamia palveluita.

5.3. Visualisoinnin liittäminen WWW-selaimeen

Visualisoinnin näyttämiseen WWW-selaimen yhteydessä on monia eri vaihtoehtoja. Näistä viisi yleisintä ovat: (1) visualisoinnin rakentaminen osaksi jo olemassa olevaa selainta, (2) *ohjelmalisäkkeen* (plug-in) käyttö, (3) ActiveX-tekniikan hyödyntäminen, (4) *sovelman* (applet) tekeminen ja (5) itsenäisen, selaimen kanssa keskustelemaan sovelluksen teko.

Visualisoinnin rakentaminen osaksi jo olemassa olevaa selainta on näistä käytettävyyden ja nopeuden kannalta paras vaihtoehto. Tarjolla on monenlaisia

avoimeen lähdekoodiin perustuvia ratkaisuita, joihin oman lisäyksen voi tehdä. Esimerkiksi Mozilla-selain on yksi tällainen avoimen lähdekoodin selain. Ongelmana suoraan selaimen tehtävissä lisäyksissä on olemassa olevaan koodiin ja ohjelmarakenteeseen tutustuminen ja tämän prosessin viemä pitkä aika. Tämän lisäksi lopullinen toteutus on täysin selainriippuvainen, mikä rajoittaa ohjelman käyttöä merkittävästi. Siinä tapauksessa, että avoimeen lähdekoodiin perustuvan ohjelman kehitys lopetetaan, täytyy toteutus tehdä kokonaan uudelleen.

Ohjelmalisäkkeet ovat toinen vaihtoehto visualisoinnin näyttämiseksi. Esimerkki laajalle levinneestä ohjelmalisäkkeestä on VRML-mallien tarkasteluun käytetty ohjelmalisäke. Ohjelmalisäkkeiden ongelmana on niiden suhteellisen suuri riippuvuus selaimesta ja myös se, että käyttäjien pitää asentaa ne erikseen selaimen yhteyteen. ActiveX-tekniikkaan perustuvat toteutukset ovat lähellä ohjelmalisäkkeitä. Niiden ongelmallisuutena on rajoittuminen pelkästään Microsoftin tuotteiden kanssa yhteensopiviksi.

Sovelman teko on myös yksi vaihtoehto. Sovelmien hyvinä puolina ovat niiden riippumattomuus selainympäristöstä ja käyttöjärjestelmästä. Hyvänä puolena on myös se, ettei käyttäjän tarvitse asentaa sovelmaa koneelleen erikseen vaan se on automaattisesti käytettävissä verkon yli. Sovelmien huonona puolena on niiden vaatima latausaika käynnistettäessä niitä verkon yli. Suuri sovelma saattaa vaatia pitkän latausajan. Tämä ongelma kuitenkin vältetään toteutettaessa navigointia tukevia visualisointeja sovelmien muodossa. Tämä johtuu siitä, että sovelma joudutaan lataamaan vain kerran istunnon alkaessa eikä jokaisen sivulatauksen yhteydessä. Ohjelmalisäkkeiden tavoin myös sovelmilla on vain rajoitettu pääsy ympäristössä sijaitseviin resursseihin ellei tätä erikseen *tunnisteiden* (certificate) avulla sallita.

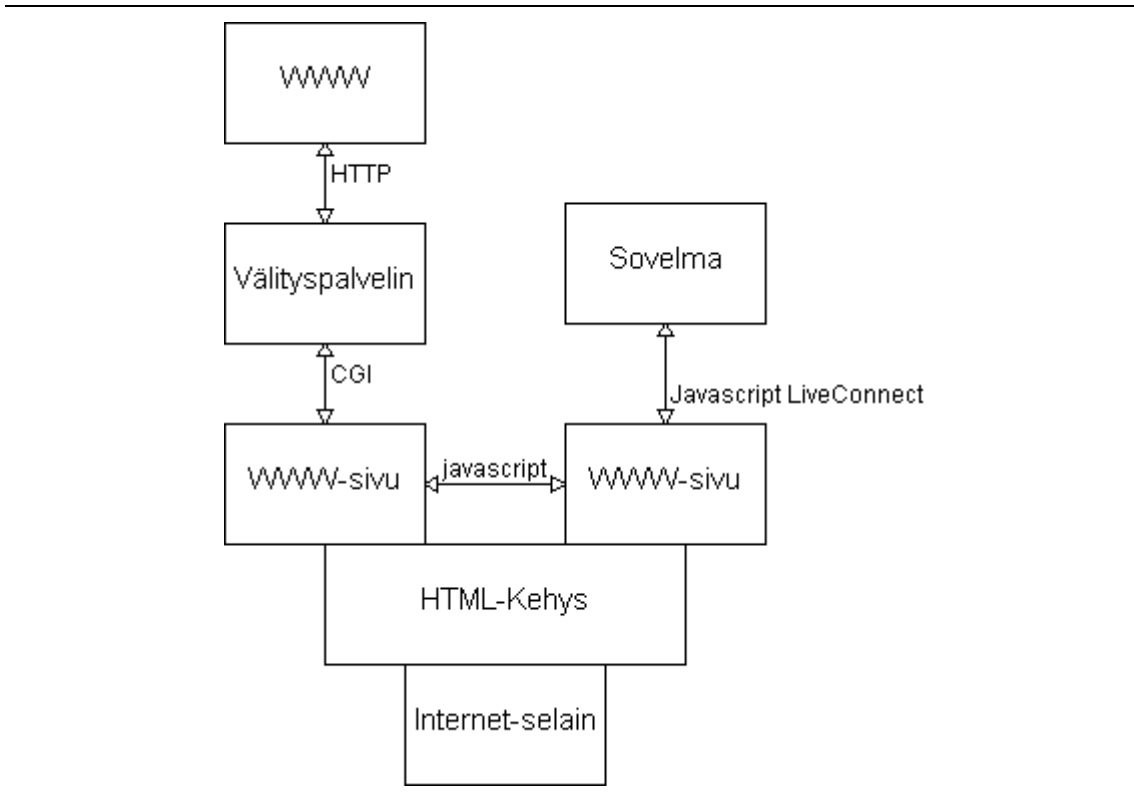
Viimeisenä vaihtoehtona voidaan mainita itsenäisen, selaimen kanssa keskustelevan sovelluksen tekeminen. Sellaisia selaimia, jotka tarjoavat ulkoapäin käsiteltävän rajapinnan, on kuitenkin vain muutama. Esimerkkinä voidaan mainita Netscapen selain Unix-käyttöjärjestelmän X Window -ikkunointijärjestelmään.

5.4. NavigationHelper-sovelluksen ympäristö

Toteutin NavigationHelper-sovelluksen, joka visualisoi käyttäjän navigointihistorian. Edellisessä kohdassa mainituista toteutusvaihtoehdoista tämä sovellus perustuu välityspalvelimen käyttöön ja sovelman käyttöön visualisoinnin näyttämässä käyttäjälle. Sovelluksen arkkitehtuuri tarjoaa yhden mahdollisen keinon liittää visualisointi osaksi nykyisiä WWW-selaimia.

Sovelman ja välityspalvelimen käyttö tarjoaa mahdollisimman yksinkertaisen ratkaisun visualisoinnin liittämiseksi osaksi WWW-selainta. Vaikka

sovelman ja välityspalvelimen käyttö toteutuksessa rajoittaa toteutettavien sovellusten ominaisuuksia, tarjoaa se kuitenkin riittävän määrän eri toteutusvaihtoehtoja.



Kuva 5.1. ApplicationHelper-sovelman ympäristö.

NavigationHelper-sovelluksen arkkitehtuuri koostuu pääpiirteissään selaimesta, välityspalvelimesta ja sovelmasta, jossa visualisointi näytetään (Kuva 5.1). Sovelma liitetään osaksi selainta HTML-kehysten avulla. Kehyksen toiseen osaan ladataan sovelma ja toisessa osassa näytetään tarkasteltavat WWW-sivut (Ks. kuva 5.3).

Netscape on kehittänyt LiveConnect-tekniikan, jonka avulla Javalla toteutetut sovelmat voivat keskustella selainten kanssa. LiveConnect-tekniikan avulla Javalla tehdyt sovelmat pääsevät käsiksi selaimen DOM-puuhun (Document Object Model), johon selain lataa WWW-sivujen sisältämät elementit. DOM-puussa on siis esitettynä kaikki tieto selaimessa näytettävistä WWW-sivujen elementeistä sekä muita tietoja selaimen tilasta. NavigationHelper-sovelma hyödyntää liityntää DOM-puuhun kutsuessaan WWW-sivuilla olevia javascript-funktioita.

Sovelma ladataan HTML-kehysten vasemmalle WWW-sivulle. Tällä WWW-sivulla on javascript-funktiot, joiden avulla NavigationHelper-sovelma keskustelelee HTML-kehysten oikeassa raamissa olevan WWW-sivun kanssa.

Sovelma kutsuu javascript-funktiota siinä tapauksessa, että käyttäjä on valinnut jonkun sovelman näyttämistä aikaisemmin vierailuista WWW-sivuista. Tässä tapauksessa sovelma käskää selainta lataamaan käyttäjän valitseman WWW-sivun HTML-kehysten oikeaan raamiin. Sivulataukset ovat ainoita pyyntöjä, joita sovelma tekee selaimelle.

Kun käyttäjä liikku WWW-sivuilla, latautuvat uudet WWW-sivut HTML-kehysten oikeaan raamiin (Ks. kuva 5.3). Joka kerta, kun oikeaan raamiin avautuu uusi sivu, saa sovelma tiedot uudesta sivusta. Sovelmalle välitetään WWW-sivun URL-osoite, otsikko (title-tunnisteen teksti) sekä sivulle johtaneen linkin teksti.

Jotta sovelmalle voitaisiin lähettää tiedot uusista WWW-sivuista, täytyy myös oikeassa raamissa näytettävillä WWW-sivuilla olla toiminto, joka välittää tiedot sovelmalle. NavigationHelper-sovelluksessa tämä on toteutettu siten, että tarkasteltavat WWW-sivut tulevat välityspalvelimen kautta, jossa WWW-sivuille lisätään tarvittavat tiedot välittävä javascript-koodi. Koodi välittää WWW-sivun latauksen yhteydessä tiedot uudesta WWW-sivusta sovelmalle.

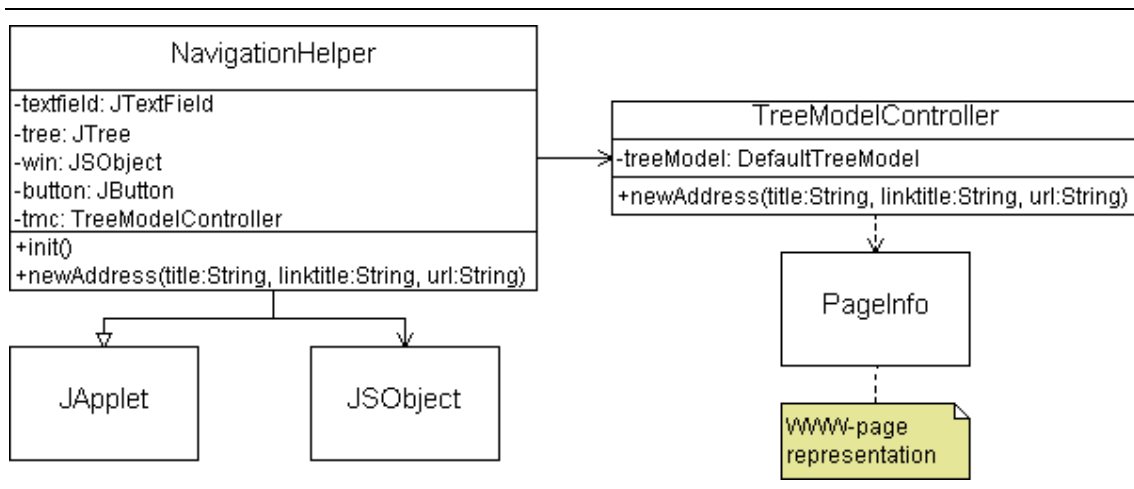
Tarkemmin ottaen ladattavan WWW-sivun javascript-koodi kutsuu sovelman sisältävän WWW-sivun javascript-funktiota välittäen tälle uuden sivun tiedot. Tämän jälkeen sovelman sisältävän WWW-sivun javascript-funktio kutsuu sovelman funktiota, joka lisää tiedot sovelmaan.

Välityspalvelimen käyttö on myös siinä suhteessa välttämätöntä, että eri URL-osoitteista tulevien WWW-sivujen keskustelu keskenään on estetty. Välityspalvelimen ansiosta sovelman sisältävän WWW-sivun sekä WWW-ympäristöstä haettavan WWW-sivun URL-osoite on selaimen näkökulmasta sama, mikä mahdollistaa vuoropuhelun sivujen välillä.

5.5. NavigationHelper-sovelman rakenne

Sovellus perustuu suurelta osin MVC-arkkitehtuuriin (Model Controller View). Perinteisestä MVC-arkkitehtuurista sovelma poikkeaa kuitenkin siinä, että ohjain ja malli ovat hieman tavallista kiinteämmin yhteydessä toisiinsa. Sovelma on kuitenkin niin suppea, että varsinaiseen MVC-arkkitehtuuriin ei ole tarvetta.

NavigationHelper-sovelman tärkeimmät luokat ovat: (1) NavigationHelper, (2) TreeModelController ja (3) PageInfo (Kuva 5.2). NavigationHelper-luokka hoitaa näkymän alustamisen ja päivittämisen. Jokaisen Java-sovelman tavoin luokka perii ominaisuudet sovelman tavittavan toteutuksen JApplet-luokalta. Näkymässä WWW-sivut esitetään hakemistorakenteena, jonka esitykseen valmis JTree-luokka tarjoaa miltei kaikki tarvittavat ominaisuudet. Tämän lisäksi NavigationHelper-luokka sisältää viitteen JObject-luokan ilmentymään, jonka avulla hoidetaan yhteys WWW-selaimen DOM-puuhun.

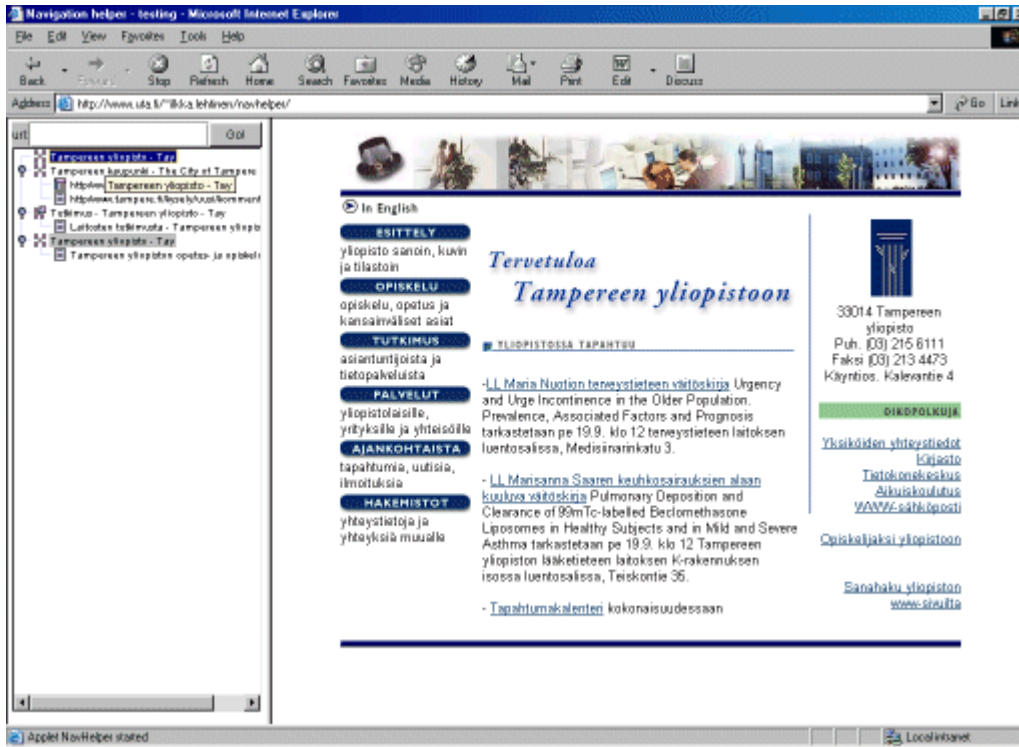


Kuva 5.2. NavigationHelper-sovelluksen luokkakaavio.

Selailuhistoria esitetään sovelmassa puurakenteena. Tämän puurakenteen esityksestä ja muutoksista vastaa TreeModelController-luokka. Javan tarjoama DefaultTreeModel-luokka tarjoaa valmiin toteutuksen sovelmassa käytettävälle puu-tietorakenteelle. Sovellus ei kohdistu puu-tietorakenteeseen muita operaatioita kuin uusien solmujen lisäyksiä. Tietorakenteessa WWW-sivut esitetään PageInfo-luokan avulla. Luokka sisältää tiedot sivun otsikosta, URL-osoitteesta sekä muista sivuun liittyvistä tiedoista.

5.6. NavigationHelper-sovelluksen käyttöliittymä

NavigationHelper-sovelma latautuu osaksi WWW-selainta HTML-kehikseen (Kuva 5.3). Kehyksen vasemmassa raamissa on NavigationHelper-sovelma ja oikeassa tarkasteltava WWW-sivu. Sovelma sisältää toiminnot sivuhistoriassa taaksepäin kulkemiseen (back), sivujen uudelleenlataukseen (refresh), sovelman uudelleenkäynnistykseen (restart) sekä URL-kentän, jonka avulla käyttäjä pystyy siirtymään haluttuun URL-osoitteeseen.

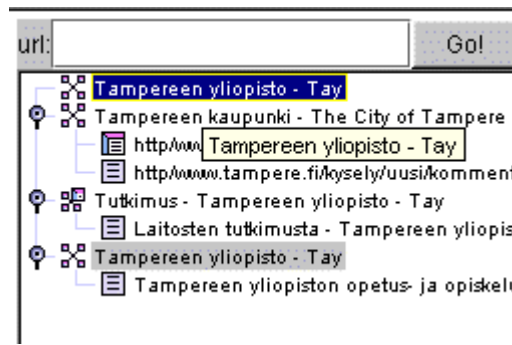


Kuva 5.3. NavigationHelper osana selainta.

Sovelman yläosan työkalujen alapuolella on selailuhistorian WWW-sivujen näyttämiseen käytetty alue (Kuva 5.4). Sivut esitetään tässä näkymässä siten, että ylimpänä on aina viimeisin vierailtu sivu, ja tämän alapuolella aiemmin vierailleet sivut. Sivut esitetään näkymässä siten, että uudemmat sivut ovat aina ylempänä listassa kuin vanhemmat sivut.

Sivujen esittämiseen käytetään näkymässä sekä tekstiä että ikoneita. Sivun voidaan esittää neljällä eri ikonilla. Sivun liitettävän ikonin tyyppiin vaikuttaa WWW-sivun ominaisuudet ja WWW-sivun rooli käyttäjän selailuhistoriassa. Ikonilla, jossa on kolme vaakasuuntaista viivaa ja violetti tausta, kuvataan tavallista WWW-sivua, jolla ei ole mitään erityistä roolia käyttäjän selailuhistoriassa. Samankaltaisella ikonilla kuvataan sivua, joka on WWW-ympäristössä pääsivu, mutta ei käyttäjän selailuhistoriassa sisällä mitään erityistä roolia. Tällä ikonilla on sininen reunus ja punainen yläosa.

Kaksi muuta ikonia kuvastavat edellisten ominaisuuksien lisäksi WWW-sivun erityistä roolia käyttäjän selailuhistoriassa. Ikonilla, jossa on neljä sakaraa käytetään kuvaamaan WWW-sivua, jolla käyttäjä on vierailut monta kertaa ja joka on tavallinen WWW-sivu. Edellisen määrittelyn perusteella voidaan sanoa, että tällä ikonilla kuvataan säteittäisnavigoinnissa syntyviä keskussivuja. Vastaavasti neljä sakaraa sisältävä ikoni, jossa yksi sakara on muita paksumpi, kuvaa WWW-sivua, joka on sekä keskussivu, että pääsivu.



Kuva 5.4. NavigationHelper-sovelman käyttöliittymä.

WWW-sivun voidaan päätellä olevan pääsivu, jos sen nimi sisältää "index"-merkkijonon tai sivun URL-osoite ei määrittele WWW-sivulle tiedostonnimeä. Jälkimmäisessä tapauksessa tiedoston nimi sisältää myös "index"-merkkijonon, koska tätä käytetään oletusarvona URL-osoitteissa, joissa tiedoston nimeä ei ole määritelty. Tämä jako pääsivuihin ja tavallisiin sivuihin on melko keinotekoinen eikä kovinkaan luotettava.

Ikonin ohella WWW-sivu esitetään näkymässä merkkijonona, joka pyrkii kuvaamaan sivua. Yleensä tämä merkkijono on WWW-sivun title-tunnisteen sisältämä merkkijono. Mikäli title-tunnisteen avulla ei ole määritelty nimeä sivulle, käytetään sivua kuvaamaan sille johtaneen linkin nimeä. Linkit on useasti nimetty suhteessa linkin sisältävään sivuun eikä tämä nimi välttämättä ole käyttäjälle avuksi sivua tunnistettaessa. Linkin nimen esittäminen on kuitenkin parempi vaihtoehto kuin tyhjä merkkijono. Mikäli myöskään linkin nimeä ei ole saatu selville, esitetään WWW-sivu näkymässä sen URL-osoiteen avulla.

Sen lisäksi, että sivua kuvaavat merkkijonot voivat olla käyttäjän kannalta mitäänsanomattomia, voivat ne olla myös liian pitkiä. Tässä sovelluksessa nimien pituusongelmaa on pyritty välttämään yrittämällä mahdollisuuksien mukaan lyhentää merkkijonoja. Merkkijonoa lyhennetään ottamalla niiden keskiosasta merkkejä pois. Merkkejä poistetaan tilanteessa, jossa edellisen WWW-sivun nimi sisältää samoja merkkejä. Tällainen tilanne syntyy usein silloin, kun sivuston nimi on title-tunnisteessa aina ensimmäisenä ja tämän jälkeen on esitetty sivuston osan nimi.

Esimerkiksi Tampereen yliopiston pääsivun nimi on "Tampereen yliopisto - Tay" ja opiskelusivun nimi "Tampereen yliopisto - opetus- ja opiskelusivut". Tässä tilanteessa jälkimmäistä nimeä voidaan lyhentää muotoon "Tampe... - opetus ja opiskelusivut" lyhemmän esityksen saavuttamiseksi. Tällainen esitys on myös käyttäjän kannalta usein mielekäs ja mahtuu pienempään tilaan.

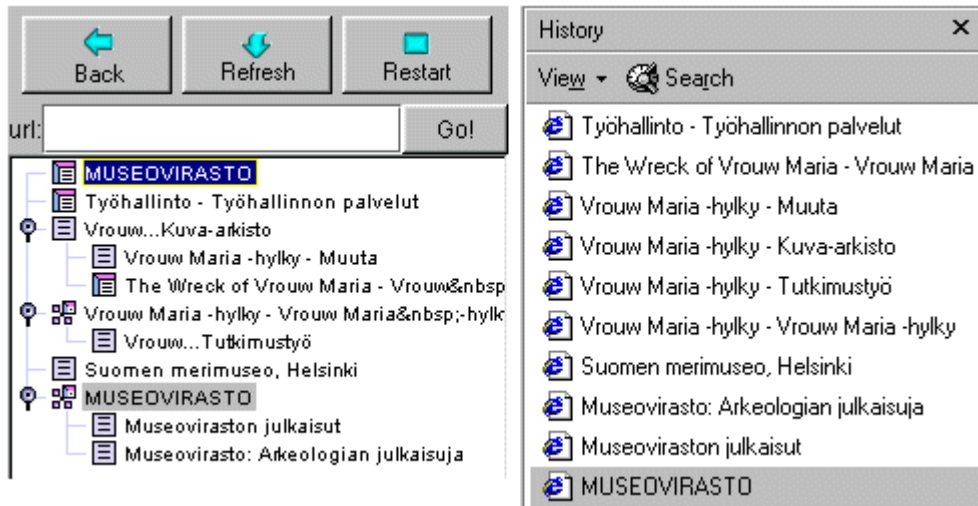


Kuva 5.5. Sivuhierarkia (vasen) ja sivujen vierailujärjestys.

NavigationHelper järjestää selailuhistorian käyttäjän WWW-sivujen selailun mukaisesti. Kuvassa 5.5 on esitetty osa Museoviraston WWW-sivujen (<http://www.nba.fi/>) sivuhierarkiasta. Tämän hierarkian läpikäymiseen on monia mahdollisuuksia, joista yksi on esitetty kuvassa. Suluissa olevat sivujen nimet kertovat käyttäjän siirtymisestä taaksepäin selailuhistoriassa, joko taaksepainikeen tai muun vastaavan toiminnon avulla.

Käyttäjä aloittaa sivujen läpikäynnin Museoviraston pääsivulta. Tältä sivulta käyttäjä siirtyy sivulle, jossa esitellään arkeologian julkaisuja. Tämän jälkeen käyttäjä palaa takaisin pääsivulle ja siirtyy edelleen Suomen merimuseota esittelevälle sivulle. Suomen merimuseon sivuilla käyttäjä käy tarkastelemassa tarkemmin Vrouw Maria -hylystä kertovia sivuja.

Kuvassa 5.6 on esitetty kaksi vaihtoehtoista tapaa edellisen navigointihistorian esittämiseksi. Kuvassa vasemmalla on NavigationHelper-sovelluksen esitys selailuhistoriasta ja oikealla Internet Explorerin vaihtoehtoinen esitys. Internet Explorer sisältää myös kolme muuta tapaa näyttää selailuhistoria. Kuvassa sivut ovat järjestetty siten, että viimeksi vierailtu sivu on ylimpänä listassa. Vaihtoehtoisesti sivut voidaan järjestää päivän, vierailujen tai sivuston perusteella.



Kuva 5.6. NavigationHelperin (vasen) ja Internet Explorerin selailuhistoria.

NavigationHelper-sovellus järjestää sivut käyttäjän liikkeen mukaan, etsien käyttäjän navigoinnista navigointikuvioita (Ks. kuva 2.5). Tällä tavalla järjestettynä (1) säteittäisnavigoinnissa esiintyvät sivut järjestyvät keskussivun alaisuuteen, (2) silmukkaisessa navigoinnissa silmukkaan liittyvät sivut järjestyvät silmukan aloitussivun alaisuuteen, (3) polkumaisessa navigoinnissa polun sivut järjestyvät polun aloitussivun alaisuuteen ja (4) rengasmaisessa navigoinnissa renkaiden sivut järjestyvät renkaiden aloitussivujen alaisuuteen.

Kuvan 5.5 esimerkissä on sekä säteittäistä että polkumaista navigointia. Navigoinnin alussa käyttäjä aloittaa Museoviraston etusivulta ja siirtyy arkeologian julkaisuja esittelevälle sivulle. Tämän jälkeen käyttäjä palaa takaisin Museoviraston aloitussivulle, jonka johdosta syntyneen säteen sivut siirtyvät keskussivun alaisuuteen. Vastaavasti toiseksi säteittäisnavigoinnin keskussivuksi muodostuu myöhemmin vierailtava Vrouw Maria -hylystä kertova sivu. Esimerkin navigoinnissa syntyy polku, kun käyttäjä siirtyy Vrouw Maria -hyllyn kuva-arkistosta kertovalle sivulle ja edelleen muille hylystä kertoville sivuille. Näiden sivujen keskuudesta ei löydy keskussivua ennen kuin käyttäjä siirtyy Työministeriön sivuille ja näin katkaisee navigointikuvion.

Navigoinnissa syntyvät kuviot ovat useasti sisäkkäisiä. Esimerkiksi käyttäjä voi kulkea silmukan mukaista reittiä pitkin tehden yhdestä silmukkaan kuuluvasta sivusta keskussivun säteittäiselle navigointikuviolle ja jatkaen edelleen silmukassa kulkemista. Mikäli sisäkkäiset navigointikuviot otettaisiin huomioon, syntyisi selailuhistoriaa esittävän puun rakenteesta syvä hierarkia. Tällaisen hierarkian muodostaa esimerkiksi WebView-sovellus (Kuva 4.5).

NavigationHelper-sovelluksessa hierarkian syntyminen estetään hajottamalla syntyvä hierarkia ajan suhteen. Syntyviä navigointikuvioita ei liitetä edellisten navigointikuvioiden osaksi vaan jokainen navigointikuvio esitetään päätasolla muiden navigointikuvioiden rinnalla. Aiemmin esitetyssä esimerkissä Suomen merimuseon pääsivu jää irralliseksi muista navigointikuvioista, koska käyttäjä muodostaa museon sivulla käymisen jälkeen uuden navigointikuvion Vrouw Maria -hylystä kertovan sivun ympärille.

Koska sisäkkäisiä navigointikuvioita ei esitetä NavigationHelper-sovelluksessa, tulee esitykseen usein sama WWW-sivu moneen kertaan, jokaisella kerralla liittyen eri navigointikuvioon. Sovelluksessa moneen kertaan esiintyvät sivut esitetään käyttäjälle harmaan taustan avulla (Kuva 5.6). Tämän korostustoiminnon avulla käyttäjä pystyy helpommin näkemään mitkä muut sivut liittyvät tähän kyseiseen sivuun aiemmista navigointikuvioista.

6. Käyttöliittymän testaaminen ja arviointi

Tässä luvussa käydään läpi NavigationHelper-sovelluksen arvioimiseksi järjestettyjä testejä. Käyttöliittymä tarjoaa uudenlaisen tavan selailuhistorian jäsentämiseksi ja tämän menetelmän toimivuuden toteamiseksi testaaminen oikeassa ympäristössä on välttämätöntä. Testien avulla haluttiin saada vastaus siihen, onko käyttöliittymä mahdollisesti parempi kuin jo olemassa olevat käyttöliittymät ja millaisena käyttäjät sen käytön kokevat.

Ennen testien tekemistä käyttöliittymän onnistuneelle toiminnalle asetettiin tavoitteet. Uuden käyttöliittymän tulisi helpottaa käyttäjän siirtymistä aiemmin vierailemalleen sivulle. Käyttöliittymä siis toimisi selailun apuvälineenä tehokkaammin kuin nykyiset selaimet. Toisena tavoitteena oli saada positiivista palautetta uudesta käyttöliittymästä ja siten tukea sille ajatukselle, että se tarjoaa nykyisiin selaimiin verrattuna paremman käyttöliittymän.

Toteutettu käyttöliittymä on hyvin samankaltainen kuin nykyisissä selaimissa olevat käyttöliittymät, minkä vuoksi oli luontevaa verrata uuden käyttöliittymän toimintaa jo olemassa olevien käyttöliittymien toimintaan. Testien tulosten avulla on mahdollista sanoa, tarjoaako uusi käyttöliittymä paremman käyttöliittymän kuin nykyiset selaimet. Sen lisäksi, että käyttöliittymää verrattiin muihin vastaaviin käyttöliittymiin haluttiin tietää miten käyttäjät suhtautuvat uuteen käyttöliittymään. Henkilökohtaiset arvioit käyttöliittymästä antavat kuvan siitä, onko käyttöliittymän toteutuksessa onnistuttu halutulla tavalla.

6.1. Testien järjestelyt

Testeissä NavigationHelper-sovellusta verrattiin Mozilla Firebird -selaimen tarjoamaan selailuhistoriaan. Vertailtavaksi selaimeksi Mozilla valittiin sen takia, että sen tarjoama selailuhistoria oli tarpeeksi nopeasti päivittyvä ja se tarjosi myös vaihtoehtoisia tapoja selailuhistorian järjestämiseksi.

Testiin valittiin kaksi WWW-sivustoa, Työministeriön kotisivu (www.mol.fi) ja Museoviraston kotisivu (www.nba.fi). Kummatkin näistä sivustoista ovat nopeatoimisia, mikä on tärkeätä testin onnistumisen kannalta. Valitut sivustot sisälsivät myös selkeän hierarkian, joka helpotti testitehtävien suunnittelua ja testihenkilöiden navigointia sivuilla. Oleellista oli myös se, että kummankin sivuston sivut oli nimetty title-tunnisteen avulla järkevästi, koska kummatkin vertailtavista selailuhistorioista käyttivät title-tunnisteita sivujen esittämiseen.

Testejä varten tein kaksi erillistä tehtäväsarjaa, jotka kummatkin sisälsivät 41 tehtävää (Liite 3). Tehtäväsarjoissa oli kahdenlaisia tehtäviä, joista toisessa

tehtävätyypissä käyttäjän tuli etsiä WWW-sivuilta tietty sivu ja toisessa käyttäjän tuli siirtyä jollekin aiemmin vierailemalleen sivulle. Näitä ns. paluutehtäviä oli kummassakin tehtäväsarjassa kahdeksan sijoiteltuna eri kohtiin tehtäväsarjoja.

Testeissä tehtäväsarjojen ja käyttöliittymien järjestystä vaihdeltiin siten, että jokainen tehtäväsarja ja käyttöliittymä oli kulloinkin käytössä eri järjestyksessä. Ennen tehtäväsarjojen tekemistä testihenkilöille annettiin täytettäväksi taustatietolomake. Lomakkeen täytön jälkeen testihenkilöille kerrottiin, kuinka tiedot käyttäjän toimista kerätään videolle ja kuinka tehtäväsarjat tulee suorittaa. Tehtäväsarjojen suorituksen jälkeen kukin testihenkilö täytti kertoi mielipiteensä käyttöliittymistä kyselylomakkeen avulla.

Selailuhistorian toimintaa ei kummankaan käyttöliittymän kohdalla selvitetty käyttäjälle erityisen tarkasti. Navigoinnin helpottamiseen tarkoitetun selailuhistorian tulisi ideaalitapauksessa olla sellainen, että sen käyttö onnistuu intuitiivisesti, ilman erillisiä ohjeita.

Tehtävien suoritusten analysoimiseksi tietokoneen ruutu videoitiin tehtävien suorituksen aikana. Analysoitavia asioita olivat erilaiset käyttöliittymätapahtumat tehtävien suoritusten aikana. Kunkin tehtävän kohdalla kirjattiin tehtävän suoritukseen kulunut aika, taakse-painikkeen painokerrat, selailuhistorian painokerrat, sivulatausten määrä sekä tehtävän suoritustarkkuus selailuhistoriaa käytettäessä. Edellä mainitut tiedot kerättiin jokaisesta testien aikana suoritetusta tehtävästä.

6.2. Koehenkilöiden taustatiedot

Testeihin osallistui 13 henkilöä, joista naisia oli kymmenen ja miehiä kolme. Ennen testiä testihenkilöitä pyydettiin täyttämään taustatietolomake. Lomakkeen tarkoituksena oli selvittää testihenkilöiden kokemusta Internetin suhteen ja erityisesti tehdä suppea selvitys siitä, miten testihenkilöt ovat tottuneet käyttämään Internet-selainta.

Testihenkilöistä kuusi ilmoitti olevansa selainten käyttäjinä peruskäyttäjiä. Neljä kertoi käyttävänsä myös joitain selainten erikoistoimintoja ja loput kolme ilmoittivat räätälöivänsä selaimen ulkoasun ja toiminnan mieleisekseen. Selainten käyttäjinä testihenkilöt jakautuivat melko tasan kolmeen edelliseen ryhmään kuitenkin siten, että peruskäyttäjiä oli eniten.

Koska testissä käytettiin yleisesti nähtävillä olevia Työministeriön ja Museoviraston sivuja, kysyttiin käyttäjiltä myös, kuinka tuttuja nämä sivut olivat heille. Museoviraston sivuilla testihenkilöistä aiemmin oli käynyt vain yksi henkilö. Museoviraston sivu oli siis suurimalle osalle testihenkilöistä tuntematon. Sen sijaan Työministeriön sivuilla oli käynyt kymmenen henkilöä eli miltei jokainen.

Kaikki käyttäjät kävivät säännöllisesti Internetissä. Seitsemän henkilöä ilmoitti käyvänsä Internetissä useita kertoja päivässä. Päivittäin Internetissä ilmoitti käyvänsä kolme henkilöä ja useita kertoja viikossa myös kolme.

Testin kannalta ehkä oleellisin taustatietokysymys koski selailuhistorian käyttöä Internetissä liikkumisen ohella. Henkilöistä kolme ilmoitti käyttävänsä selailuhistoriaa useasti. Joskus selailuhistoriaa ilmoitti käyttävänsä suurin osa eli kahdeksan testihenkilöä. Testihenkilöistä kaksi ei ollut koskaan käyttänyt selailuhistoriaa.

Viimeiseksi taustatietolomakkeessa tiedusteltiin joidenkin väittämien paikkansapitävyyttä testihenkilön kohdalla. Selaimen käyttöä koskevissa väittämissä tiedusteltiin usean selainikkunan käyttämistä. Vain yksi testihenkilö ilmoitti pitävänsä selailun aikana yleensä vain yhtä ikkunaa auki. Suurin osa eli yhdeksän henkilöä ilmoitti pitävänsä useita ikkunoita auki samanaikaisesti ja tietävänsä kulloinkin mitä missäkin ikkunassa on. Loput kolme ilmoitti pitävänsä myös useita ikkunoita auki, mutta unohtavansa silloin tällöin mikä sivu on avoinna missäkin ikkunassa.

Testihenkilöistä kolme tunnusti harhailevansa joskus WWW-sivuilla muistamatta mitä alun perin lähti hakemaan. Seitsemän testihenkilöä ilmoitti olevansa selailun suhteen järjestelmällisiä. Edelleen viisi henkilöä ilmoitti käyttävänsä kirjanmerkkejä useasti. WWW-ympäristön käytön koki mukavaksi kymmenen henkilöä. Yli puolet eli seitsemän koehenkilöä ilmoitti hukkaavansa ajoittain sivuja löytämättä myöhemmin sivulle takaisin.

Yleisesti ottaen testiin osallistuvat henkilöt edustivat melko tasaisesti erilaisia selainten käyttäjiä eikä mikään ryhmä ollut ylikorostettuna. Osa käytti selainta päivittäin ja hyödynsi myös selainten tarjoamia vähemmän käytettyjä toimintoja, kun taas osa käytti selainta harvoin tyytyen käyttämään selaimen tarjoamia perusominaisuuksia.

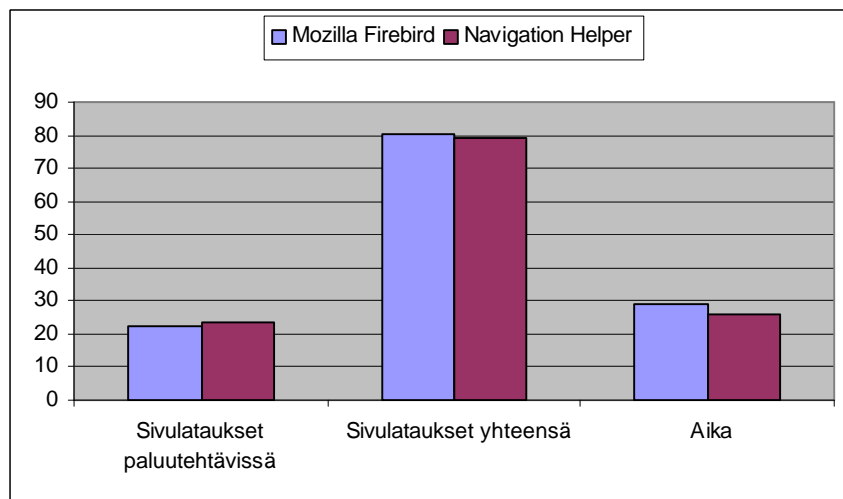
6.3. Käyttöliittymien käytössä havaitut erot

Testiin osallistuneista 13 henkilöstä kukin suoritti kaksi tehtäväsarjaa – kummankin eri käyttöliittymällä. Tehtäväsarjoja ja käyttöliittymiä vaihdeltiin siten, että kaikki tehtäväsarjojen ja käyttöliittymien mahdolliset järjestykset tulivat useaan kertaan testatuiksi. Kolmentoista testihenkilön joukosta poistettiin yhden testihenkilön tulokset, jotta testiaineisto saatiin symmetriseksi selainten ja tehtäväsarjojen suhteen. Tämän testihenkilön tulos poikkesi eniten kaikkien muiden testihenkilöiden tuloksista.

Testihenkilöt jakautuivat neljään eri ryhmään sen mukaan, kuinka he käyttivät tarjolla olevien käyttöliittymien ominaisuuksia tehtäviä tehdessään. Kaksi testihenkilöä suoritti kummankin tehtäväsarjan käyttämättä taaksepainiketta tai selailuhistoriaa. Nämä testihenkilöt liikkuivat sivustoilla

pelkästään seuraamalla linkkejä ja siirtymällä sivustojen välillä osoite-kentän avulla. Lisäksi kaksi testihenkilöä suoritti tehtävät käyttämällä edellä mainittujen keinojen lisäksi myös taakse-painiketta. Viisi testihenkilöä suoritti tehtävät käyttäen pääasiallisesti vain selailuhistoriaa. Jäljelle jäivät neljä testihenkilöä käyttivät tehtävien tekemiseen taakse-painiketta ja selailuhistoriaa.

Tehtäväsarjojen suoritukseen liittyvät keskiarvot on esitetty kuvassa 6.2. Tehtävien suoritusten analysoinnissa kunkin tehtävän kohdalla on mitattu tehtävän tekemiseen kulunut aika ja sivulatausten määrä. Paluutehtävien kohdalla on myös selvitetty, miten paluutehtävä on onnistunut. Sivulatausten määrä kertoo sen, kuinka useasti testihenkilö on siirtynyt sivujen välillä tehtävää suorittaessaan. Paluutehtävissä sivulatausten määrä on ideaalitapauksessa yksi, jolloin testihenkilö on löytänyt aiemmin vieraillun sivun selailuhistoriasta ja siirtynyt suoraan tälle sivulle. Tällaiset yhdellä sivulatauksella suoritettavat paluutehtävät on laskettu erikseen ja ilmaistu tehtävän kohdalla täysosumana.

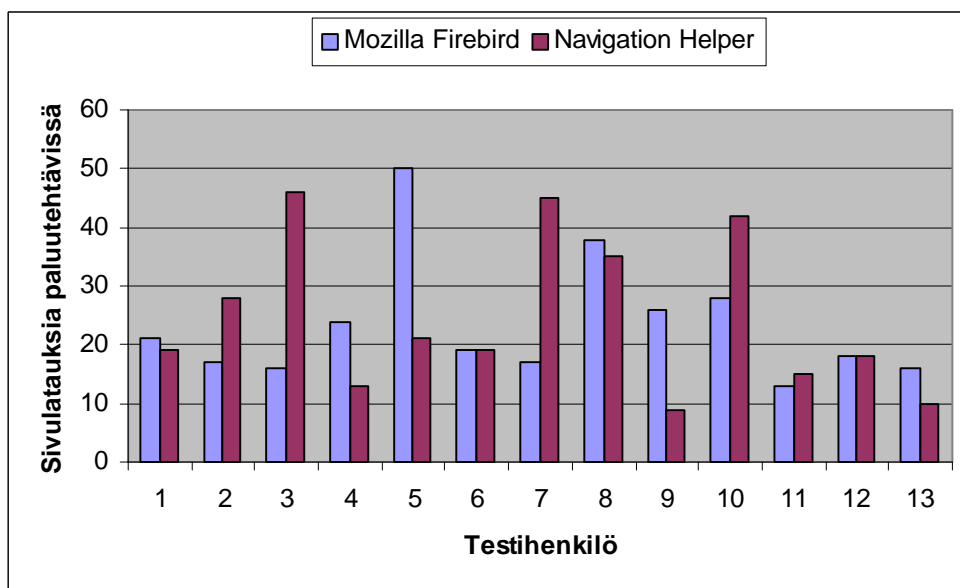


Kuva 6.2. Tehtäväsarjojen suoritusarvot eri käyttöliittymillä.

Kuvasta 6.2 nähdään, että molempien käyttöliittymien yhteenlasketut tulokset ovat hyvin lähellä toisiaan. Paluutehtävissä NavigationHelperin käyttäjät ovat käyttäneet keksimäärin hieman enemmän sivulatauksia, kuin Mozilla Firebird –selaimen käyttäjät. Koko tehtäväsarjojen suorituksessa Mozilla Firebird –selaimen käyttäjät ovat käyttäneet hieman enemmän sivulatauksia. Tehtävien tekemiseen kulunut aika on keskimäärin pienempi NavigationHelperiä käytettäessä.

Erot keskiarvoissa osoittautuivat erittäin pieniksi. Sen sijaan yksityiskohtaisempi tarkastelu osoittaa, että eroja löytyy selainten väliltä.

Kuvassa 6.3 on esitetty testihenkilöittäin paluutehtäviin käytetyt sivulataukset. Kuvasta voidaan huomata, että osalla käyttäjistä erot eri selainta käytettäessä ovat suuret. Testihenkilöistä suurin piirtein puolet on suorittanut tehtäväsarjat molemmilla selaimilla käyttäen yhtä monta sivulatausta. Toinen osa testihenkilöistä on käyttänyt tehtäväsarjojen tekemiseen huomattavasti enemmän sivulatauksia toisella selaimella. Kuten jo aiemmin mainittiin jakautuivat käyttäjät neljään eri ryhmään sen mukaan, miten he selaimen käyttöliittymää hyödynsivät. Testihenkilöt 1, 2, 4, 6, 9, 11, 12 ja 13 käyttivät pääasiassa selailuhistoriaa paluutehtävissä. Näiden testihenkilöiden kohdalla selaimet ovat olleet tasavertaisia tai NavigationHelper on tuottanut paremman tuloksen. Testihenkilöt, joiden kohdalla sivulatauksia on keskimääräistä enemmän, ovat käyttäneet paluutehtävissä jotain muuta menetelmää selailuhistorian sijasta. Testihenkilöt 5 ja 8 ovat kulkeneet enimmäkseen linkkejä pitkin ja testihenkilöt 7 ja 10 ovat käyttäneet runsaasti taakse-painiketta.

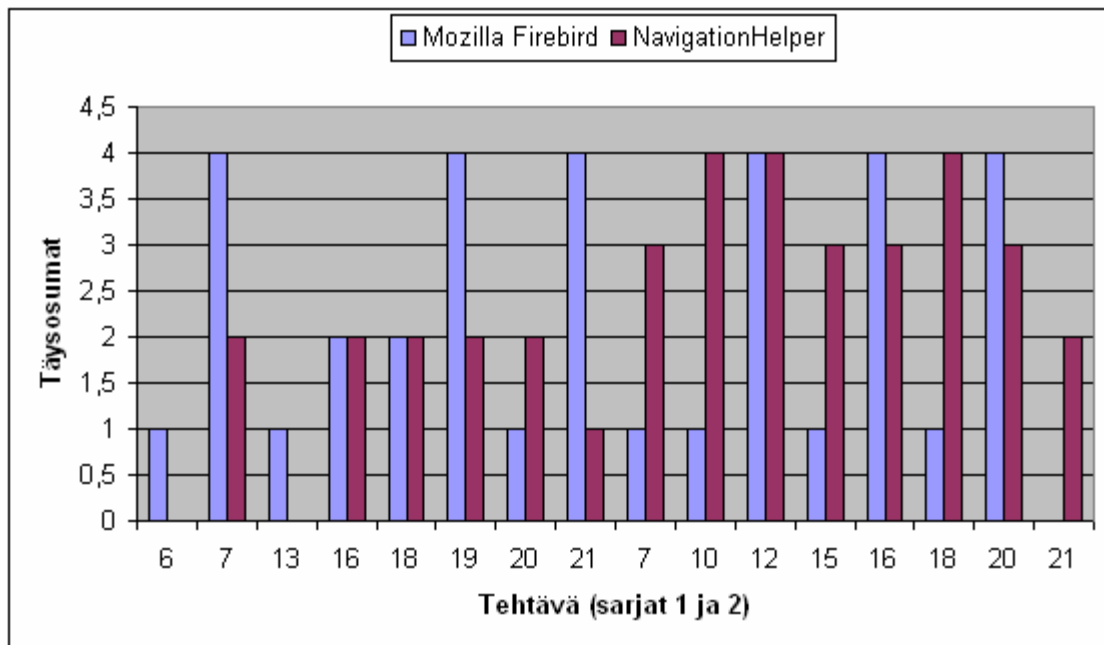


Kuva 6.3. Paluutehtäviin käytetyt sivulataukset testihenkilöittäin.

Kuvassa 6.4 on esitetty paluutehtävien täysosumien määrä tehtävittäin. Kuvassa vasemmalla puolella ovat ensimmäisen tehtäväsarjan paluutehtävät 6, 7, 13, 16, 18, 19, 20 ja 21. Vastaavasti oikealla puolella ovat toisen tehtäväsarjan paluutehtävät 7, 10, 12, 15, 16, 18, 20 ja 21. Tästä kuviosta nähdään, että toisen tehtäväsarjan paluutehtävät ovat olleet ensimmäisen tehtäväsarjan tehtäviä helpompia. Mielenkiintoinen havainto on, että Mozillan käyttäjät ovat joko kaikki saaneet suoritettua paluutehtävän yhdellä sivulatauksella tai vain harvat

ovat saaneet suoritettua tehtävän yhdellä sivulatauksella. NavigationHelper-sovelluksen kohdalla tällaista jakoa ei esiinny vaan yleensä käyttäjistä useampi kuin yksi on kyennyt yhden sivulatauksen suoritukseen. Käyttäjien suoritukset eroavat täysosumien suhteen enemmän NavigationHelper-sovelluksen kohdalla kuin Mozillan kohdalla.

Paluutehtävien kohdalla sivulataukset ja täysosumat ovat suoraan verrannollisia toisiinsa. Esimerkiksi toisen tehtäväsarjan tehtävän 12 ovat kaikki käyttäjät saaneet suoritettua yhdellä sivulatauksella. Vastaavasti ensimmäisen tehtäväsarjan tehtävät 13 ja 16 ovat vaatineet keskimäärin muita tehtäviä enemmän sivulatauksia ja näiden tehtävien kohdalla täysosumien määrä onkin minimaalinen.



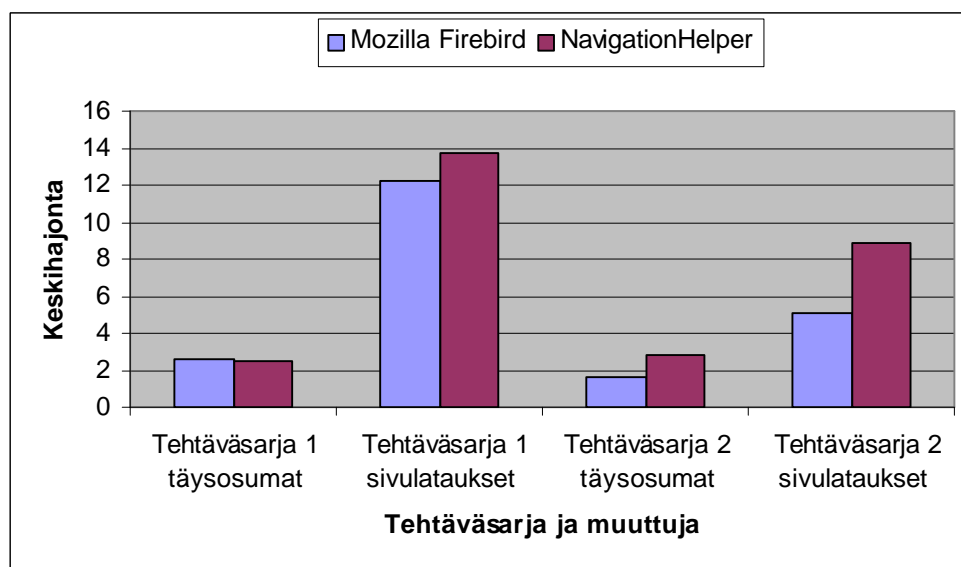
Kuva 6.4. Täysosumat paluutehtävien kohdalla eri selaimilla.

Kaikissa tehtävissä jako ei kuitenkaan ole näin selkeä. Esimerkiksi toisen tehtäväsarjan tehtävässä kymmenen vain yksi Mozillan käyttäjä on suorittanut tehtävän täysosuman muodossa, mutta kuitenkin keskimäärin tehtävään on käytetty vain kaksi sivulatausta. Tämä kertoo siitä, että käyttäjät ovat siirtyneet sivulle selailuhistorian avulla, mutta eivät ole heti onnistuneet löytämään oikeaa sivua.

Tehtävien suorituksessa testihenkilöiden välinen hajonta oli keskimäärin suurempaa NavigationHelper-sovelluksen kohdalla (Kuva 6.5). Ainoastaan ensimmäisen tehtäväsarjan täysosumien keksihajonta testihenkilöiden välillä on yhtä suuri kummankin käyttöliittymän kohdalla. Ensimmäisen tehtävä-

sarjan sivulatausten sekä toisen tehtäväsarjan täysosumien ja sivulatausten suhteen keskihajonta on suurempaa NavigationHelper-sovelluksen kohdalla.

Tämä kertoo ehkä siitä, että suurin osa testihenkilöistä ymmärsi Mozillan käyttöliittymän samalla tavalla ja myös käytti sitä samalla tavalla. Sen sijaan NavigationHelper-sovelluksen kohdalla testihenkilöiden suoritukset poikkesivat enemmän toisistaan. Tämä kertoo siitä, että osa testihenkilöistä ehkä mielsi käyttöliittymän eri tavalla ja onnistui hyödyntämään sitä eri tavoin kuin toiset.

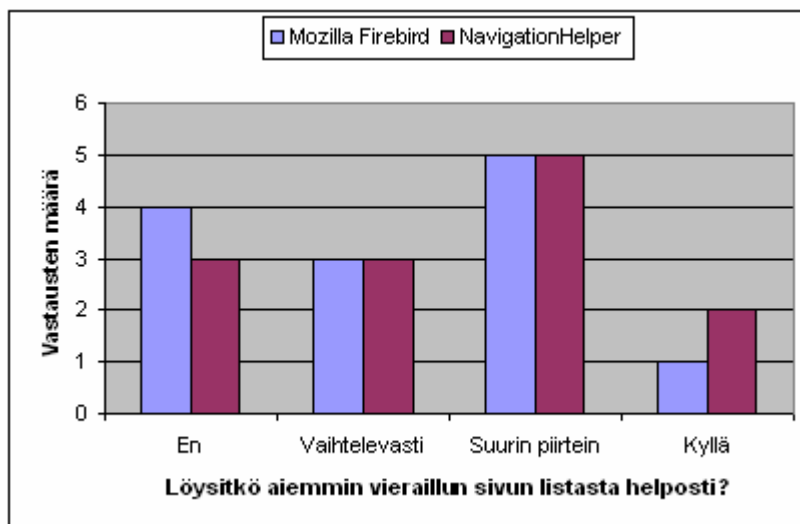


Kuva 6.5. Käyttäjien suoritusten keskihajonta selaimen ja tehtäväsarjan suhteen.

6.4. Koehenkilöiden mielikuvat testistä ja käyttöliittymistä

Testin jälkeen testihenkilöiltä tiedusteltiin mielikuvia testissä käytetyistä käyttöliittymistä (Liite 2). Henkilöitä pyydettiin arvioimaan, kuinka helposti he olivat löytäneet aiemmin vierailut sivut selailuhistoriaa käyttämällä. Käyttäjät arvioivat löytäneensä sivut asteikolla hyvin, suurin piirtein, vaihtelevasti, tai ei ollenkaan (Kuva 6.1).

Vastausten perusteella testihenkilöt kokivat löytävänsä WWW-sivut NavigationHelper-sovelluksen avulla hieman helpommin, joskin ero Mozillan on hyvin pieni. Yhteensä neljä testihenkilöä ilmoitti löytäneensä sivut paremmin NavigationHelper-sovelluksen avulla. Seitsemän testihenkilöä koki löytäneensä sivut kummallakin käyttöliittymällä yhtä hyvin. Kaksi testihenkilöä ilmoitti löytävänsä sivut helpommin Mozillan selailuhistorian avulla.



Kuva 6.1. Käyttäjien vastaukset käyttöliittymien tehokkuudesta.

Sivujen löydettävyyden lisäksi testihenkilöitä pyydettiin kertomaan mielipide käyttöliittymistä. Monet kokivat listojen käytön hankalaksi, koska listojen koko suureni nopeasti selailun edetessä. Testihenkilöillä ei tuntunut olevan selvyyttä siitä, jäivätkö kaikki sivut listoihin. Osa testihenkilöistä koki kaikkien sivujen jäämisen listaan ongelmaksi, kun taas osan mielestä listasta hävisi sivuja. Kummassakin listassa kaikki vierailut sivut löytyivät niistä – NavigationHelper-sovelluksessa joskus myös useasta eri kohdasta.

Monen testihenkilön mielestä paluutehtävien suorittaminen visuaalista muistia hyödyntäen WWW-sivuilla kulkien oli helpompaa kuin selailuhistorian käyttö. Testihenkilöt kokivat, että sivujen tekstimuotoinen esitys vastasi huonosti varsinaisten sivujen sisältöä ja listan läpikäyminen oli puuduttavaa. Kuten aikaisemmissa tutkimuksissa, myös näiden testien perusteella on todettava, että sivujen esittäminen tekstin avulla on monessa suhteessa epätyytyttävä ratkaisu.

NavigationHelper-sovelluksesta heräsi monella testihenkilöllä kysymyksiä ja mielipiteitä. Useat kokivat sivujen hierarkkisen järjestymisen hyödylliseksi ja kertoivat sen helpottavan sivujen löytymisessä. Hierarkian järjestymisen oli kuitenkin kaikille testihenkilöille epäselvää. Osa testihenkilöistä ei kiinnittänyt tähän huomiota, mutta osa piti huonona ominaisuutena sitä, että hierarkian rakenne ei vastannut WWW-sivuston rakennetta.

Samana sivun esiintyminen useaan kertaan NavigationHelper-sovelluksen listassa häiritsi osaa testihenkilöistä. Listan koettiin kasvavan liian suureksi, koska se saattoi sisältää saman sivun moneen kertaan. Jotkut testihenkilöistä kuitenkin osasivat hyödyntää tehtävien tekemisessä sitä listan ominaisuutta,

joka näyttää korostetusti moneen kertaan esiintyvät sivut käyttäjän saapuessa sivulle, missä hän on jo aiemmin käynyt.

Jotkut testihenkilöistä myös ihmettelivät NavigationHelper-sovelluksessa käytettyjen kuvakkeiden tarkoitusta. Kuvakkeiden merkitys jäi hämäräksi eikä kukaan testihenkilöistä osannut sanoa, minkä logiikan mukaan kuvakkeet ilmestyvät sivujen yhteyteen. Tämä on ymmärrettävä, koska listan toimintalogiikkaa ei kerrottu käyttäjille eikä kuvakkeilla voinut ollakaan mitään erityistä merkitystä käyttäjien kannalta. Mikäli kuvakkeiden merkitys olisi selvitetty testihenkilöille olisivat henkilöt mahdollisesti osanneet hyödyntää käyttöliittymää hieman paremmin. Kuvakkeiden avulla testihenkilöt olisivat esimerkiksi pystyneet erottamaan useasti vierailemansa sivut kerran vierailuista. Kuvakkeiden antama informaatio oli kuitenkin vähäistä nykyisellä toteutuksella, kuten edellisessä luvussa mainittiin.

Testihenkilöistä yksitoista oli sitä mieltä, että selailuhistorian esittämiseen selaimissa tulisi tulevaisuudessa olla paremmat työkalut. Selailuhistorian kehittäminen tällä hetkellä koetaan hyödylliseksi, vaikka useimmat testihenkilöt eivät sitä nykyään omassa selailussaan hyödynnä. Selailuhistorian kehittämiseen on siis olemassa selkeä tarve. Vaikka useimmat testihenkilöistä toivoivat selaimiin parempia selailuhistorioita ei heillä kuitenkaan ollut selvää kuvaa siitä, minkälainen toimivan selailuhistorian tulisi olla.

Lopuksi testihenkilöiltä kysyttiin, miten hyvin he kokivat testitehtävien vastaavan normaalia selailua. Suurin osa eli yhdeksän henkilöä kertoi tehtävien vastanneen melko hyvin normaalia selailua. Kolme henkilöä kertoi tehtävien melko huonosta vastaavuudesta ja yksi henkilö piti tehtäviä hyvin normaalia selailua vastaavina.

6.5. Testien arviointia

WWW-ympäristöön tarkoitettujen työkalujen testaaminen käyttäjillä on aina hankalaa, koska ympäristö koostuu niin monesta eri palasesta ja kontrolloitavia muuttujia on paljon. Ympäristössä on vaikeaa tutkia juuri tietyn osan toimintaa ilman, että muut ympäristön osat vaikuttaisivat tuloksiin merkittävästi. Edellä kuvatussa testissä suurimpien ongelmien takana olivat käyttäjien erilaiset tavat liikkua WWW-sivuilla sekä testissä käytetyt testitehtävät. Käyttäjien suuri eroavaisuus siinä, miten ja kuinka paljon he hyödyntävät selainten tarjoamia ominaisuuksia aiheutti myös ongelmia.

Selailuhistorian päivittymisen nopeus oli testeissä tärkein tekijä vertailtavan selaimen valinnassa. Testitehtävien luonteen takia selailuhistorian täytyi pysyä koko ajan ajantasaisena selailun kanssa. Esimerkiksi Internet Explorer -selaimen selailuhistoria päivittyi liian hitaasti ja oli myös epätäydellinen, minkä vuoksi se ei ollut täysin vertailukelpoinen NavigationHelper-sovelluksen kanssa.

Käyttäjät käyttävät selaimia eri tavoin ja omaavat erilaisia tapoja liikkua WWW-sivuilla. Kuten jo aiemmin todettiin, tämän testin testihenkilöt jakautuivat selkeästi neljään eri luokkaan sen mukaan, miten he liikkuvat WWW-sivuilla. Toisten käyttäjien normaali tapa käyttää selainta saattoi sopia paremmin yhteen selailuhistorian käytön kanssa kuin toisten. Useimpien käyttäjien täytyi muuttaa käyttäytymistään normaalista poikkeavaksi käyttäessään selailuhistoriaa.

Käyttäjät ovat useasti taipuvaisia säilyttämään opitut tapansa ja saattavat siten olla vastahakoisia muuttamaan käyttäytymistään hetkellisesti. Tämä saattaa vaikuttaa käyttöliittymien käytössä siten, että tuttua käyttöliittymää osataan ja halutaan käyttää mieluummin kuin normaalista poikkeavaa käyttöliittymää. Uuden käyttöliittymän pitäisi tarjota jotain merkittävää parannusta, jotta käyttäjät olisivat valmiita ottamaan sen käyttöön.

On mahdollista, että testistä saadut tulokset käyttöliittymien välisistä eroista kertovat enemmän käyttäjien välisistä eroista kuin käyttöliittymien välisistä eroista. Varsinkin tässä testissä käyttäjien erilaisuus korostui, koska käyttöliittymät olivat toistensa kanssa hyvin samankaltaisia.

Käyttäjien erilaisuuden ohella toinen ongelma syntyy siitä, että testi-tehtävien tekeminen ei välttämättä johda käyttäjien kohdalla selailuun, joka vastaisi selailua normaaliolosuhteissa. Tehtävien tekeminen ei johda välttämättä luonnolliseen selailutapahtumaan, mikä voi vaikuttaa testin tuloksiin.

Tämä ongelma yritettiin välttää testeissä jakamalla tehtävät selailutehtäviin ja paluutehtäviin. Tehtävien jakaminen selailutehtäviin ja paluutehtäviin auttoi erottamaan selailun ja selailuhistorian käytön toisistaan. Paluutehtäviin käytetty aika, sivulataukset ja paluun tarkkuus ovat suorassa yhteydessä siihen, miten hyvin selailuhistoria helpottaa käyttäjää palaamaan aiemmin vierailemalleen sivulle. Testin tulosten kannalta muut kuin paluutehtävät eivät olleet oleellisia. Näiden tehtävien tarkoituksena oli vain varmistaa, että käyttäjä on käynyt tietyllä sivulla ja että sivu on siten esitettyä myös selailuhistoriassa.

Toinen ongelma tehtävien kohdalla syntyy siitä, että testihenkilöt eivät aina sisäistä tehtäviä halutulla tavalla. On eri asia hakea tietoa WWW-sivustolta annetun tehtävän perusteella kuin suorittaa tiedonhaku alusta alkaen itse. Kun käyttäjä on itse hahmottanut tiedon tarpeen, valinnut menetelmän tiedon hakemiseen ja suorittanut näin tiedonhaun alusta alkaen itse, on toiminta ehkä erilaista kuin valmiiksi mietityn tehtävän pohjalta.

7. Testin arviointi selainten jatkokehityksen kannalta

Testiin osallistuneista henkilöistä kolme ilmoitti käyttävänsä selailuhistoriaa aktiivisesti WWW-selailun yhteydessä. Olin yllättynyt, että selailuhistorian käyttäjiä löytyi testihenkilöiden joukosta näinkin paljon, koska edelliset tutkimukset kertovat selailuhistorioiden vähäisestä käytöstä. Esimerkiksi Catledge ja Pitkow [1995] havaitsivat tutkimuksessaan, että 0.1% selailun aikaisista käyttöliittymätapahtumista liittyy selailuhistorian käyttöön. Suurimmalle osalle testihenkilöistä selailuhistoria oli kuitenkin melko vieras työkalu ennen testejä.

Kerrotuista mielipiteistä selailuhistoriaa aiemmin käyttäneet erottuivat siten, että he vertailivat selailuhistorioita aiempiin kokemuksiinsa nähden. Selailuhistoriaa vähemmän käyttäneiden mielipiteet ja kysymykset olivat yleisluontoisempia. Mielipiteissä näkyy siten onneksi kummankin ryhmän mielipide selailuhistorioiden toiminnasta.

Kymmenen testihenkilön mielestä selailuhistorian esittämiseksi tulisi selaimissa olla tulevaisuudessa paremmat työkalut. Myös selailuhistorioita aktiivisemmin käyttäneet käyttäjät ilmoittivat kehityksen tarpeesta. Selailuhistorian esittämiseen käytettyjen työkalujen halutaan siis kehittyvän jatkossa.

Tässä luvussa käydään läpi käyttäjien mielipiteissä ja tehtävien suorituksissa esiin nousseita asioita sen kannalta, miten selailuhistoria voitaisiin tulevaisuudessa esittää paremmin. Selailuhistorian jäsentäminen ja esittäminen olivat kaksi esiin nousutta pääteemaa. Seuraavissa kohdissa käydään tarkemmin läpi jäsentämiseen ja esittämiseen liittyviä parannusehdotuksia. Luvussa neljä näitä molempia tarkasteltiin aikaisempien tutkimusten valossa. Lopuksi esitetään yksi mahdollinen luonnos siitä, minkälainen NavigationHelper-sovelluksen käyttöliittymä voisi olla seuraavassa versiossa ottaen huomioon testihenkilöiden mielipiteet ja tehtävien suoritukset.

7.1. Selailuhistorian jäsentäminen

Kuten luvussa kaksi todettiin, koostuu hypertekstisivuilla liikkuminen erilaisista navigointikuvioista. Sivustosta riippuen kuviot voivat olla renkaita, silmukoita, säteitä tai polkuja. NavigationHelper-sovelluksen toiminnallisuus perustuu näiden kuvioden tunnistamiseen ja esittämiseen selailuhistoriassa peräkkäisessä järjestyksessä.

Kantava ajatus kuvioden perusteella järjestettävässä listassa oli se, miten käyttäjän arveltiin ymmärtävän nämä kuviot yksittäisinä kokonaisuuksina. Näiden kokonaisuuksien esittämille sivuille käyttäjän arveltiin osaavan palata helposti. NavigationHelper-sovelluksessa kuviot esitettiin mahdollisimman yksinkertaisella tavalla hierarkkisen listan muodossa.

Testissä kävi kuitenkin ilmi, että testihenkilöt eivät hyötäneet erityisen paljon selailuhistoriasta, joka oli järjestetty navigointikuvioiden mukaan. Testihenkilöt kyllä ilmoittivat, että listan järjestäminen hierarkiaksi helpottaa listan käyttöä ja etenkin sivujen löytämistä listasta. Testihenkilöt eivät kuitenkaan ymmärtäneet automaattisesti listan järjestysperiaatetta.

Osittainen syy siihen, etteivät testihenkilöt ymmärtäneet järjestysperiaatteen toimintaa, oli myös se, että sivut näytettiin selailuhistoriassa hierarkiana eikä testihenkilöille annettu mitään visuaalisia vihjeitä sivujen suhteista toisiinsa. Käyttäjä pystyisi helpommin ymmärtämään sivujen järjestyksessä käytetyn logiikan, jos selailukuviot esitettäisiin graafisesti käyttäjälle.

Järjestysperiaatetta tärkeämpi huomio oli kuitenkin se, että testihenkilöt eivät paluutehtäviä tehdessään osanneet arvioida kovin tarkasti, mistä kohtaa listaa paluutehtävän sivu saattoi löytyä. Listaa käytiin useasti järjestyksessä läpi ylhäältä alaspäin etsien haluttua sivua. Jos paluutehtävän kohteena olevalla sivulla oli käyty paljon aikaisemmin, ei testihenkilö kuitenkaan osannut automaattisesti lähteä hakemaan sivua tietystä kohdasta selailuhistoriaa.

Testisarjojen paluutehtävät voidaan jakaa kolmeen eri ryhmään sen mukaan, kuinka kauas selailuhistoriassa tehtävän suorittajan pyydetään siirtyvän. Osassa paluutehtäviä testihenkilö siirtyi sivulle, joka oli hyvin tuoreena selailuhistoriassa. Toisissa paluutehtävissä siirtyminen tapahtui hieman kauemmas ja toisissa kauas.

Lähihistoriassa liikkumiseen testihenkilöt käyttivät useasti taakse-painiketta tai WWW-sivujen linkkejä. Siirryttäessä vähän kauemmas käyttäjät etsivät sivua selailuhistoriasta tarkemmin. Sivun etsimiseen käytettiin kohtuullisen paljon aikaa ilmeisesti siitä syystä, että sivun uskottiin löytyvän listasta helposti. Kauemmaksi siirryttäessä sivua etsittiin useasti myös selailuhistoriasta, mutta monen tällaisen tehtävän kohdalla testihenkilöt totesivat paluun tapahtuvan helpommin visuaalista muistia käyttäen ja siirtyvät kohteeseen WWW-sivujen linkkejä pitkin.

Testihenkilöiden toiminnasta paluutehtävien kohdalla tuli useasti ilmi edelliset kolme eri ryhmää. Suhteellisen tuoreille sivuille käyttäjät saattoivat palata taakse-painikkeen avulla tai käyttäen selailuhistoriaa. Mitä kauemmaksi selailuhistoriassa täytyi palata, sitä herkemmin testihenkilöt tuntuivat suorittavan tehtävän sivuston rakenteen perusteella. Tämän perusteella voisi olettaa, että selailuhistoria tulisi suunnitella ottamalla huomioon nämä kolme eri ryhmää. Historiassa kauempana olevat sivut löytyvät helpommin, jos ne on järjestetty selailuhistoriassa sivuston rakenteen mukaiseen hierarkiaan. Lähihistoriassa olevat sivut voidaan taas järjestää lineaarisemmin vierailu-ajankohdan mukaiseen järjestykseen.

Selailukuvioiden mukainen järjestäminen sopisi näiden päätelmien perusteella paluutapahtumiin, jotka eivät sijoitu kovin kauas selailuhistoriassa. Käyttäjät osaavat käyttää tehokkaasti taakse-painiketta ja WWW-sivujen linkkejä kulkemiseen pienellä alueella. Selailukuvioiden visualisointi voisi olla hyödyllistä sillä alueella, jossa taakse-painikkeen tehokas toiminta heikkenee, mutta käyttäjä ei vielä halua tukeutua perusteellisempaan, sivuston rakenteen avulla tapahtuvaan sivujen etsimiseen.

Testihenkilöiden suullinen palaute tehtäviä tehdessä ja testin jälkeen kertoo siitä, kuinka hyvin testihenkilöt muistivat paluutehtäviä tehdessään kulkemansa reitit. Paluutehtävien kohdalla testihenkilöt sanoivat usein, että ”se oli se paikka, mihin minä menin sitä reittiä pitkin” tai ”se oli se toinen paikka”. Tiettyä WWW-sivua hakiessaan hakijat usein miettivät sitä reittiä, jonka he olivat kulkeneet tietylle sivulle päästäkseen. Paluutehtävien kohdalla tätä reittiä yritettiin hahmottaa selailuhistoriasta, mutta yleensä huonoin tuloksin.

7.2. Selailuhistorian esittäminen

Selailuhistorian järjestysperiaatteen lisäksi pitää myös päättää, miten selailuhistoria esitetään selaimen käyttöliittymässä. Testihenkilöistä osa oli halukas sulkemaan selailuhistorian kokonaan selailun ajaksi. Selailuhistorian viemää ylimääräistä tilaa ei yleensä pidetä hyvänä asiana, mikäli selailuhistoria on pidettävä koko ajan auki.

Selainten käyttöliittymissä on yleistynyt taakse-painikkeen kohdalla avautuva lista aiemmin vierailuista sivuista (Kuva 4.2). Tämänkaltaisen ratkaisu voisi sopia hyvin myös perusteellisemmän selailuhistorian esittämiseen. Selailuhistoria on helposti ymmärrettävissä ja nopeatoiminen. Lista pidetään lyhyenä näyttämällä vain uusimmat sivut listassa. Tarpeen tullen avattavalle listalle on myös mahdollista varata enemmän näyttöpinta-alaa, koska käyttäjä keskittyy kerralla kuitenkin vain yhteen näyttöön.

Selailuhistorian viemän tilan lisäksi ongelmaksi koettiin selailuhistorioiden paisuminen tehtävien tekemisen edetessä. Kummankin tehtäväsarjan puolella välissä lista alkoi jo olla niin täynnä, etteivät testihenkilöt viitsineet lähteä etsimään listasta tiettyä sivua. Listan esityksen tulisi siis olla toisaalta sellainen, ettei käyttäjän tarvitse erikseen kaivaa esityksestä esiin tarvitsemiaan sivuja vaan ne löytyisivät silmäilemällä listaa. Toisaalta listalla tulisi olla suhteellisen vähän sivuja, jotta käyttäjä olisi valmis silmäilemään listaa ja etsimään sieltä haluamansa sivun.

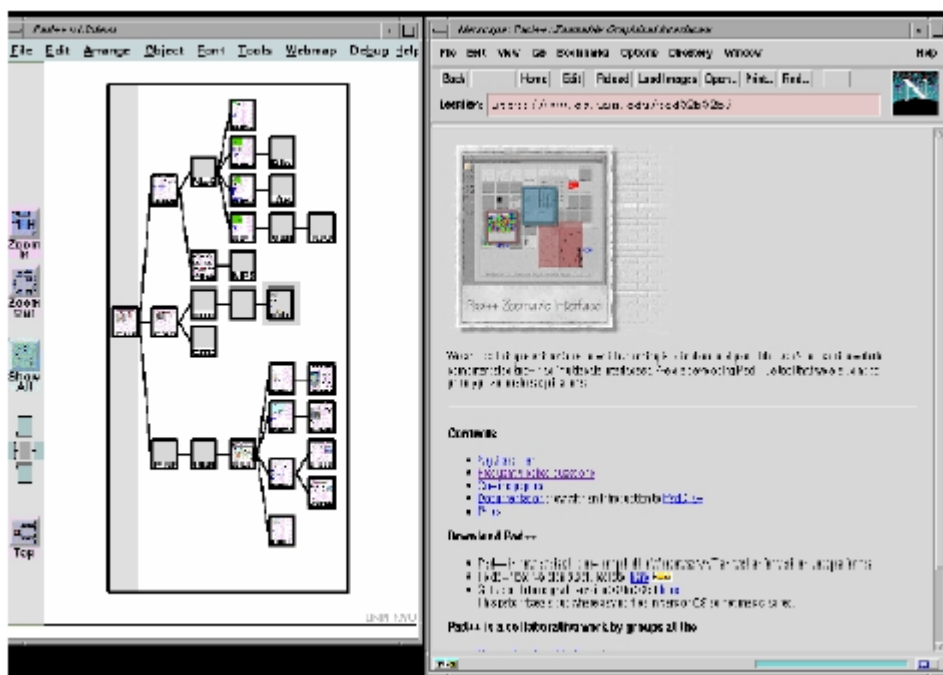
Edellisen ongelman ratkaisemiseksi täytyy keksiä keino selvittää, mitkä sivut eivät kiinnosta käyttäjää selailuhistoriassa. Tällaisia sivuja voivat esimerkiksi olla sivut, joita pitkin käyttäjä kulkee siirtyessään haluamalleen

sivulle. Turhien sivujen karsiminen listasta mahdollistaisi listan pitämisen pienempänä.

Edellisessä kohdassa esitetty kolmiosainen lista on myös yksi ratkaisu listan tila- ja kasvuongelmaan. Selailukuvioita ylläpitävän listan kasvu voitaisiin rajoittaa tiettyyn mittaun, jonka jälkeen kuvat tippuisivat listasta pois ja löytyisivät tarvittaessa sivuston rakenteen perusteella järjestetystä listasta. Tällä tavalla toimivassa selailuhistoriassa käyttäjä voisi käyttää taakse-painiketta lähihistoriassa siirtymiseen, hieman kauemmaksi käyttäjä voisi siirtyä käyttäen selailukuvioiden perusteella järjestettyä listaa ja edelleen selailuhistoriassa kauaksi siirtymiseen käyttäjä voisi käyttää sivuston rakenteen perusteella järjestettyä listaa.

Selailuhistorian esittäminen osana selainta selailutapahtuman aikana ja selailuhistorian tila- ja kasvuongelmat ovat ratkaistavissa suunnittelemalla erilaisia prototyyppejä vaihtoehdoista ja valitsemalla parhaiten toimiva. Sen sijaan suurempi ongelma on sivujen esittäminen selailuhistoriassa. Mikäli sivut esitetään tekstinä, täytyy olla jokin keino nimetä sivut yhtenäisesti. Sematic web voisi tulevaisuudessa olla yksi ratkaisu, jonka avulla sivuista saadaan kuvaavampia tekstimuotoisia esityksiä.

Aiemmin luvussa neljä esiteltiin Cockburnin ja Greenbergin [1999] (Kuva 4.3) ehdotus sivujen esittämiseksi pienten kuvakkeiden avulla. Kuvakkeiden käyttö tekstin ohella onkin varteenotettava vaihtoehto. Kuvakkeiden ja tekstin käytön ohella voisi olla hyödyllistä miettiä myös sivujen suhteiden esittämistä graafisesti selkeämmin kuin NavigationHelper- tai WebView-sovelluksessa.



Kuva 7.1. PadPrints-sovelluksen selailuhistoria [Hightower et al., 1998].

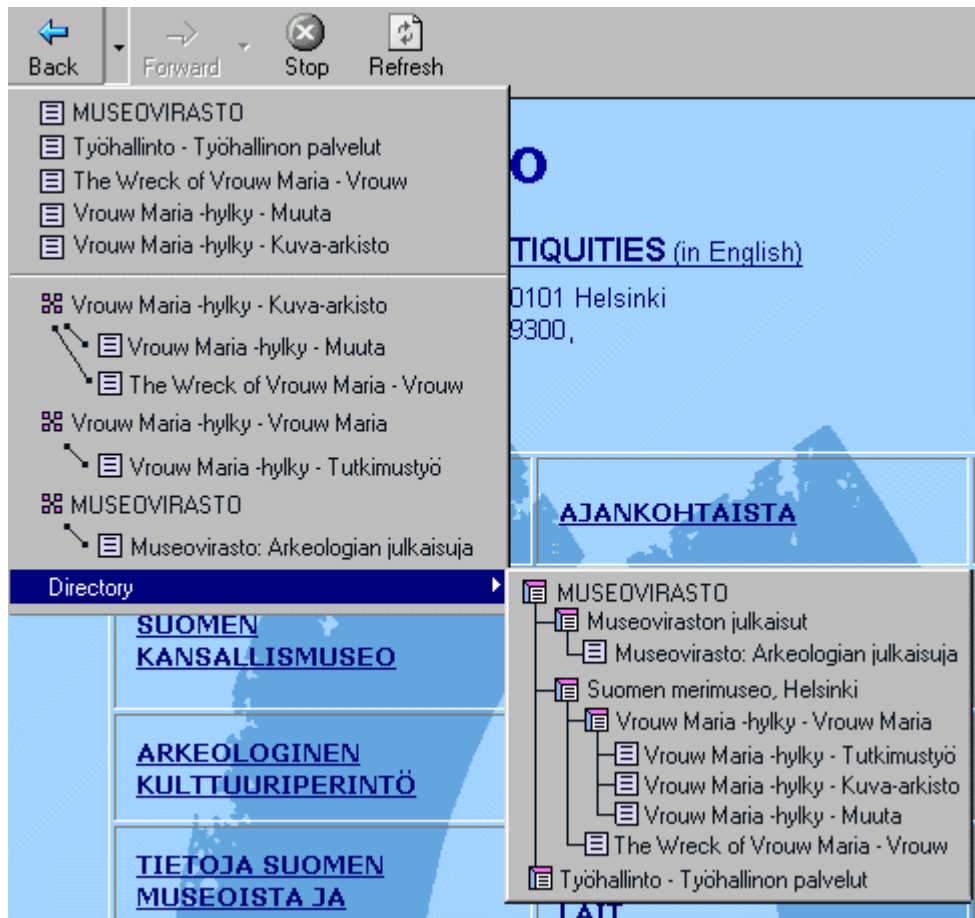
PadPrints-sovellus (Kuva 7.1) edustaa toista ääripäätä edellisiin selailuhistorioihin verrattuna, koska siinä selailuhistoria ja historiassa esiintyvät sivut esitetään täysin grafiikan avulla. Kuten aiemmin mainittiin, kaipasivat testihenkilöt kauemmas selailuhistoriassa palatessaan tietoa sivuston rakenteesta. PadPrints esittää sivuston rakenteen siten, että sivuille pystyy siirtymään sivuston rakenteen perusteella. PadPrints kuvaa selailuhistorian yhden näkymän avulla, jonka esitystapaa käyttäjä voi vaihdella. Ongelmana esitysmuodossa on näytettävän rakenteen paisuminen selailun edetessä [Hightower et al., 1998].

Tehokkaassa selailuhistorian esityksessä PadPrints-sovelluksen kaltaista esitystä ja minimaalisempaa NavigationHelper- ja WebView-sovellusten käyttämää tekstimuotoista esitystä ja rakennetta voitaisiin yhdistää tilanteen mukaan.

7.3. Kolmiosaisen selailuhistorian hahmotelma

Vertailtaessa NavigationHelper-sovellusta ja Mozilla Firebird -selainta testeissä syntyi testihenkilöiden palautteen ja käyttäytymisen kautta uusia ideoita ja vaatimuksia selailuhistorian toiminnallisuudelle. Kuvassa 7.2 on esitetty hahmotelma käyttöliittymästä, jossa edellisissä kohdissa mainitut vaatimukset on otettu huomioon. Kuva esittää selailuhistoriaa sen jälkeen, kun käyttäjä on liikkunut Museoviraston sivuilla kuvan 5.5 mukaisessa järjestyksessä.

Käyttöliittymässä selailuhistoria olisi jaettu kolmeen eri tavoin toimivaan osaan. Ensimmäisessä osassa käyttäjälle tarjotaan näkymä selailun lähihistoriaan. Tässä osassa käyttäjälle näytettäisiin viimeksi vierailut sivut niiden vierailujärjestyksessä. Esitys vastaisi nykyisistä selaimista löytyvää listaa viimeksi vierailuista sivuista. Toisessa osassa käyttäjälle näytettäisiin selailuhistoriassa kauempana sijaitsevat sivut. Tämä lista olisi järjestetty selailukuvioden mukaan. Listassa selailuhistoria tulisi esittää niin, että käyttäjä pystyy yhdistämään esityksen niihin reitteihin, joita pitkin hän on kulkenut WWW-sivustolla ja siten palaamaan helposti aiemmin vieraillemilleen sivuille. Selailuhistorian kolmannessa osassa WWW-sivut olisivat esitettynä niiden loogisen rakenteen mukaisessa hierarkiassa. Tämän esityksen avulla käyttäjä pystyisi palaamaan sivuille sivuston rakenteen perusteella.



Kuva 7.2. Kolmiosaisen selailuhistorian mahdollinen käyttöliittymä.

8. Päätelmät

Hypertekstirakenteen visualisointi on herättänyt kiinnostusta siitä lähtien, kun hyperteksti keksittiin ja siitä tuli yksi keino tiedon esittämiseen. Laajempaa tutkimusta aiheen parissa ryhdyttiin tekemään, kun World Wide Web saavutti enemmän suosiota ja kasvoi kiihtyvällä vauhdilla. Tutkimuksilla on selvitetty muun muassa sitä, miten WWW-sivuilla liikkuvat ihmiset käyttäytyvät, miten he hahmottavat hypertekstin avulla esitetyn materiaalin, sekä miten WWW:n rakenne pystytään visualisoimaan ja sitä kautta helpottamaan WWW-ympäristön käyttöä.

Mukherjea ja Foley [1995] esittivät ensimmäisten joukossa erilaisia tapoja visualisoida WWW-ympäristön rakennetta. Edelleen Catledge ja Pitkow [1995] selvittivät ihmisten liikkumista WWW-ympäristössä ja Cockburn ja Jones [1996] selvittivät, miten selaimet toimivat käyttäjien kannalta. Näiden tutkimusten pohjalta on myöhemmin kiinnitetty enemmän huomiota siihen, miten WWW-ympäristön käyttöä voidaan kehittää käyttäjäystävällisempään suuntaan eri tutkimusten tuloksia soveltamalla.

Tässä tutkimuksessa lähdettiin liikkeelle tarkastelemalla ihmisten liikkumista WWW-ympäristössä. Käyttäjien liikkeitä hypertekstisivuilla tarkasteltiin sivujen rakenteen ja ajallisen järjestymisen mukaan. Edelleen kiinnitettiin huomiota erilaisiin tapoihin visualisoida käyttäjän kannalta oleelliset asiat WWW-ympäristöstä. Tutkimuksen tarkoituksena oli toisaalta selvittää, mitä erilaisia tapoja selailun helpottamiseen on kehitelty ja toisaalta miettiä selailun helpottamiseen aiemmista toteutuksista poikkeavaa ratkaisua.

Sivujen järjestämistä käyttäjien liikkeiden mukaisesti on tarkasteltu myös muissa tutkimuksissa, kuten WebWiew- ja PadPrints-sovellusten kohdalla. NavigationHelper-sovelluksen pohjaksi otettiin selailuhistoria, joka järjestetään käyttäjän liikkeiden mukaan. Tämän lisäksi selailuhistorian järjestämisessä otettiin huomioon ajallinen järjestäminen. NavigationHelper-sovellus tarjoaa yksinkertaisen toteutuksen, jossa käyttäjän selailuhistoria on jäsennetty käyttäjän liikkeiden mukaan aikajärjestykseen.

Käyttöliittymän toimivuutta testattiin testihenkilöillä, jotka tekivät tehtäväsarjoja NavigationHelper- ja Mozilla Firebird -sovelluksilla. Vertailun tarkoituksena oli selvittää, miten käyttäjät kokevat selailukuvioiden mukaan järjestetyn selailuhistorian käytön. Tutkimusten tulosten perusteella nähtäisiin, hyötyvätkö käyttäjät erityisesti uudesta selailuhistoriasta ja miten testihenkilöt kokevat uuden käyttöliittymän.

Testien tuloksena havaittiin, että selkeästi havaittavissa olevaa eroa ei käyttöliittymien välillä ollut. Käyttöliittymät erosivat toisistaan kuitenkin

tehtäväsarjojen ja tehtävien kohdalla jonkin verran. Tehtäväsarjojen tekemisen systemaattinen tarkastelu ei tuonut esiin mitään suurempia eroja käyttöliittymien välillä. Tutkimuksessa saatiin kuitenkin paljon tietoa siitä, mitä mieltä käyttäjät olivat selailuhistoriasta ja kuinka käyttäjät käyttivät selailuhistoriaa. Myös nykyisten selainten ominaisuuksien käytöstä saatiin uutta tietoa. Testien aikana huomattiin myös se, miten ongelmallista normaalin selailun toteuttaminen laboratorio-olosuhteissa on. Testien yksi päähavainto on, että käyttäjät eroavat WWW-ympäristön käytön suhteen suuresti toisistaan. Ympäristö tarjoaa monia eri tapoja sivujen välillä liikkumiseen ja monilla testihenkilöillä olikin erilaiset tavat liikkua ympäristössä.

Testeistä saadun palautteen avulla hahmoteltiin vaatimuksia paremman selailuhistorian toteutukselle. Testintehtävien suoritusten analysointi antoi monenlaista tietoa, jonka avulla uuden käyttöliittymän vaatimuksia saattoi hahmotella. Tärkeimpiä havaintoja oli se, että tekstimuotoinen esitys sivuista ei ajoittain toiminut ollenkaan. Tämä on havaittu myös aiemmissa tutkimuksissa. Myöskään selailukuvioiden esittäminen ajallisesti järjestettynä ei toiminut toivotulla tavalla ja sai testihenkilöiltä myös negatiivista palautetta. Tutkimuksen suunta oli siinä mielessä kuitenkin oikea, että käyttäjät ilmaisivat tarvitsevansa tulevaisuudessa selailuhistorian esittämiseen parempia työkaluja.

Tällä hetkellä olemassa olevia selailuhistorian esittäviä sovelluksia on olemassa paljon. Myös muita navigoinnin helpottamiseen tarkoitettuja sovelluksia on tarjolla. Näistä osa on suoraan käytettävissä nykyisten selainten yhteydessä ja osa on käytettävissä siinä ympäristössä, johon ne on suunniteltu. Ero tutkimusten tuloksena syntyneiden sovellusten ja nykyisten selainten välillä on suuri. Tulevaisuudessa selainten käyttäjien mieltymykset määräävät sen, syntykö selailuhistoriasta nykyistä tärkeämpi ja näkyvämpi osa selainta. Erilaiset mallitoteutukset, kuten esimerkiksi WebView, PadPrints ja NavigationHelper, tarjoavat esimerkkejä mahdollisista tulevaisuudessa käytettävistä selailuhistorioista.

Viiteluettelo

- [Byrne et al., 1999] Michael D. Byrne, Bonnie E. John, Neil S. Wehrle, David C. Crow, The tangled web we wove: a taskonomy of WWW use. In: *Proc of CHI 99: Conference on Human Factors in Computing Systems*. 544 - 551, 1999.
- [Catledge and Pitkow, 1995] Lara D. Catledge and James E. Pitkow, Characterizing browsing strategies in the World-Wide Web, *Computer Networks and ISDN Systems* **27**, 6 (Apr. 1995), 1065 – 1073.
- [Cava et al., 2002] Ricardo A. Cava, Paulo R.G. Luzzardi and Carla M.D.S. Freitas, The bifocal tree: a technique for the visualization of hierarchical information structures. In: *Proc of IHC 02: Symposium on Human Factors in Computer Systems.*, 2002.
- [Chen, 1998] Chaomei Chen, Bridging the gap: the use of pathfinder networks in visual navigation. *Journal of Visual Languages and Computing* **9** (1998), 267 – 286.
- [Chi, 2000] Ed H. Chi, A taxonomy of visualization techniques using the data state reference model. In: *Proc of the IEEE Symposium on Information Visualization*. 69-75, 2000.
- [Chi and Riedl, 1998] Ed H. Chi and John. T. Riedl, An operator interaction framework for visualization systems. In: *Proc of InfoVis 98: Symposium on Information Visualization*. 63 – 70, 1998.
- [Cockburn and Greenberg, 1999] Andy Cockburn and Saul Greenberg, Issues of page representation and organisation in web browser's revisitaton tools, *Australian Journal of Information Systems* **7**, 2 (2000), 120 - 127.
- [Cockburn et al., 1999] Andy Cockburn, Saul Greenberg, Bruce McKenzie, Michael Janonsmith and Shaun Kaasten, Webview: a graphical aid for revisiting web pages. In: *Proc of OZCHI 99: the Australian Conference on Human-Computer Interaction*. 15 - 22, 1999.
- [Cockburn and Jones, 1996] Andy Cockburn and Steve Jones, Which way now? analysing and easing inadequacies in WWW navigation, *International Journal of Human-Computer Studies* **45** (1996), 105 – 129.
- [Cockburn and Jones, 1997] Andy Cockburn and Steve Jones, Design issues for World Wide Web navigation visualisation tools. In: *Proc of RIAO 97: The Fifth Conference on Computer-Assisted Research of Information*. 55 - 74, 1997.
- [Cockburn and McKenzie, 2000] Andy Cockburn and Bruce McKenzie, What do web users do? An empirical analysis of web use, *International Journal of Human-Computer Studies* **54**, 6 (2000), 903 – 922.
- [Cockburn et al., 2002] Andy Cockburn, Bruce S. McKenzie and Michael JasonSmith, Pushing back: evaluating a new behaviour for the back and

forward buttons in web browsers, *International Journal of Human-Computer Studies* **57**, 5 (2002), 397 – 414.

- [Dieberger and Frank, 1998] Andreas Dieberger and Andrew U. Frank, A city metaphor for supporting navigation in complex information spaces, *Journal of Visual Languages and Computing* **9** (1998), 597 - 622.
- [Farris et al., 2002] J. Shawn Farris, Keith S. Jones, Peter D. Elgin, Users' schemata of hypermedia: what is so 'spatial' about a website?, *Interacting with Computers* **14** (2002), 487 – 502.
- [Freitas et al., 2002] Carla M. D. S. Freitas, Paulo R. G. Luzzardi, Ricardo A. Cava, Marco A. A. Winckler, Marcelo S. Pimenta and Luciana P. Nedel, Evaluating usability of information visualization techniques. In: *Proc of IHC2002*. 2002.
- [Greenberg and Cockburn, 1999] Saul Greenberg and Andy Cockburn, Getting back to back: alternate behaviors for a web browser's back button. In: *Proc of the 5th Annual Human Factors and the Web Conference*. 1999.
- [Gunnarsson, 1996] Gunnar Gunnarsson, *Internetin Tekniikka*. Pagina, 1996.
- [Head et al., 2000] Milena Head, Norm Archer and Yufei Yuan, World wide web navigation aid, *International Journal of Human-Computer Studies* **53** (2000), 301 - 330.
- [Hightower et al., 1998] Ron R. Hightower, Laura T. Ring, Jonathan I. Helfman, Benjamin B. Bederson and James D. Hollan, Graphical multiscale web histories: a study of PadPrints. In: *Proc of Hypertext 98: the 9th ACM Conference on Hypertext and Hypermedia*. 58 - 65, 1998.
- [Levene and Loizou, 2002] Mark Levene and George Loizou, Web interaction and the navigation problem in hypertext. In: A. Kent, J.G. Williams and C.M. Hall (eds.), *Encyclopedia of Microcomputers*, 2002, 381 - 398.
- [Mackinlay et al., 1991] Jock D. Mackinlay, George G. Robertson and Stuart K Card, Perspective wall: detail and context smoothly integrated. In: *Proc of CHI 91*. 173 - 179, 1991.
- [McDonald and Stevenson, 1998] Sharon McDonald and Rosemary J. Stevenson, Navigation in hyperspace: An evaluation of the effects of navigational tools and subject matter expertise on browsing and information retrieval in hypertext, *Interacting with Computers* **10** (1998), 129 - 142.
- [Modjeska and Marsh, 1997] David Modjeska and Anna Marsh, Structure and Memorability of web sites. University of Toronto, Computer Systems Research Group, Technical report, 1997.

- [Mukherjea and Foley, 1995] Sougata Mukherjea and James D. Foley, Visualizing the world-wide web with the navigational view builder, *Computer Networks and ISDN Systems* **27**, 6 (Apr.1995), 1075 - 1087.
- [Mukherjea and Hara, 1997] Sougata Mukherjea and Yoshinori Hara, Focus+context views of world-wide web nodes. In: *Proc of the Eighth ACM Conference on Hypertext*, 187 - 196, 1997.
- [Mullier et al., 2002] Duncan J. Mullier, David J. Hobbs and David J. Moore, Identifying and using hypermedia browsing patterns, *Journal of Educational Multimedia and Hypermedia* **11**, 1 (2002), 31 - 50.
- [Nation et al., 1997] David A. Nation, Catherine Plaisant, Gary Marchionini and Anita Komlodi, Visualizing websites using a hierarchical table of contents browser: WebTOC. In: *Proc. of the 3rd Conference on Human Factors and the Web*, 1997.
- [Nielsen, 1994] Jacob Nielsen, Heuristic evaluation. In: Jacob Nielsen and Robert L. Mack (eds.), *Usability Inspection Methods*, John Wiley & Sons, 1994.
- [Nielsen, 1999] Jakob Nielsen, User interface directions for the web. *Communications of the ACM* **42** (1999), 65- 72.
- [Nilsson and Mayer, 2002] Rachel M. Nilsson and Richard E. Mayer, The effects of graphic organizers giving cues to the structure of a hypertext document on users' navigation strategies and performance, *International Journal of Human-Computer Studies* **57** (2002), 1 - 26.
- [Preece et al., 2002] Jennifer Preece, Yvonne Rogers and Helen Sharp, *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, 2002.
- [Spence, 1999] Robert Spence, A Framework for Navigation, *International Journal of Human-Computer Studies* **51** (1999), 919-945.
- [Spence, 2001] Robert Spence, *Information Visualization*. Addison-Wesley, 2001.
- [Tauscher and Greenberg, 1997] Linda Tauscher and Saul Greenberg, Revisitation patterns in world wide web navigation. In: *Proc. of the ACM SIGCHI 97 Conference on Human Factors in Computing Systems*, 22 - 27, 1997.
- [Weinreich and Lamersdorf, 2000] Harald Weinreich and Winfried Lamersdorf, Concepts for improved visualization of web link attributes, *Computer Networks* **33** (2000), 403 – 416.

Liite 1

Koehenkilö: _____

Taustatietolomake**1. Millaisena WWW-selainten käyttäjänä pidät itseäsi?**

- a) Peruskäyttäjä
- b) Käytän myös joitain erikoisempia toimintoja
- c) Räättälöin selaimen ulkoasun ja toiminnan mieleisekseni

2. Oletko käynyt Museoviraston WWW-sivuilla (www.nba.fi)?

- a) Kyllä
- b) En

3. Oletko käynyt Työministeriön WWW-sivuilla (www.mol.fi)?

- a) Kyllä
- b) En

4. Kuinka usein vieraillet WWW:ssä?

- a) Useita kertoja päivässä
- b) Päivittäin
- c) Useita kertoja viikossa
- d) Harvemmin

5. Käytätkö hyväksi selainten tarjoamaa listaa aiemmin vierailtuista sivuista?

- a) Useasti
- b) Joskus
- c) En ollenkaan
- d) En ole kuullutkaan

6. Selainten käyttäjillä on erilaisia tapoja järjestää selailutapahtuma. Valitse seuraavista ne, jotka sopivat omiin tapoihisi.

- a) Pidän yleensä vain yhtä selaimen ikkunaa auki.
- b) Pidän monta eri selaimen ikkunaa auki ja tiedän kullakin hetkellä, mikä sivu on missäkin ikkunassa avoinna.
- c) Pidän monta sivua avoinna, mutta ei aina tiedä mikä sivu on missäkin paikassa.
- d) Lähdän usein harhailemaan eri sivuilla, enkä muista alkuperäistä selailun tarkoitusta.
- e) Liikun WWW-sivuilla usein järjestelmällisesti.
- f) Siirryn WWW-sivuille usein kirjamerkkien avulla.
- g) Koen WWW-ympäristön käytön mukavaksi.
- h) Hukkaan joskus WWW-sivuja selailun yhteydessä, enkä löydä niitä jälkeempään.

Liite 2

Koehenkilö: _____

Mielikuvat testistä**1. Löysitkö Mozillan listasta aiemmin vierailemasi sivut helposti?**

- a) Kyllä b) Suurin piirtein c) Vaihtelevasti d) En

2. Herättikö Mozillan lista kysymyksiä, ajatuksia tai mielipiteitä?**3. Löysitkö NavigationHelperin listasta aiemmin vierailemasi sivut helposti?**

- a) Kyllä b) Suurin piirtein c) Vaihtelevasti d) En

4. Herättikö tämä NavigationHelperin lista kysymyksiä, ajatuksia tai mielipiteitä?**5. Pitäisikö selaimissa tulevaisuudessa olla paremmat työkalut selailuhistorian esittämiseen käyttäjälle?**

- a) Kyllä
b) Ei

6. Vastasivatko tehtävät mielestäsi normaalia WWW-surffausta?

- a) Hyvin b) Melko hyvin c) Melko huonosti d) Huonosti

Liite 3

Tehtäväsarja I:

Suomen museoviraston kotisivu: <http://www.nba.fi/>

Työministeriön kotisivu: <http://www.mol.fi/>

1. Siirry Museoviraston sivujen osioon, jossa esitellään Suomen kansallismuseota. Siirry edelleen sivulle, jossa kerrotaan tilojen vuokrauksesta.
2. Löytyykö vastaavaa, tilojen vuokrausta käsittelevää sivua, Suomen kansallismuseon englanninkielisiltä sivuilta.
3. Onko kummallakin sivulla mainittu sama pääsymaksu aikuisille?
4. Etsi museoviraston sivuilta "Kirjasto- ja arkistopalvelut" -osiosta asiakirjatiedot Urajärven kartanosta. Asiakirjatiedot löytyvät "Muut asiakirja-arkistot" ja edelleen "Kirjaston arkisto" -osiosta.
5. Museoviraston sivuilta löytyy myös sivu, joka esittelee tarkemmin Urajärven kartanomuseota. Etsi kartanoa esittelevä sivu ja näiltä sivuilta edelleen kartanon historiasta kertova sivu.
6. Urajärven kartanon historiasta kertovan sivun viimeisessä kappaleessa mainitaan lahjakirja. Onko lahjakirja mainittu asiakirja-arkistossa?
7. Siirry takaisin sivulle, jossa oli kerrottu Suomen kansallismuseon tilojen vuokrauksesta.
8. Mikä on Urajärven kartanomuseon pääsymaksu aikuisilta?
9. Työministeriön sivuilla on osio, jossa käsitellään maahanmuuttoasioita. Etsi tältä sivulta ajankohtaiset asiat -osio.
10. Suomen kansallismuseossa on vuoden loppuun asti avoinna eräs vaihtuva näyttely. Mikä on näyttelyn nimi?
11. Onko näyttely mainittu Suomen kansallismuseon englanninkielisellä sivulla?
12. Siirry merimuseon pääsymaksuista kertovalle sivulle.

13. Palaa sivulle, jossa oli lueteltuna Urajärven kartanoon liittyvät arkistoidut paperit.
14. Etsi työministeriön sivuilta myyntiedustustyön avoimet työpaikat. Työpaikat löytyvät kaupallisen työn kategoriasta.
15. Etsi myös linja- ja henkilöautonkuljettajille tarjolla olevat avoimet työpaikat. Työpaikat löytyvät kuljetus- ja liikennetyö -kategoriasta.
16. Aiemmin, tehtävässä neljä, etsit asiakirjatietoja Urajärven kartanosta. Asiakirjatiedot löytyivät muut asiakirja-arkistot -osiosta. Sivulla on mainittu myös arkeologinen keskusarkisto. Siirry tämän arkiston sivulle.
17. Linkki edellisessä tehtävässä etsitylle arkeologista keskusarkistoa esittelevälle sivulle löytyy myös Museoviraston sivujen osiosta: Arkeologinen kulttuuriperintö. Etsi linkki.
18. Palaa aiemmin vierailemallesi työministeriön sivulle, jossa oli kerrottu ajankohtaisista maahanmuuttoasioista.
19. Siirry aiemmin vierailemallesi sivulle, jossa oli lueteltuna myyntiedustustyön avoimet työpaikat.
20. Tehtävässä 17 etsit sivun, jossa oli linkki Arkeologista keskusarkistoa esittelevälle sivulle. Palaa takaisin tälle sivulle.
21. Palaa sivulle, jossa on lueteltuina linja- ja henkilöautonkuljettajien avoimet työpaikat.

Tehtäväsarja II:

Suomen museoviraston kotisivu: <http://www.nba.fi/>

Työministeriön kotisivu: <http://www.mol.fi/>

1. Museoviraston sivujen osiosta "Museoviraston muut museot ja linnat" Löytyy kartanomuseo Alikartanoa esittelevä sivu. Siirry sivulle, jossa on mainittu Alikartanon aukioloajat.
2. Siirry vastaavalle englanninkieliselle Alikartanon aukioloajoista kertovalle sivulle.
3. Selvitä mikä on Alikartanon suomenkielisellä sivulla mainittu opastuskierroksen hinta.

4. Museoviraston sivuilta löytyy osio, jossa on lueteltuna museoviraston julkaisut. Etsi sivu, jossa luetellaan arkeologian julkaisut.
5. Museoviraston sivulla on myös osio, jossa kerrotaan arkeologisesta kulttuuriperinnöstä yleensä. Osion nimi on "Arkeologinen kulttuuriperintö". Etsi osion sivuilta Susiluolan tutkimusten historiasta kertova sivu.
6. Museoviraston sivuilla on osio, jossa kerrotaan avustuksista, luvista ja laeista. Mene sivulle, jossa kerrotaan tutkimusluvista.
7. Siirry takaisin sivulle, jossa oli kerrottu Alikartanon aukioloajoista.
8. Siirry Työministeriön www-sivuille (www.mol.fi) ja etsi museotyöntekijöille tarjolla olevat avoimet työpaikat. Älä käytä hakutoimintoa.
9. Etsi myös sosiaalityöntekijöille tarjolla olevat avoimet työpaikat.
10. Palaa sivulle, joka kertoi Susiluolan tutkimusten historiasta.
11. Museoviraston sivuilta löytyy myös Merihistoria-osio. Etsi näiltä sivuilta Vrouw Mariaa käsittelevä sivu ja edelleen sivu, jolla on lueteltuna hyllyltä nostetut esineet.
12. Palaa sivulle, jossa oli kerrottu tutkimusluvista.
13. Etsi museoviraston sivuilta "Kirjasto- ja arkistopalvelut" -osiosta rakennushistorian osaston arkisto. Arkisto löytyy "Muut asiakirja-arkistot" -osiosta.
14. Museoviraston sivuilla on osio, joka käsittelee rakennusperintöä. Etsi myös näiltä sivuilta edellisen tehtävän rakennushistorian osaston arkistosta kertova sivu.
15. Siirry takaisin Työministeriön www-sivuille (www.mol.fi) ja tarkasta, minkä muiden alojen työntekijät ovat museotyöntekijöiden kanssa samassa kategoriassa?
16. Palaa sivulle, jossa oli lueteltuna museoviraston julkaisut
17. Aiemmin tehtävässä 14 kävit rakennusperinnöstä kertovilla sivuilla. Etsi näiltä sivuilta restaurointialan koulutusta käsittelevä sivu.

18. Aiemmin hait sosiaalityöntekijöille tarjolla olevia avoimia työpaikkoja. Siirry takaisin tälle sivulle, jossa avoimet työpaikat ovat lueteltuna.
19. Kuva-arkisto löytyy aiemmin vierailemasi "Kirjasto- ja arkistopalvelut" -osiosta. Etsi historian kuva-arkisto.
20. Palaa sivulle, joka kertoi Vrouw Maria -hylystä nostetuista esineistä.
21. Palaa sivulle, jossa on esiteltynä rakennushistorian osaston arkisto.