

**Janne Heinonen**

# **RIA-verkkosovellusten käytettävyyden asiantuntija-arviointi**

**Tietojärjestelmätieteen  
pro gradu -tutkielma  
30.12.2007**

**Jyväskylän yliopisto  
Tietojenkäsittelytieteiden laitos  
Jyväskylä**

## TIIVISTELMÄ

Heinonen, Janne Tapani

RIA-verkkosovellusten käytettävyyden asiantuntija-arviointi / Janne Heinonen  
Jyväskylä: Jyväskylän yliopisto, 2007.

105 s.

Tietojärjestelmätieteen pro gradu -tutkielma

Tutkielmassa tarkastellaan WWW-verkkosovelluksia ja näihin liittyviä käytettävyysskriteereitä. Tavoitteena on selvittää, kuinka erilaisten verkkosovellusten käytettävyyttä voidaan arvioida asiantuntija-arvioinnin menetelmin.

Aluksi esitellään sekä perinteisten WWW-verkkosovellusten että RIA-verkkosovellusten yleiset ominaispiirteet ja tekniikka toimintamekaniikan tasolla sekä pohditaan erilaisia käytettävyysskriteereitä. Lisäksi käydään määritellen läpi keskeiset käsitteet käytettävyys ja käyttäjäkokemus. Seuraavaksi esitellään käytettävyyden asiantuntija-arvioinnin menetelminä heuristinen arviointi, kognitiivinen läpikäynti sekä automaattiset arviointimenetelmät.

RIA-verkkosovellusten ominaispiirteistä johdetaan käytettävyyteen ja käyttäjäkokemukseen liittyviä vaatimuksia, joista konstruoidaan tutkielman keskeisenä tuloksena tarkastuslista RIA-verkkosovellusten asiantuntija-arvioinnin tueksi.

Tutkielman lähestymistapa on kartoittava sekä osin konstrukttiivinen ja käytäntöön soveltava. Aihepiiriä on käsitelty kirjallisuuden pohjalta. Tulokset soveltuvat hyvin käytännön tietojärjestelmätyötä tekeville.

AVAINSANAT: käytettävyys, käyttäjäkokemus, asiantuntija-arviointi, verkkosovellus, RIA

## **ABSTRACT**

**Heinonen, Janne Tapani**

**Expert evaluation of the RIA web application usability / Janne Heinonen**

**Jyväskylä: University of Jyväskylä, 2007.**

**105 pages**

**Thesis for the degree of Master of Science (Information systems science)**

**This thesis examines web applications and how usability issues and criteria are related to them. The thesis aims to explore how the usability of different web applications can be evaluated through expert usability evaluation methods.**

**Both traditional and rich web applications are considered within the scope of general characteristics and broad technical issues. Essential concepts of usability and user experience are defined and explained. Subsequently expert evaluation methods of the usability are discussed. Cognitive walkthrough, automatic evaluation methods and heuristic evaluation are covered.**

**Additional demands concerning usability and user experience are derived from fundamental characteristics of rich web applications. These demands act as building blocks for the check list to support expert usability evaluation of rich web applications.**

**The approach is explorative and partly constructive. Hands on heuristic evaluation is also introduced. The study has been conducted through literature. Thesis is aimed for people engaged in information systems development.**

**KEYWORDS: usability, user experience, expert evaluation, web application, Rich Internet Application**

# SISÄLLYSLUETTELO

1	JOHDANTO.....	7
2	VERKKOSOVELLUKSET .....	10
	2.1 Verkkosovellusten ominaispiirteitä .....	10
	2.2 Verkkosovellusten toimintaperiaate .....	14
	2.3 Yhteenveto.....	22
3	RIA-VERKKOSOVELLUKSET .....	24
	3.1 Web 2.0 – muuttunut toimintaympäristö .....	24
	3.2 RIA-verkkosovellusten ominaispiirteitä.....	27
	3.3 RIA-verkkosovellusten toimintaperiaate.....	30
	3.4 AJAX .....	35
	3.5 Yhteenveto.....	38
4	VERKKOSOVELLUKSIEN KÄYTETTÄVYYDEN ARVIOINTI .....	40
	4.1 Käytettävyys.....	41
	4.2 Käyttäjäkokemus .....	44
	4.3 Käytettävyyden ja käyttöliittymän arviointiin kehitettyjä menetelmiä.....	48
	4.3.1 Kognitiivinen läpikäynti .....	50
	4.3.2 Automaattinen arviointi.....	53
	4.3.3 Heuristinen käytettävyysarviointi .....	56
	4.4 Esimerkki heuristisen käytettävyysohjeiston käytöstä.....	65
	4.5 Yhteenveto.....	70
5	ASiantuntija-Arviointin soveltaminen RIA- VERKKOSOVELLUKSIIN .....	74
	5.1 RIA-verkkosovellukset ja käyttäjäkokemus .....	75
	5.2 RIA-verkkosovellukset ja käytettävyys .....	79
	5.2.1 RIA-verkkosovellusten ominaispiirteet käytettävyysvaatimusten lähteenä.....	79
	5.2.2 Konventiot.....	82
	5.2.3 Sovelluksen sijainti ja löydettävyys .....	84
	5.3 Tarkastuslista RIA-verkkosovellusten käytettävyyden asiantuntija- arviointin tueksi.....	86
6	YHTEENVETO .....	91
	LÄHDELUETTELO.....	95

## KUVIOT

### KUVIO 1.

N-tasoinen verkkosovelluksen arkkitehtuurimalli Offuttia (2002) mukailleen... 18

### KUVIO 2.

Esimerkki HTML-sivunkuvauskielellä toteutetusta WWW-lomakkeesta McDonaldia (2006, 5) mukailleen..... 19

### KUVIO 3.

Verkkosovelluksen suorittaminen asiakas-palvelin -mallia noudattaen sekä verkkosovelluksen suorittaminen asiakaskoneella McDonaldia (2006, 8) mukailleen..... 21

### KUVIO 4.

AJAX-verkkosovelluksen toimintomalli Zakasia, McPeakia ja Fawcettia (2006) mukailleen..... 37

### KUVIO 5.

Käytettävyyden osatekijät Nielsenin mukaan (Nielsen 1993, 25)..... 43

### KUVIO 6.

Duunitalkoot-sivustoon upotettu Suunnitelmakorit-verkkosovellus (Työterveyslaitos 2007) ..... 66

### KUVIO 7.

Käyttöliittymän hauskuutta lisäävät elementit Shneidermania (2004) mukailleen..... 77

## TAULUKOT

### TAULUKKO 1.

Sovellustyyppien luokittelu Ginigen ja Murugesanin (2001) mukaisesti..... 12

### TAULUKKO 2.

RIA-verkkosovellusten toteutustekniikoiden vertailua mukailleen lähdettä Noda & Helwig (2005). ..... 33

### TAULUKKO 3.

Kognitiivisen läpikäynnin neljä arviointikysymystä (Wharton ym. 1994)..... 52

### TAULUKKO 4.

Keinosen (1998) yleistämät seitsemän keskeistä käytettävyyden osatekijää..... 58

### TAULUKKO 5.

Nielsenin Kymmenen heuristiikan lista (Nielsen 1994, 30) ..... 60

**TAULUKKO 6.**

**Shneidermanin Kahdeksan kultaista sääntöä dialogin suunnitteluun (Shneiderman 1992, 72-73) ..... 63**

**TAULUKKO 7.**

**Nielsenin viisiportainen vakavuusluokitus havaituille käytettävyysoongelmille (Nielsen 1994, 49) ..... 64**

**TAULUKKO 8.**

**Yhteenveto Suunnitelmakorit-verkkosovelluksen heuristisesta arviosta..... 68**

**TAULUKKO 9.**

**Tarkastuslista RIA-verkkosovellusten käytettävyyden asiantuntija-arvioinnin tueksi..... 87**

## 1 JOHDANTO

WWW-verkkosovellus on usein hajautettu asiakas-palvelinjärjestelmä, joka on toteutettu useilla eri ohjelmointikielillä. WWW-verkkosovelluksen ohjelmakoodia voidaan palvelimen lisäksi suorittaa myös asiakkaan koneella. WWW-verkkosovellusta käytetään lähinnä internet-selaimella. (Offutt 2002) WWW-verkkosovelluksissa ovat keskeisessä asemassa käyttäjän toimet ja käyttäjän antamien syötteiden vastaanottaminen ja käsittely, järjestelmän antama palaute sekä tilatieto (Hassan & Holt 2001). Tässä tutkielmassa WWW-verkkosovelluksesta käytetään jatkossa lyhyempää verkkosovellus-termiä.

Nielsenin (1993, 26) mukaan käytettävyys ei ole käyttöliittymästä havaittava yksiulotteinen ominaisuus, vaan se koostuu useista tekijöistä. Sinkkosen, Kuoppalan, Parkkisen ja Vastamäen (2002, 19) mukaan käytettävyys on kokonaisuudessaan menetelmä- ja teoriakenttä, jonka kautta käyttäjän ja laitteen yhteistoiminta pyritään saamaan sekä tehokkaaksi että miellyttäväksi. Toisinaan myös subjektiivinen käyttäjän kokemus on tapa määritellä käytettävyyttä (Ovaska, Aula & Majaranta 2005, 4). Vaikka useat tutkijat ovat määritelleet käytettävyyden käsitteen, sille ei ole pystytty tuottamaan yhtä yksikäsitteistä määritelmää (Ovaska, Aula & Majaranta 2005, 3). Käytettävyys on epämääräinen käsite, jota voidaan tarkastella suppeasti yhtenä laatuominaisuutena esimerkiksi tehokkuuden näkökulmasta, tai ajatella sitä laajemmin tuotteen käytöstä seuraavana kokonaisvaltaisena laatukokemuksena.

Perinteiset verkkosovellusten esitystavat ja -teknologiat ovat pakottaneet luomaan monimutkaisistakin sovelluksista toiminnallisuudeltaan lineaariseen sivupohjaiseen prosessiin perustuvia. Tästä aiheutuu esimerkiksi lukuisia sivujen uudelleenlatauksia, mikä on yksi perinteisten verkkosovellusten ongelma. Sujuvan käytön kannalta olisi kuitenkin tavoiteltavampaa, että käyttäjän ei tarvit-

sisi olla tietoinen asiakas-palvelin-arkkitehtuurista johtuvasta liikennöinnin mekaniikasta. (Perfetti & Spool 2002) Toinen ongelma perinteisiä verkkosovelluksia koskien on, etteivät ne myöskään tue ominaisuuksiltaan multimediaa tai informaation visuaalista esittämistä parhaalla mahdollisella tavalla, mikä on merkittävä asia sekä käytettävyyden että käyttäjäkokemuksen kannalta. (Saari-  
luoma 2004, 116) Näihin haasteisiin vastaavat vähitellen yleistyvät uusien rikkaiden verkkosovellusten, eli RIA-verkkosovellusten, toteutusmallit ja teknologiat.

Perinteiset käytettävyyden asiantuntija-arvioinnin menetelmät muodostavat käytettävyyttä arvioitaessa ja käytettävyydeltään korkeatasoisia tuotteita laadittaessa onnistuneen toteutuksen kivijalan. Koska perinteiset käytettävyyden asiantuntija-arvioinnin menetelmät jättävät kuitenkin joiltain osin RIA-verkkosovelluksissa ilmeneviä piirteitä käsittelemättä, on olemassa tarve erityisesti RIA-verkkosovelluksille kohdistetuille tarkastuslistoille, joilla voidaan tukea ja täydentää olemassa olevia asiantuntija-arvioinnin menetelmiä (Asleson & Schutta 2006, 16).

Käytettävyyden arviointimenetelmiksi on valittu sellaisia, joilla suunnittelija voi selvittää tuotteen käytettävyyttä ilman loppukäyttäjää tai testiryhmää ja jotka soveltuvat käytettäväksi jo kehitystyön varhaisessa vaiheessa. Koska verkkosovelluksia ei eri syistä johtuen useinkaan altisteta käytettävyydestä huolehtimiseen tällaisessa tilanteessa erityisen arvokas. Tutkielmassa esitetyt käytettävyyden asiantuntija-arvioinnin menetelmät soveltuvat joustavasti erilaisten verkkosovellusten arviointiin. Verkkosovellusten luokka ja laajuus voivat vaihdella esimerkiksi pienestä informaation visualisointiin tehdystä komponentista verkkokauppaan.



Tutkimusongelman muodostavat seuraavat kysymykset: Millä tavalla RIA-verkkosovellukset eroavat perinteisistä verkkosovelluksista? Miten näiden käytettävyyttä ja käyttäjäkokemusta voidaan tutkia ja arvioida asiantuntija-arvioinnin menetelmin?

Tutkimusongelmien alakysymyksiä ovat: Miten käytettävyys määritellään? Minkälaisia verkkosovelluksille sopivia käytettävyysohjeistoja on olemassa ja miten niitä voidaan soveltaa? Millaisia periaatteita RIA-verkkosovellusten käytettävyyden asiantuntija-arvioinnin tueksi kehitetty tarkistuslista sisältää?

Tässä tutkielmassa verkkosovellukset esitellään käytettävyyspohdintojen edellyttämällä tasolla. Verkkosovellusten ohjelmointia tai eri ohjelmointikielten ominaisuuksia ei esitellä lainkaan. Tutkielmassa ei oteta kantaa verkkosovelluksiin, joilla on tarkoitus korvata työpöytäsovelluksia. Tällaisia pois suljettavia tapauksia ovat siten esimerkiksi selaimella käytettävä sähköposti ja taulukkolaskenta. Mobiilisovellukset ja mobiilikäyttö rajataan myös pois, sillä ne muodostavat haasteineen aivan oman ongelmakenttänsä.

Tutkielman seuraavassa luvussa esitellään perinteisten verkkosovellusten ominaispiirteet sekä tarkastellaan niiden toimintaperiaatteita. Luku 3 kuvaa millaisia ovat puolestaan RIA-verkkosovellukset. Luvussa 4 perehdytään käytettävyyteen, käyttäjäkokemukseen sekä käytettävyyden asiantuntija-arvioinnin menetelmiin ja annetaan lopuksi käytännön esimerkki heuristisen käytettävyysohjeiston soveltamisesta. Luvussa 5 pohditaan käytettävyyttä ja käyttäjäkokemusta RIA-verkkosovellusten kontekstissa ja kuinka käytettävyyden asiantuntija-arvioinnin menetelmiä voidaan soveltaa RIA-verkkosovellusten arviointiin. Luvun 5 keskeinen tulos on tarkistuslista RIA-verkkosovellusten käytettävyyden asiantuntija-arvioinnin tueksi. Luku 6 on yhteenveto.

## 2 VERKKOSOVELLUKSET

Alkujaan WWW eli World Wide Web suunniteltiin informaation esitysalustaksi, jossa sisältö koostui hyperlinkitetyistä tekstidokumenteista. Sitten WWW-ympäristöstä on tullut alusta myös erilaisille verkkosovelluksille. (Offutt 2002) Offuttin (2002) ja Pressmanin (2000) mukaan verkkosovellusten määrä ja merkitys ovat kasvaneet valtavasti ja verkkosovellukset ovatkin olleet jo vuosia yksi suurimmista ja merkittävimmistä ohjelmistoteollisuuden osa-alueista.

Ensimmäisessä aluvussa kuvataan millaisia yleisiä ominaispiirteitä verkkosovellukset omaavat. Toisessa aluvussa valotetaan verkkosovellusten toiminnan mekaniikkaa. Kolmas alaluku on yhteenveto.

### 2.1 Verkkosovellusten ominaispiirteitä

*Verkkosovelluksissa* ovat keskeisessä asemassa käyttäjän toimet ja käyttäjän antamien syötteiden vastaanottaminen ja käsittely, järjestelmän antama palaute sekä tilatieto (Hassan & Holt 2001). Verkkosovellus voi olla yksinkertaisimmillaan tietoa jakava WWW-sivusto ja toisessa ääreläidassa WWW:n kautta käytettävä hajautettu tietojärjestelmä, jossa on useita eri koneissa suoritettavia ohjelmistokomponentteja (Hassan & Holt 2001; Offutt 2002). Verkkosovelluksen koodia voidaan palvelimen lisäksi suorittaa myös asiakkaan koneella (Hassan & Holt 2001). Oman haasteensa verkkosovelluksien toteuttamiselle asettaa myös loppukäyttäjien laitteistojen lukemattomat erilaiset variaatiot.

Verkkosovellus voi koostua esimerkiksi sekalaisesta joukosta ohjelmakomponentteja, tulkattavista skripteistä, HTML- (Hyper Text Markup Language) tiedostoista, tietokannoista, grafiikasta, monimutkaisista käyttöliittymistä sekä edellisten lukuisista kombinaatioista. Edelleen verkkosovelluksen suunnittelijalla on mahdollisuus valita sovellusta koostaessaan mittatilaustyönä toteutettu-

jen ohjelmistokomponenttien lisäksi erilaisia modifioituja valmisohjelmia ja muita kolmansien osapuolien tuottamia ratkaisuja. Näiden lukuisten eri osien ja niiden välillä suoritettujen monimutkaisten valinta- ja integrointiprosessien jälkeen haasteeksi nousee ohjelmistotuotteen laatu. (Offutt 2002) Myös ohjelmistotuotteen käytettävyys on yksi laadun osatekijä (ISO 9126, 1991) ja Offuttin (2002) mukaan myös merkittävä kilpailuetuun vaikuttava seikka WWW:n ollessa toimintaympäristönä.

Verkkosovelluksilla voidaan lisätä sivuston toiminnallisuutta, jolloin käyttäjät voivat olla pelkkien tekstien ja linkkien sijaan vuorovaikutuksessa liiketoimintatiedon kanssa. Nielsen (2000, 258-260) jaottelee verkkosovellukset kahteen pääluokkaan, toiminnallisiin sovelluksiin ja sisältösovelluksiin. Toiminnalliset sovellukset ovat itsenäisiä minisovelluksia, joissa on erilaisia tiloja ja näkymiä. Toiminnallisten sovelluksien avulla käsitellään usein ”todellista” dataa, joka on olemassa myös sivun ulkopuolella, esimerkiksi yhteiskäyttöisissä tietokannoissa. Tällainen sovellus voi olla esimerkiksi pankkisovellus, varastonhallintaso- vellus tai palvelimen konfigurointia varten tehty liittymä. Sisältösovellukset ovat kiinteästi kytköksissä verkkosivun sisältöön ja niiden ajaminen vaikuttaa tyypillisesti vain sivuston ulkoasuun ja kulloinkin esillä olevaan näkymään, eli mitään tietoa ei oteta pitempiaikaisesti talteen. Sisältöön vaikuttava sovellus voi toteuttaa esimerkiksi navigointiin tarvittavat toiminnot tai olla pelkästään havainnollistavaksi elementiksi laadittu interaktiivinen animaatio.

Ginige ja Murugesan (2001) ovat luoneet hienojakoisemman ja toimintopainotteisemman luokittelun, jossa verkkosovelluksille on seitsemän erilaista kategoriaa (TAULUKKO 1). WWW-portaalin nostaminen omaksi sovellustyyppikseen on mielestäni kuitenkin turhaa, sillä se voitaisiin hyvin sisällyttää esimerkiksi luokkaan informaatiota jakavat sovellukset. Mitä useampaan osaan jako teh-

dään, sitä useammin verkkosovelluksessa on ominaisuuksia useasta eri kategoriasta.

**TAULUKKO 1. Sovellustyyppien luokittelu Ginigen ja Murugesanin (2001) mukaisesti.**

Kategoria	Esimerkkisovelluksia
Informaatiota jakavat (informational)	Verkkolehdet, tuotekatalogit, sähköiset ohjekirjat, sähköiset kirjat
Interaktiiviset, joissa informaatio käyttäjältä (interactive, user provided information)	Rekisteröintilomakkeet, verkkopelit
Tapahtumaperustaiset (transactional)	Sähköiset kauppapaikat, verkkopankki
Työnkulkua edistävät (workflow)	Verkkopohjaiset suunnittelu- ja projektihallintatyökalut
Ryhmätöohjelmistot (collaborative work environments)	Ryhmätöitä edistävät verkkosovellukset
Verkkoyhteisöt (online communities, marketplaces)	Keskusteluryhmät, suosittelupalvelut, verkkohuutokaupat
WWW-portaalit (web portals)	Portaalit

Verkkosovellusten kehitystyön ominaispiirteiksi on usein mainittu amatöörimäisyys ja menetelmällinen paluu kymmenien vuosien takaisin ohjelmistotuotannon alkuvuosiin, jolloin ad-hoc -tyyppiset kehitysmenetelmät olivat käytetyimpiä (Lang & Fitzgerald 2005). Toisaalta Constantine ja Lockwood (2002) ovat sitä mieltä, että raskaat analyysi- ja suunnittelumallit eivät sovi verkkosovellusten kehityksen nopeaan luonteeseen ja tyypillisesti vaadittuihin nopeisiin toimitusaikoihin. Verkkosovellusten tuotannon ongelmallisuus liittyy

Pressmanin (2000) mukaan ensisijaisesti siihen, että kyseessä on yhä suhteellisen uusi sovellusalue. Tällöin käytetään usein vakiintumattomia ratkaisuja ja lukuisia ohjelmointikieliä sekaisin kunkin kehittäjän mieltymysten mukaisesti (Ginige & Murugesan 2001; Offutt 2002).

Verkkosovellusten tuotannosta tekee ongelmallisen myös se, että usein niiden tuotanto on ulkoistettua ja toimittajan laatu voi olla vaikeaa selvittää tilaajalle. Sekä tilaajan että toimittajan on lisäksi hyvin hankalaa arvioida etukäteen kuluja ja aikataulukysymyksiä, koska tyypillisesti tuote elää ja muuttuu dramaattisesti kehitysprosessin aikana. Verkkosovelluksia on mahdollista luoda myös pienemmin resurssein kuin varhaisempia tietojärjestelmiä ja ne voivat perustua useiden pienten sovellusten verkostoon. (Pressman 2000) Riippuvuus yksittäisistä henkilöistä on korostunut sekä verkkosovellusten tuotannon että dokumentaation osalta, ja henkilövaihdokset voivat rampauttaa järjestelmän – erityisesti ylläpidon ja jatkokehityksen osalta (Hassan & Holt 2001).

Viime aikoina puhe verkkosovellusten ohjelmistokriisistä on kuitenkin vähentynyt ja aikaisempi diskurssi tästä nähdään liioiteltuna. Yhtenä syynä tähän ehdotetaan alan kypsymistä ja tervehtymistä, johon sisältyy tärkeänä tekijänä vanhojen ja hyväksi koettujen järjestelmällisten tuotantomallien sovittaminen uuteen ympäristöön. (Lang & Fitzgerald 2005) Pressmanin (2000) mielestä yleiset ohjelmistotuotannon toimintamallit ovatkin hyödynnettävissä myös verkkosovelluksia tehtäessä. Vanhoja toimivia malleja ei tulisi hylätä kokonaan uuden ohjelmistoparadigman myötä, mikä koskee pitkälti myös käytettävyyssajattelua ja käytettävyyssuunnittelun soveltamista. Lisäksi uuden sovellustyyppin ja yleisen internet-huuman mukanaan tuoma alkuryntäys on hiipunut ja epäpätevimmat toimijat ovat mahdollisesti karsiutuneet markkinoilta (Lang & Fitzgerald 2005).

## 2.2 Verkkosovellusten toimintaperiaate

Verkkosovellusten merkittävänä etuna on pidetty asiakaskoneista johtuvien ylläpitokustannusten poistumista, koska yksinkertaisimmillaan palvelimelta jaettava sovellusta varten ei tarvitse huolehtia erillisistä asennuksista asiakaskoneelle – riittää kun asiakkaalla on toimiva WWW-selain, joka jo itsessään on käyttöliittymä verkkosovellusta varten (Polvinen 1999, 176). Verkkosovellusten käyttöliittymä on perinteisesti ja standardinmukaisesti toteutettu HTML-sivunkuvauskielellä, jossa tarvittavat elementit ovat olleet jo yli kymmenen vuotta olemassa (McDonald 2006, 6). Verkkosovellusten ohjelmakoodi voi sijaita erillisissä tiedostoissa tai upotettuna HTML-koodin sekaan, joskin asiakkaan koneelle tulostetaan yleensä vain tuloksena syntyvä päällisin puolin staattiselta HTML-sivulta vaikuttava tiedosto. Seuraavaksi esitellään perinteisten WWW-verkkosovellusten toiminnan ja tietoliikenteen mekaniikkaa sekä asiakaspalvelin -arkkitehtuuri.

*HTTP (Hyper Text Transfer Protocol)* on asiakaskoneen ja verkkopalvelimen keskinäiseen tiedonsiirtoon käyttämä yhteyskäytäntö. Nimestään huolimatta protokolla soveltuu käytännössä monenlaisen datan siirtoon. Tämä standardoitu käytäntö on omaksuttu yleisesti yhteensopivuusongelmien välttämiseksi. HTTP-siirtotapahtuma koostuu kahdesta HTTP-viestistä (*engl. HTTP message*), jotka ovat asiakaspyyntö (*engl. HTTP request*) ja pyyntöjä kuuntelevan palvelimen vastaus (*engl. HTTP response*). Verkkopalvelin käsittelee pyynnöt ja normaalisti vastaa lähettämällä selaimelle pyydetyn datan, esimerkiksi HTML-sivun kuvineen, jonka selain esittää käyttäjälle. (Rantala 2005, 99-100)

Kun palvelin HTTP-siirtotapahtumassa lähettää vastauksena asiakkaan pyytämän datan, palvelin ei pidä tämän toimenpiteen jälkeen yllä mitään tähän transaktioon liittyvää tietoa. Jos asiakas tekee hetken kuluttua uuden pyynnön

palvelimelle, palvelin ei tiedä näin ollen mistään, että kyseessä on sama asiakas kuin juuri hetki sitten. Tästä johtuen HTTP:tä kutsutaankin tilattomaksi protokollaksi. (Kurose & Ross 2005, 90) Verkkosovellusten, kuten virtuaalisen kaupapaikan, pitäisi kuitenkin pystyä pitämään yllä tilatietoa, asiakkaan tietoja sekä tietoja asiakkaan aiemmista toimenpiteistä palvelussa eri näkymien tai sivulatausten välillä, jotta monivaiheinen prosessi voitaisiin suorittaa onnistuneesti.

Verkkosovelluksen käytön aikana kerättävää tila- ja prosessitietoa voidaan kuljettaa sovelluksen näkymästä toiseen, eli useimmiten verkkosivulta toiselle, verkko-osoitteeseen liitettynä. Tällöin muuttujatieto URL-koodataan selaimen osoitekentässä sijaitsevan osoitetekstijonon perään, jolloin seuraava sovelluksen vaihe voi käyttää osoitekentässä sijaitsevaa tietoa. Muuttujatietoa voidaan kuljettaa läpi verkkosovelluksen prosessin myös tietoturvaltaan paremmalla tavalla HTTP-viestien data-kentässä, jolloin muuttujatieto näkyy vain tarvittaessa sovelluksen kautta. (Rantala 2005, 80; 107-108) Kummassakin tapauksessa selaimen sulkeminen hävittää tiedot, joten tiedon pidempiaikaiseen tallentamiseen on käytettävä muita keinoja.

**Keksi** (*engl. cookie*) on pieni käyttäjän koneella säilytettävä tiedosto, johon tallennetaan dataa myöhemmin hyödynnettäväksi. Keksiin voidaan tallentaa esimerkiksi käyttäjätunnus ja salasana tai verkkosivun käyttäjäkohtaisia asetuksia, jolloin niitä ei tarvitse syöttää joka vierailukerralla uudelleen. Keksit ovat myös keskeisiä tilatiedon hallinnassa useampien käyttökertojen välillä. (Powell 2002, 686) Kun käyttäjä avaa palvelimelta keksejä hyödyntävän verkkosivun, käyttäjälle luodaan ainutkertainen tunniste ja käyttäjän tietokoneelle sitä vastaava keksitiedosto, jonka hallinnoimisesta selain huolehtii. Vastaavasti palvelimen yhteydessä toimivaan tietokantaan tehdään merkintä, jolloin voidaan seurata tietyn käyttäjän toimia tunnistenumerotasolla ja ylläpitää siihen liittyviä tietoja. Keksitiedot välittyvät HTTP-viesteissä selaimen ja palvelimen välillä. Käyttäjän

koneella oleva aktiivinen keksi antaa eri käyttökertojen välillä tietonsa palvelimelle, joka tunnusteen perusteella tietää käyttäjän olevan vanha vierailija ja pysyy hyödyntämään tunnistesidonnaisia tallennettuja tietoja. (Kurose & Ross 2005, 98-99)

*Sessio* eli *istunto* on vielä yksi tapa pitää yllä verkkosovelluksen tilatietoa sovelluksen käytön aikana. Sovelluksen käynnistyessä käynnistetään istunto, jolloin käyttäjälle luodaan ainutkertainen tunniste, istuntoavain. Verkkosovelluksen palvelimelle luodaan tätä tunnistetta nimeltään vastaava tiedosto, johon voidaan tallettaa tilatiedot säilyttävät muuttujatiedot kuten myös muita muuttujia. Niin kauan kuin istunto on voimassa, palvelimen istuntotiedostoon tallennettu muuttujatieto on käytettävissä globaalisti kaikilla samaisella palvelimella sijaitsevan sovelluksen sivuilla. Istunto päättyy määritellyn ajan kuluttua, kun palvelin, asiakas tai molemmat päätteet katkaisevat yhteyden. Tavallisesti istunnon tiedot häviävät sen päättyessä, mutta istunto on myös mahdollista tallentaa, jolloin vanha istunto uusitaan seuraavalla käyttökerralla. (Greenspan & Bulger 2001, 83-84)

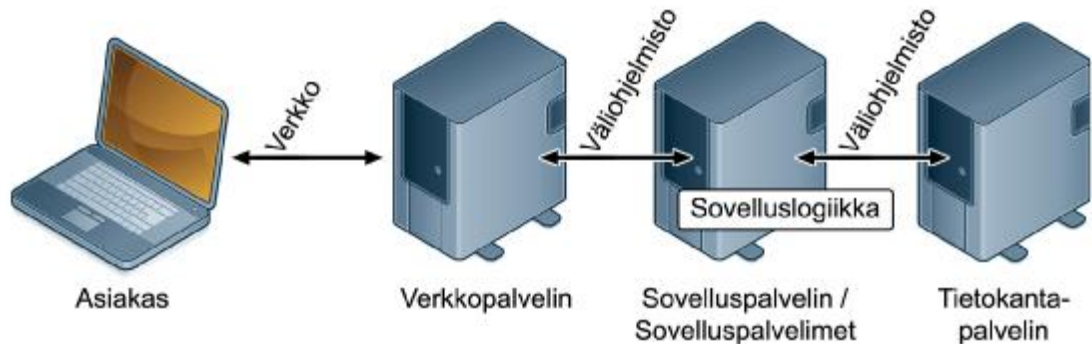
Tietokantayhteys verkkosovellukseen voidaan luoda lukuisilla vaihtoehtoisilla tavoilla, riippuen myös käytetystä arkkitehtuurimallista. Tietokannan sisältö saadaan verkkosovelluksen käyttöön esimerkiksi käyttöoikeuksien tarkistamisessa hyödynnetyn LDAP (Lightweight Directory Access Protocol) -protokollan sekä tietokannan ja verkkosovelluksen yhdistävän ODBC (Open Database Connectivity) -tietokantarajapinnan ja SQL-kyselyiden avulla. (Garfinkel & Spafford 1997, 194)

Fyysisesti toteutetussa *asiakas-palvelin -arkkitehtuurissa* (engl. *client-server architecture*) on käyttöliittymä ja sovelluslogiikka sekä usein tietokanta, jotka on yhdistetty toisiinsa verkolla (Polvinen 1999, 21). Asiakas on niin sanottu *laiha asiakas*



(*engl. thin client*), joka vastaa vain palvelimen tuottamien dokumenttien esittämisestä ja uusista dokumenttipyyntöistä. Kaksitasoisessa (*engl. two-tier*) mallissa palvelin sisältää sekä sovelluslogiikan että tietokantapalvelut yhdessä paikassa. (Fraternali 1999) Palvelimella sijaitsevat myös staattiset verkkosivutiedot sekä mahdolliset erilaiset multimediaobjektit (Hassan & Holt 2001). Kaksitasoisen mallin suurin ongelma on sen skaalautuvuudessa ja kestäkyvyssä tietoliikennemäärien kasvaessa, eikä sitä juurikaan enää käytetä (Offutt 2002).

Kolmitasoisen (*engl. three-tier*) arkkitehtuurimallin fyysinen toteutus poikkeaa kaksitasoisesta mallista siten, että asiakkaan ja sovelluspalvelimen lisäksi on erillinen tietokantapalvelin. Ohjelmien suorittamisesta huolehtiva sovelluspalvelintaso pitää sisällään tällöin ainoastaan sovelluslogiikan varsinaisen dataaineiston ollessa tallessa tietokantapalvelimella. (Polvinen 1999, 176) Kolmitasoisen arkkitehtuurin lisäksi käytetään usein tätä laajempaa N-tasoista arkkitehtuuria (KUVIO 1). Verkkosovelluksen eri ohjelmistokomponentit voivat sijaita myös erillisillä sovelluspalvelimilla hajautetusti. Järjestelmän komponenttien välillä voi toimia edelleen väliohjelmistoja, jotka hoitavat liikennöintiä eri osien välillä yhteensopivuudesta huolehtien. N-tasoinen arkkitehtuurimalli lisää palvelun luotettavuutta, jolloin esimerkiksi laiterikko ei välttämättä lamauta koko systeemiä. Tarpeiden kasvaessa on myös helpompaa lisätä laitteistoarseenaa joustavasti tai vaihtaa eri tasoilla toimivia ratkaisuja niiden ollessa keskitettyihin arkkitehtuurimalleihin verrattuna toisistaan vähemmän riippuvaisia. (Offutt 2002; Roe & Gonik 2002)



KUVIO 1. N-tasoinen verkkosovelluksen arkkitehtuurimalli Offuttia (2002) mukailten.

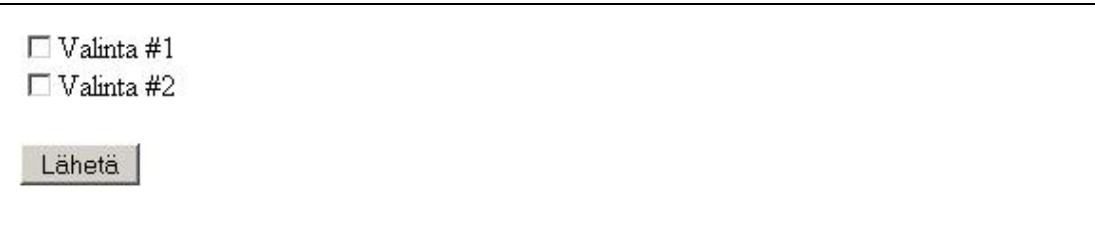
Loogisesti kaikissa arkkitehtuurimalleissa on esitystaso, sovellustaso ja datataso. Datataso tallentaa dataa ja antaa dataa sovellustasolle, sekä huolehtii datan eheydestä. Sovellustaso ottaa esitystasolta, eli käytännössä loppukäyttäjältä, tulevan syötteen, käsittelee sen ja lähettää tulokset takaisin. Esimerkiksi kyseessä voisi olla tilauksen vastaanotto, sen tallentaminen tietokantaan ja ilmoittaminen käyttäjälle tilauksen onnistumisesta. Esitystaso tarjoaa käyttäjälle näkyvän käyttöliittymän. (Polvinen 1999, 22) Verkkosovelluksessa esityskerros on osittain myös palvelimella selaimen lisäksi, koska palvelin lähettää selaimelle tyypillisesti HTML-koodia osallistuen näin esityspalveluiden muodostukseen. Selain näyttää loppukäyttäjälle sovelluksen käyttöliittymän ja datan ymmärrettävässä muodossa. (Polvinen 1999, 178)

Kuvion 2 esimerkissä käyttäjän painaessa LÄHETÄ-painiketta (KUVIO 2), kaikki selaimessa näkyvälle lomakkeelle annettu data lähetetään palvelimelle käsiteltäväksi yhtenä pitkänä merkkijonona. Palvelinpäässä sovelluslogiikka

ottaa datan vastaan ja tekee sille tarvittavat toimenpiteet sekä mahdollisesti palauttaa uuden käyttöliittymänäkymän käyttäjälle.

Perinteisiä HTML-lomakkeita käyttävät verkkosovellukset eroavat toisistaan viime kädessä eniten juuri palvelinohjelmistojen eli sovellustason osalta. (McDonald 2006, 6) Sovellustaso on voitu toteuttaa esimerkiksi PHP:llä, C#:lla tai monella muulla vaihtoehtoisella ohjelmointikielellä (McDonald 2006; Rantala 2005).

```
...  
<form method="get" action="datan_kasitteleva_skripti">  
  <input type="checkbox">Valinta #1<br>  
  <input type="checkbox">Valinta #2<br><br>  
  <input type="submit" value="Lähetä">  
</form>  
...
```



The image shows a rectangular box divided into two horizontal sections. The top section contains HTML code for a form with two checkboxes and a submit button. The bottom section shows the rendered output of that code: two unchecked checkboxes and a button labeled 'Lähetä'.

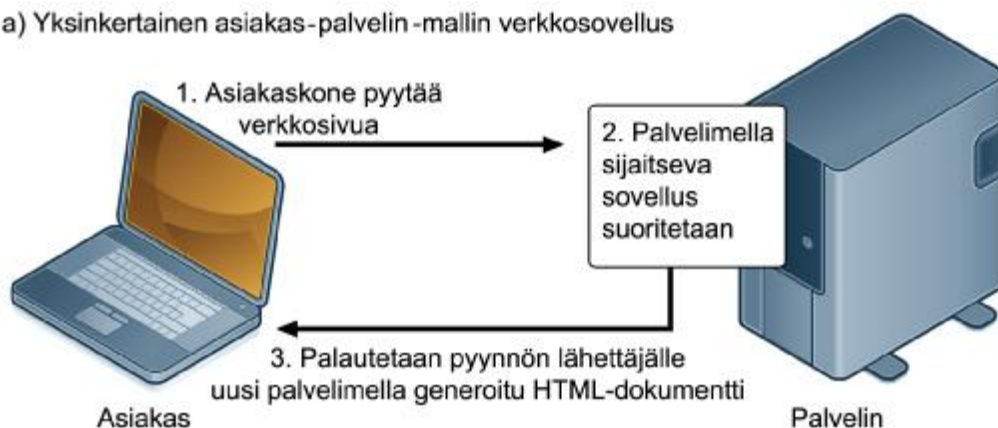
**KUVIO 2.** Esimerkki HTML-sivunkuvauskielellä toteutetusta WWW-lomakkeesta McDonaldia (2006, 5) mukaillen.

Verkkosovellukset voivat toimia myös asiakaskoneen varassa, jolloin puhutaan *lihavasta asiakkaasta* (*engl. fat client*) (Polvinen 1999, 22). Verkkosivuihin voidaan upottaa erilaisia multimediaelementtejä sekä esimerkiksi Java-sovelmia tai JavaScript-sovelluksia. Asiakaspuolen sovellukset eivät hyödynnä palvelinta las-

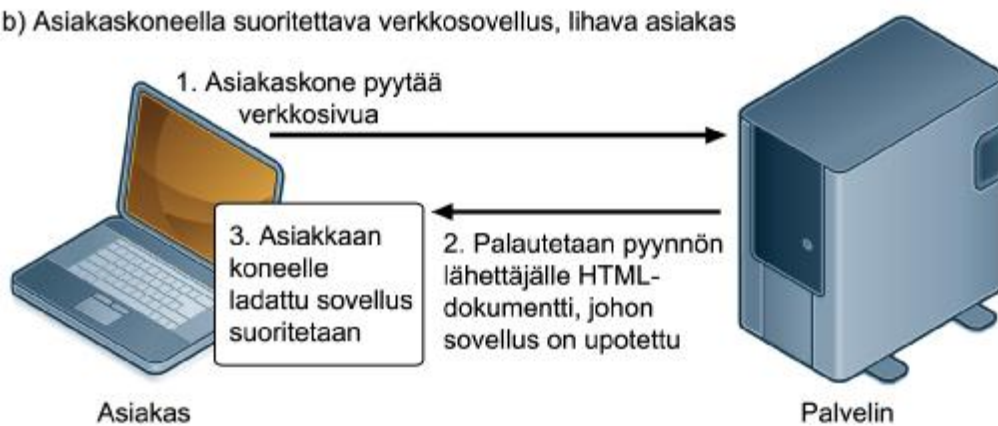
kennassa, vaan sovellukset ladataan käyttäjän koneelle, joka suorittaa ne paikallisesti. Asiakaskoneella hoidetaan datan käsittelyn lisäksi myös käyttöliittymään kytkeytyvät prosessit ja palvelinta tarvitaan lähinnä datan säilytykseen. (McDonald 2006, 7)

Verkkosovelluksen suoritukseen liittyvä etenemisjärjestys ja suoritettavan sovelluksen sijainti on havainnollistettu seuraavassa kuviossa (KUVIO 3). Sekä asiakas-palvelin -mallin verkkosovelluksessa että asiakaskoneella suoritettavan verkkosovelluksen suorittaminen lähtevät liikkeelle asiakaskoneen lähettämältä pyynnöstä. Eroavaisuudet syntyvät siitä, suoritetaanko sovellus tämän jälkeen palvelimella vai asiakkaan koneella.

## a) Yksinkertainen asiakas-palvelin-mallin verkkosovellus



## b) Asiakaskoneella suoritettava verkkosovellus, lihava asiakas



KUVIO 3. Verkkosovelluksen suorittaminen asiakas-palvelin -mallia noudattaen sekä verkkosovelluksen suorittaminen asiakaskoneella McDonaldia (2006, 8) mukailleen.

Verkkosovellukset voivat kommunikoida myös keskenään toistensa resursseja hyödyntäen, joko käyttäjän nimenomaisen pyynnön seurauksena tai itsenäisesti. Verkkosovellusten keskinäistä yhdistämistä varten ovat syntyneet XML-pohjaiseen viestintään perustuvat *verkkopalvelut* (*engl. web services*). Tätä tekniikkaa käyttäviin verkkosovelluksiin on määritelty, millaisia pyyntöjä kukin sovellus ottaa vastaan ja mitä tietoa lähetetään vastauksena. Internetissä hajallaan sijaitsevat sovellukset tai niiden osat voidaan integroida verkkopalveluiden avulla siten, että asiakas voi käyttää muodostuvaa kokonaisuutta yhden

liittymän kautta ikään kuin yksittäistä sovellusta. Verkkopalveluiden ja väliohjelmistojen avulla voidaan yhdistää myös yritysten erilaisia taustajärjestelmiä verkkosovelluksen osaksi. (Newcomer 2002, 6-7)

### 2.3 Yhteenveto

Verkkosovelluksissa ovat keskeisessä asemassa käyttäjän toimet ja käyttäjän antamien syötteiden vastaanottaminen ja käsittely, järjestelmän antama palaute sekä tilatieto. Verkkosovelluksilla voidaan lisätä sivuston toiminnallisuutta, jolloin käyttäjät ovat pelkkien tekstien ja linkkien sijaan vuorovaikutuksessa oikean ohjelman kanssa. Verkkosovellusten merkittävänä etuna on pidetty asiakaskoneista johtuvien ylläpitokustannusten poistumista, koska yksinkertaisimmillaan palvelimelta jaettava sovellusta varten ei tarvitse huolehtia erillisistä ohjelmistoasennuksista asiakaskoneelle. Verkkosovelluksen esitystasosta huolehtii internetselain, jolloin verkkosovellukset eroavat toisistaan viime kädessä eniten palvelinohjelmistojen osalta.

Verkkosovellusten kehitystyöhön sisältyy erilaisten ohjelmistokomponenttien toteutus-, valinta- ja integrointityötä, jotka vaikuttavat ohjelmistotuotteen laatuun sekä yksittäisinä komponentteina että yhteisvaikutuksen kautta. Verkkosovellusten kehitystyötä on vielä muutamia vuosia sitten moitittu usein amatöörimäiseksi perinteisiin ohjelmistotuotantoihin verrattuna. Nykyisin tällaiset väitteet ovat menettäneet merkitystään, sillä laadukkaisiin lopputuloksiin johtavat yleiset ohjelmistotuotannon toimintamallit ovat hyödynnettävissä myös verkkosovelluksia tehtäessä. Ohjelmistotuotteen käytettävyys on yksi laadun osatekijä.

Verkkosovelluksen fyysinen arkkitehtuuri on yleensä vähintään kolmitasoinen, jolloin toiminnasta vastaavat asiakaskoneen lisäksi sovelluspalvelin ja erillinen

tietokantapalvelin. Verkkosovellukset voivat toimia vaihtoehtoisesti myös asiakoneen varassa, jolloin ohjelmakoodi suoritetaan palvelimen sijaan asiakkaan koneella. Tällöin palvelinta tarvitaan lähinnä datan säilytykseen. Tiedon siirtoon käytetään yleisesti HTTP-protokollaa ja tilatiedon säilyttämisessä voidaan hyödyntää esimerkiksi keksejä tai istuntoja. Muuttujatietoja voidaan kuljettaa myös verkko-osoitteeseen koodattuna sovelluksen osasta toiseen siirryttäessä.

### 3 RIA-VERKKOSOVELLUKSET

RIA-termin juuret yltävät vuoteen 2002, jolloin yritys nimeltä Macromedia lanseerasi sen. Nimitys RIA tulee englanninkielisistä sanoista Rich Internet Application, eikä sille ole olemassa suomenkielistä vastinetta. (Noda & Helwig 2005) Jatkossa näihin niin sanottuihin rikkaisiin verkkosovelluksiin viitataan termillä RIA-verkkosovellukset. Normanin (2006) mukaan keskeisiä RIA-toteutustekniikoiden mahdollistamia etuja ovat sisällön lataaminen asynkronisesti selaimen välimuistiin ja asiakaspään laskentatehon hyödyntäminen. Viesintä palvelimen kanssa ei pysäytä muun koodin suoritusta, ja palvelimen lähettämä data ja asiakkaan pyytämä data eivät välttämättä täsmää keskenään, vaan käyttäjän koneelle voidaan esimerkiksi ladata etukäteen RIA-verkkosovelluksen tarvitsemaa dataa ilman erillistä pyyntöä sovelluksen toimintaa ennakoiden. Tuloksena näistä tekijöistä syntyvät esimerkiksi RIA-verkkosovellusten nopeat vasteajat ja sulavasti päivittyvät dynaamiset grafiikat.

Seuraavassa alaluvussa kuvataan WWW-ympäristön kehittymistä, joka asettaa myös verkkosovelluksille uusia vaatimuksia. Toisessa alaluvussa esitellään RIA-verkkosovellusten ominaispiirteitä ja kolmannessa alaluvussa tehdään katsaus RIA-verkkosovellusten toimintaperiaatteisiin yleisellä tasolla. Neljännessä alaluvussa perehdytään tarkemmin RIA-verkkosovellusten toteutusteknologia AJAX:iin. Viides alaluku on yhteenveto.

#### 3.1 Web 2.0 – muuttunut toimintaympäristö

Internet ei ole yhtenäinen entiteetti, josta voitaisiin selkeästi erottaa sen kehitysvaiheet ja versionumeroida ne. Tästä huolimatta on syntynyt käsite Web 2.0, jolla pyritään erottamaan aikaisemmin WWW-ympäristössä käytetyt toimintamallit ja tekniikat uudemmissa toteutustekniikoista ja toimintatavoista. Vaikka



internetin versionumeroinnista ollaankin montaa mieltä, ovat innokkaimmat alkaneet puhumaan jo Web 3.0:sta. Versioevoluution voidaan katsoa koskevan myös tätä ilmiötä siinä missä vaikkapa käyttöjärjestelmiä. Wainewrightin (2007) mukaan Web 2.0:ssa ja sen seuraajissa ei ole kyse vain viihteestä tai erilaisista hakupalveluista. Toimintaympäristön kehittyminen tuo tullessaan myös uuden sukupolven liiketoimintasovelluksia verkkoon.

Web 2.0:n sovelluksille tyypillisiä piirteitä ovat verkko sovellusalustana, rikkaat käyttäjäkokemukset, sosiaalinen sisällöntuotanto, ohjelmistojen laiteriippumattomuus, kevyet ja helpot ohjelmointiparadigmat sekä puuttuva ohjelmistojen julkaisusykli, jonka tilalla on sovellusten elinkaaren ajan jatkuva kehittäminen (Kangas, Toivonen & Bäck 2007, 12). Web 2.0:ssa ei ole kyse pelkästään mediasista ja tiedon jakamisesta vaan kommunikaatiosta, jossa laajat käyttäjäjoukot osallistuvat sovellusten tarjoamiin palveluprosesseihin ja tuottavat itse sisältöä. Helpoksi tehdyn sosiaalisen sisällön tuottamisen ja julkaisun haittapuolena on verkkosisältöjen signaali-kohina -tason heikkeneminen entisestään. (Boll 2007)

Web 2.0:n vaatima teknologia on sekä käyttöä helpottavaa käytettävyyteen ja käyttöliittymään liittyvää että enemmän taustalla pyöriviä palvelin- ja tietokantasovelluksia (Kangas ym. 2007, 52). Yksi Web 2.0:n merkittävimmistä painopisteistä on hyvä käytettävyys. Uuteen Web 2.0 -kategoriaan luokiteltavat sovellukset muistuttavatkin jo enemmän työpöytäsovelluksia kuin verkkosivuja paitsi ulkoasultaan myös toiminnoiltaan, minkä pitäisi olla tuntuva etu käytettävyyttä tavoiteltaessa. (Lewis 2006) Useita tällaisia toisen aallon verkkosovelluksia kutsutaan nimellä RIA-verkkosovellukset, joskin tietyin reunaehdoin (Noda & Helwig 2005).

Nyttemmin seurannut RIA-verkkosovellusten yleistymisen ja nousu laajempaan tietoisuuteen ovat useamman tekijän aikaansaannosta. Yksi tärkeimmistä

tekijöistä on suurten yritysten myönteinen suhtautuminen RIA-verkkosovelluksiin ja niiden toteuttamat RIA-verkkosovellukset. Esimerkiksi Google on tehnyt RIA-sovellusparadigmaa noudattaen Google Maps -karttasovelluksen sekä Gmail-selainsähköpostipalvelun ja tutustuttanut suuren määrän netin käyttäjistä rikassisältöiseen ja sujuvaan toiminnallisuuteen selainpohjaisissa sovelluksissa. (Noda & Helwig 2005).

Web 2.0 sisältää RIA-verkkosovellusten välityksellä myös kaikki perinteisen multimedian ainekset. Multimedian määrittelyssä ovat mukana jatkuvaluonteiset mediat, kuten video ja ääni, sekä diskreetit mediatyypit teksti ja kuva yhdessä ilmeten. Multi- eli monimedia vaatii ainakin yhden jatkuvaluonteisen ja yhden diskreetin mediatyypin ilmenemistä ja esittämistä yhdessä siten, että niihin liittyy myös interaktiivinen kontrollointimahdollisuus. (Boll 2007)

Verkossa esitettävien interaktiivisten prosessien monimutkaistumisen myötä ja verkon vakiinnutettua asemansa toimintaympäristönä verkkosovellukset voidaan nähdä myös laajemmin palveluina. Onhan käyttäjän ja sovelluksien välisessä kommunikaatiossa kyse varsin usein esimerkiksi ostotapahtumasta tai muusta tosielämänkin palvelutilanteeseen rinnastettavasta tapahtumasta. Grönroosin (2001, 79) mukaan palvelu on ainakin jossain määrin aineettomien toimintojen sarjasta koostuva prosessi, jossa toiminnot tarjotaan ratkaisuna asiakkaan ongelmiin ja jossa toimitaan yleensä keskinäisessä vuorovaikutuksessa asiakkaan ja palvelun tarjoajan välillä.

Palvelut koetaan subjektiivisesti ja kokemukseen vaikuttaa olennaisesti se, mitä osapuolten välisessä vuorovaikutuksessa, eli palvelutapaamisessa, tapahtuu. Koetulla laadulla on pohjimmiltaan kaksi ulottuvuutta: tekninen eli lopputulosulottuvuus ja toiminnallinen eli prosessiulottuvuus. Laatukokemukseen vaikuttaaakin se, millä tavalla tekninen laatu eli prosessin lopputulos toimitetaan.

(Grönroos 2001, 100-101) Vaikka käyttäjä saisi kahdelta kilpailevalta verkkosovellukselta saman lopputuloksen, koetaan niiden kokonaislaatu toisistaan poikkeavalla tavalla sen mukaan, kuinka hyvin ennen lopputulosta ilmenevä toiminnallinen ulottuvuus on osattu hoitaa. Tyypillisesti helppokäyttöisin ja mukavin palvelu nousee suosituimmaksi monesta samanlaisesta (Kangas ym. 2007, 52).

Web 2.0:aan ja sen käyttöliittymäilmentymiin liittyy myös verkkopalveluita toiminta-ajatukseltaan sivuava ilmiö *yhdistelmäpalvelut* (engl. *mash-up*). Tällaisissa palveluissa on yhdistetty sisältöä kahdesta tai useammasta erillisestä palvelusta, siis verkkosovelluksesta, yhden käyttöliittymän kautta käytettäväksi. Edellytyksenä tälle ilmiölle ovat uudet Web 2.0 -teknologiat, kuten avoimet rajapinnat sekä sisällön ja ulkoasun tehokas erottaminen toisistaan. (Kangas ym. 2007, 16) Yhdistelmäpalvelut hyödyntävät muiden palveluiden kapasiteettia, mikä mahdollistaa houkuttelevien palveluiden luomisen pienin resurssein. (Kangas ym. 2007, 53)

Useimmat RIA-toteutusteknologiat jakavat yhteisen ja ehkä hieman ristiriitaisenkin tavoitteenasettelun, jossa sovellusten kehitysaikojen pitäisi olla pienemmät ja tuotantohinnan halvempi kuin perinteisiä verkkosovelluksia tehtäessä, mutta samalla käyttäjäkokemuksesta olisi kuitenkin määrä tehdä monipuolisempi ja laadukkaampi (Noda & Helwig 2005). Seuraavassa alaluvussa perehdytään tarkemmin RIA-sovellusten ominaispiirteisiin.

### 3.2 RIA-verkkosovellusten ominaispiirteitä

Perinteiset verkkosovellusten esitystavat ja -teknologiat ovat pakottaneet luomaan monimutkaisistakin sovelluksista toiminnallisuudeltaan lineaariseen sivupohjaiseen prosessiin perustuvia. Tästä aiheutuu esimerkiksi lukuisia sivujen

uudelleenlatauksia, joka on yksi perinteisten verkkosovellusten ongelma. Sujuvan käyttäjäkokemuksen kannalta olisi kuitenkin tavoiteltavampaa, että käyttäjän ei tarvitsisi olla tietoinen asiakas-palvelin -arkkitehtuurista johtuvasta liikennöinnin mekaniikasta. (Perfetti & Spool 2002) Preciado, Linaje, Sanchez & Comai (2005) kuvaavat tätä käyttöliittymään liittyvänä prosessiongelmana sekä palauteongelmana, jotka liittyvät läheisesti toisiinsa. Molemmissa on avaintekijänä vaade edetä usean sivun kautta yhden tehtävän suorittamiseksi ja palautteen toimivan vasteen saamiseksi. Esimerkiksi autotehtaan autontilaussivuilla oleva räätälöintityökalu ei perinteistä verkkosovellusparadigmaa käyttäen näytä auton uutta erikoisväriä yhdellä käyttäjän suorittamalla klikkauksella vaan vaatii vähintäänkin lomaketietojen lähettämistä palvelimelle ja sivun uudelleen lataamista.

RIA-verkkosovelluksissa edes useampivaiheista kokonaisuutta ei ole välttämätöntä jakaa useille erillisille sivuille. Tästä on huomattavaa etua tietoa käsittelevälle käyttäjälle, jonka ei tarvitse pitää mielessään useiden sivujen avaintietoja. Näin myös varmistetaan paremmin, että ajatuksen punainen lanka ei katoa navigoitaessa yksittäisten sivujen välillä. Työmuistin ja tarkkaavaisuuden tarpeeton kuormittaminen vie tiedonkäsittelykapasiteettia pois informaation merkityksen arvioinnista, informaation jäsenykseltä ja asiakokonaisuuksien hahmottamiselta. (Müller 2004, 71) RIA-verkkosovelluksiakin tehtäessä on kuitenkin vältettävä kiusausta esittää samalla näytöllä liikaa tietoa kerrallaan. RIA-verkkosovellusten etuna on sivun osien dynaaminen päivitysmahdollisuus, jolloin vain tarvittavat osat päivittyvät tarpeita vastaavaksi. Lisäksi verkkoliikennöinnin mekaniikka voidaan eräissä tapauksissa häivyttää taustalle ja tällöin sovelluksen toiminnallisuus alkaa lähentyä jo huomattavasti työpöytäsovellusta (Perfetti & Spool, 2002).

Toinen ongelmakokonaisuus perinteisissä verkkosovelluksissa on, etteivät ne tue ominaisuuksiltaan multimediaa tai informaation visuaalista esittämistä parhaalla mahdollisella tavalla. Visuaalisemman esitysmuodon avulla voidaan saavuttaa parempi tiedon laatu lisäämällä sen ymmärrettävyyttä ja tarkkuutta (Preciado ym. 2005). Tavallisia visualisoinnin kohteita ovat esimerkiksi tilastolliset tulokset, lääketieteelliset ilmiöt tai koneiden toimintarakenteet. Visualisoinnin piiriin voidaan lukea animoinnista virtuaalitodellisuuteen liittyviä ilmiöitä, joten näin ollen visualisoinnin kehittäminen muodostaa hyvin kokonaisvaltaisen ja tärkeän osan käyttöliittymäsuunnittelua. (Saariluoma 2004, 116)

Preciado ym. (2005) kutsuvat tiedon epäintuitiivista saatavuutta dataongelmaksi ja ehdottavat nimenomaan tiedon visualisointia ratkaisuksi. Käytettävyyden parantamiseksi tehdyssä informaation visualisoinnissa olennaisinta on käsitteellisten viestien välittäminen (Saariluoma 2004, 116). RIA-verkkosovellukset vastaavat edellä kuvattuihin haasteisiin, sillä niissä yhdistyvät visuaalisesti rikas sisältö, joustavat käyttöliittymät ja perinteistä verkkosovellusta korkeampi toiminnallisuuden taso. (Perfetti & Spool, 2002) Usein myös positiivisen käyttäjäkokemuksen tavoittelulle on annettu suuri painoarvo, mikä näkyy yleisimmin huomattavina panostuksina RIA-verkkosovellusten audiovisuaalisessa toteutuksessa. Edellä mainitut käytettävyys ja käyttäjäkokemus esitellään tarkemmin seuraavassa luvussa.

RIA-verkkosovellusten toteutustekniikoiden mukanaan tuoma toiminnallisuus ja visuaalinen kapasiteetti mahdollistavat myös pelinomaiset elementit käyttöliittymässä. Vaikka RIA-verkkosovellus ei olisikaan varsinaisesti peli, se voi sisältää peleistä tuttuja piirteitä, kuten esimerkiksi kerronnallisia elementtejä ja interaktiivista grafiikkaa sekä tavallista enemmän käyttäjältä vaadittavia ohjaustoimenpiteitä. (Chao 2004) Näiden kautta voidaan vaikuttaa esimerkiksi sovelluksen opittavuuteen molempiin suuntiin ja parantaa käytön kokemukselli-

suutta. (Pagulayan & Steury 2004) Pelit ja vapaa-ajan harrastukset on usein tehty tarkoituksella laaja- ja syvärakenteisiksi, jotta ne olisivat vaikeita ja vaatisivat ajattelutyötä. Tällainen suunnittelu on oikeutettua sovelluksissa, joita käytetään silloin, kun on aikaa ja halua pohdiskeluun. (Norman 1991, 178-179) Yksittäiset ominaisuudet, jotka tekevät pelistä viihdyttävän, eivät kuitenkaan välttämättä toimi irrallaan pelien kontekstista. Peleistä suoraan otetut toiminnan metaforat johtavat myös harhaan, mikäli tavoite sovelluksessa ja pelissä ei ole sama. (Chao 2004)

Vaikka RIA-verkkosovellusten toteutustekniikat mahdollistavatkin mielikuvituksellisia ja omaperäisiä ratkaisuja, ovat käyttäjälle näkyvät RIA-ominaisuudet usein kuitenkin melko hienovaraisia käytön helpottamiseksi laadittuja toimintoja. Esimerkiksi kursorin kohdistaminen sivun tiettyyn kohtaan voi kasvattaa tekstikokoa tai muuttaa sen väriä. Toinen usein käytetty ominaisuus on mahdollisuus skaalata ja liikuttaa näytöllä näkyviä elementtejä, jolloin näkymän järjestely muistuttaa ikkunoiduista käyttöjärjestelmistä tuttuja toimintoja. (Paulson 2005) Ominaisuuskirjostaan huolimatta RIA-verkkosovellukset eivät ole ratkaisu kaikkiin haasteisiin ja niiden ideana ei ole myöskään korvata kaikkea perinteistä WWW-sisältöä. Esimerkiksi HTML-sivu toimii erinomaisesti yksinkertaisen tiedon jakamisen alustana. (Preciado ym. 2005)

### 3.3 RIA-verkkosovellusten toimintaperiaate

Useimmat RIA-verkkosovellukset vaativat pienen ohjelmiston lataamista käyttäjän koneelle, jonka avulla sovellukset toimivat. Kaikki ylimääräinen lataaminen on muodostanut pienen esteen tai ainakin merkittävän hidasteen takavuosina, kun verkkoyhteydet ovat olleet vaatimattomia tiedonsiirtokapasiteetiltaan. Laajakaistojen yleistymisen myötä ongelma on väistymässä ja rikassisältöiset ohjelmistot voivat sisältää lisäksi vaikkapa virtaustekniikalla haettavaa ääntä ja

videota. Edelleen työasemien ja palvelimien välinen ero laskentatehossa on ka-ventunut merkittävästi. Työasemat, kannettavat ja jopa kämmenlaitteet tarjoa-  
vat sulavaa äänen ja värikuvavideon toistoa. RIA-verkkosovellukset hyödyntä-  
vätkin perinteiseen asiakas-palvelin -arkkitehtuurimalliin verrattuna runsaasti  
asiakaspuolen laskentatehoa. Useat RIA-ratkaisut ovat esitystasolla toimivia  
käyttöliittymätoteutuksia. RIA-verkkosovellukset voidaan kytkeä esimerkiksi  
käyttöliittymäksi olemassa oleviin väliohjelmistoihin ja yhdistää tätä kautta  
edelleen taustatietojärjestelmiin. (Noda & Helwig 2005)

Paulsonin (2005) mukaan RIA-verkkosovellukset päihittävät nopeudessa perin-  
teisemmät verkkosovellukset, koska sivut reagoivat tarkemmin käyttäjän toi-  
miin, eivätkä sivujen latautumisesta aiheutuneet keskeytykset estä sovelluksen  
muiden osien käyttöä. Nopeuteen vaikuttaa myös se, että palvelin lähettää se-  
laimelle vain vähimmäismäärän tarvittavaa tietoa, koska kokonaisia sivunäky-  
miä ei tarvitse päivittää sivun yksittäisessä osassa tapahtuneiden muutosten ta-  
kia. RIA-verkkosovellus voi kutsua palvelinpuolen toimintoja pieninä inkre-  
mentteinä ja vastaanottaa tietoa osa kerrallaan. Päivitettävän näkymän sisältöä  
voidaan päivittää vain tarvittavilta osin. Palvelimelle lähetetyt kutsut eivät  
jähmetä myöskään koko selainta, sillä tiedonsiirto tapahtuu taustalla ja kohdis-  
tuu vain pieneen osaan kokonaisuudesta. Käyttäjä voi edelleen jatkaa sovelluk-  
sen muiden osien käyttöä. Lisäksi kutsu palvelimelle voidaan käynnistää mel-  
kein minkä tahansa käyttäjän toimen seurauksena, joita ovat esimerkiksi hiirellä  
osoittaminen, hiirellä klikkaaminen ja näppäinkomennot. (Gehtland, Galbraith  
& Almaer 2005, 6)

RIA-verkkosovellusten toteutusteknologioita ovat esimerkiksi AJAX (Asynch-  
ronous JavaScript And XML), Flash ja Java (Noda & Helwig 2005; Crane, Pasca-  
rello & James 2006, 28). AJAX:in etuna on, että sillä voidaan tehdä RIA-  
verkkosovelluksia, joihin ei käyttäjän näkökulmasta sisälly selaimen lisäksi

muita ohjelmistovaatimuksia. AJAX-toteutukset luokitellaan skriptipohjaisiin RIA-verkkosovelluksiin. (Farrell & Nezlek 2007) Flashin käyttö mielletään usein suppeasti WWW-sivuilla oleviin mainosbannereihin tai WWW-sivuihin upotettuihin videoihin. Se sisältää kuitenkin runsaasti kapasiteettia graafisesti rikkaiden ja runsaasti toiminnallisuutta sisältävien verkkosovellusten luomiseen. Java-teknologia on verkkosovelluksiin liittyen pioneerin asemassa, sillä jo kauan sitten kielellä tehtiin WWW-sivuihin upotettuja Java-sovelmia (Noda & Helwig 2005). Flash-pohjaiset ja useat Javaan perustuvat toteutukset luokitellaan selainlaajennokseen perustuviin RIA-verkkosovelluksiin (Farrell & Nezlek 2007).

Flash-ympäristön soveltuvuus RIA-verkkosovelluksille on saanut vahvistusta erityisen FLEX-nimisen RIA-palvelinympäristön myötä, josta käsin sovelluksia voidaan suorittaa ja ladata asynkronisesti. Asiakkaan koneella sijaitseva Flash-selainlaajennos toimii tällöin FLEX-pohjaisten verkkosovellusten suoritusympäristönä. (Adobe Systems Inc. 2007) Flex-ympäristössä käytetään erityistä MXML-merkkäuskieltä (*engl. Maximum eXperience Markup Language*). Se muistuttaa HTML-sivunkuvauskieltä, ja sen avulla voidaan toteuttaa sivunäkymiä deklaratiiivisesti. MXML ei kuitenkaan riitä yksinään, ja esimerkiksi liiketoimintalogiikka ohjelmoidaan usein Flash-sovelluskehittimistä tutulla ActionScript-kielellä. (Asleson & Schutta 2006, 11)

Ennen kuin Java-sovelmia tai varsinaisia Java-sovelluksia voidaan käyttää asiakas-koneella, samantyyppisesti kuin Flash-pohjaisten ratkaisuiden kanssa, täytyy käyttäjän ladata koneelleen erillinen suoritusympäristö. (Sun Microsystems Inc. 2005) Yksi Java-pohjainen RIA-verkkosovelluksen toteutusmalli on Java Web Start, jossa Java Network Launch -nimistä protokollaa käyttäen voidaan käynnistää itsenäisesti toimiva Java-sovellus internetselaimen kautta (Farrell & Nezlek 2007).



Jokaisella toteutusteknologialla on omat vahvuutensa ja heikkoutensa RIA-verkkosovellusten toteuttamisessa (TAULUKKO 2). On myös mahdollista yhdistellä edellä mainittuja tekniikoita. Esimerkiksi palvelimen sovelluslogiikka voidaan hoitaa Java-ohjelmoinnilla ja käyttöliittymä selaimen AJAX:illa. (Paulson 2005) Java on merkittävä taustateknologia myös Flex-toteutuksille, sillä Flex-palvelinohjelmisto toimii Java 2 Enterprise Edition -sovelluspalvelimen päällä (Asleson & Schutta 2006, 11). Oheisen taulukon merkitsemistapa on muutettu alkuperäislähteestä poiketen numeeriseksi vertailun helpottamiseksi.

**TAULUKKO 2.** RIA-verkkosovellusten toteutustekniikoiden vertailua mukailen lähdettä Noda & Helwig (2005).

0 huono / ei mahdollinen 1 keskinkertainen 2 hyvä	AJAX	Flash (Flex)	Java
Graafisuuden taso	1	2	1
Suoritusympäristön resurssien kulutus	2	1	0
Sovellusten lataamisajat	2	0	0
Tuki videolle ja äänelle	0	2	1
Sovelluksen ulkoasun yhtenäisyys eri suoritussympäristöissä	0	2	1
Toiminta ilman erillistä palvelinratkaisua	2	0	1
Sovelluskehityksen helppous	1	0	0
Pisteitä yhteensä	8	7	4

AJAX-pohjaisten RIA-verkkosovellusten graafinen taso on HTML-sivuja vastaavaa, ellei grafiikan esittämisessä hyödynnetä AJAX:in mahdollistamaa toi-

minällisuutta. Toiminnallisuus tuo myös grafiikan esittämiseen dynaamisuutta, jolloin AJAX:illa toteutetun RIA-sovelluksen graafisuuden tasoa voidaan pitää staattista HTML-sivua korkeampana. Java-perustaiset RIA-verkkosovellukset pääsevät graafisuuden asteella jokseenkin samalle tasolle AJAX-pohjaisten RIA-verkkosovellusten kanssa, joskin graafinen ote muistuttaa negatiivisessa mielessä enemmän perinteisiä työpöytäsovelluksia kuin visuaalista multimediaa. Java-toteutukset tukeutuvat usein käyttöjärjestelmäriippumattomaan valmiiseen Swing-käyttöliittymäkirjastoon. Flash-tekniikka puolestaan tukee nimenomaan graafisesti rikasta toteutusta ja Flash-pohjaiset RIA-verkkosovellukset näkyvät täysin samanlaisina katsottiinpa niitä minkä käyttöjärjestelmän kautta hyvänsä. Kukin käyttöjärjestelmä ja käytetty selain vaativat tosin Flash-selainlaajennoksen, jotta verkkosovelluksen näyttäminen ylipäätään on mahdollista.

Koska AJAX, toisin kuin Flash ja Java, ei vaadi erillistä selainlaajennosta tai suoritussympäristöä, on sen resurssien kulutus kaikkein vaatimattominta vertailuista tekniikoista. Kun loppukäyttäjän ei tarvitse asentaa erikseen ylimääräisiä ohjelmia järjestelmäänsä, on käyttö vaivattomampaa. AJAX-pohjaisissa RIA-verkkosovelluksissa joudutaan tosin tinkimään videon ja äänen käytöstä vertailun muihin toteutustekniikoihin verrattuna. Vähäistä resurssien kulutusta tukee myös toiminta ilman erillistä palvelinratkaisua. AJAX-toteutuksen voi ajatella olevan myös halvempi, ainakin aloituskustannuksiltaan, jos rahaa ei tarvitse sijoittaa erilliseen AJAX:ille pyhitettyyn palvelimeen ja sen ohjelmistoihin. Toisaalta myös AJAX:illa tehtävää sovelluskehitystä pidetään helpoimpana vertailuista. Tästä huolimatta esimerkiksi Smith (2006) perustelee AJAX:in yhä suhteellisen vähäistä käyttöä sillä, että AJAX-pohjaisten verkkosovellusten kehittäminen on vaikeaa. Syitä tähän ovat JavaScript-kielen rajoitukset, selainten yh-

teensopivuusongelmat sekä hyvien suunnittelu-, kehitys- ja virheenetsintätyökalujen vähäisyys.

### 3.4 AJAX

AJAX tulee sanoista Asynchronous JavaScript and XML, joskin se sisältää XML-merkkaukielen ja JavaScript-ohjelmointikielen lisäksi myös muita teknologiakomponentteja. AJAX:in käyttämiä muita teknologioita ovat CSS (Cascading Style Sheets), DOM (Document Object Model), XSLT (Extensible Stylesheet Language Transformations) ja XMLHttpRequest -olio, joka on toteutettu eri selaimissa eri tavoin. AJAX on siis oikeastaan kokoelma useita yksittäisiä teknologioita, joiden juuret juontavat pisimmillään aina 1990-luvulle. AJAX:illa on paljon yhteistä Microsoftin vuonna 1997 esittelemän DHTML-tekniikan (Dynamic HTML) kanssa. (Paulson 2005) Myös DHTML on kokoelma yhteisen yläkäsitteen alle paketoituja teknologioita, jotka ovat HTML, CSS, JavaScript ja DOM (Microsoft 2007).

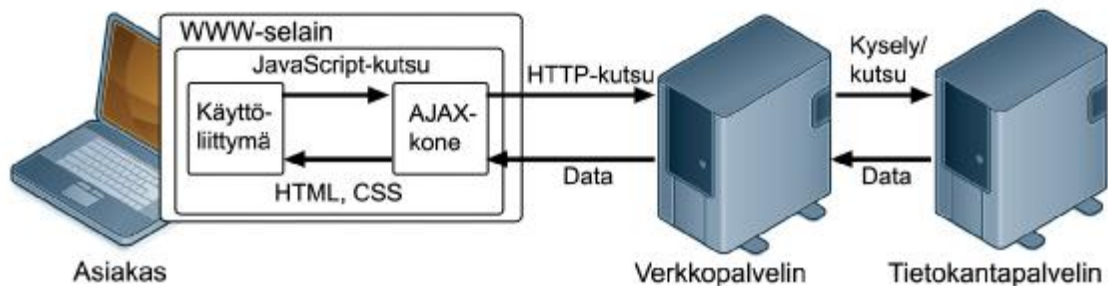
DOM:in avulla sivunäkymää voidaan käsitellä joukkona olioita, joilla on yksittäisiä dynaamisesti muokattavissa olevia ominaisuuksia (Zakas, McPeak & Fawcett 2006). DOM kuvaa dokumentin yksittäiset osat, niiden käyttäytymisen ja ominaisuudet sekä näiden väliset suhteet (Asleson & Schutta 2006, 36). DOM esitetään rakenteisena puuna, jonka solmut ovat verkkosivun osia. Muokkaamalla DOM:ia JavaScriptillä päästään käsiksi dokumentin rakenteeseen ja sisältöön, joita voidaan näin päivittää dynaamisesti. (Crane, Pascarello & James 2006, 45) XML toimii tiedon siirron formaattina, jota voidaan muuntaa tarvittaessa XSLT:n avulla eri formaatteihin, kuten selaimessa esitettäväksi sivunkuvauskieleksi. CSS-tyylikielen avulla määritellään rakenteisen dokumentin ulkoasu. XMLHttpRequest on puolestaan olio, joka huolehtii tiedonvälityspyynnöistä asiakkaan ja palvelimen välillä. (Zakas, McPeak & Fawcett 2006) AJAX

voi toimia lukuisten palvelinratkaisuiden kanssa yhdessä ja palvelinohjelmointi voi olla toteutettu esimerkiksi J2EE-, .NET- tai PHP-alustalla (Asleson & Schutta 2006, 13). JavaScriptin avulla ohjelmoidaan myös niin kutsuttu AJAX-kone (*engl. AJAX engine*), joka sitoo esiteltyt teknologiat AJAX-kokonaisuudeksi palvelinpuolen ohjelmointiratkaisuja lukuun ottamatta (Zakas & Fawcett 2006). JavaScript on näin ollen AJAX-verkkosovellusten ydin ja koossapitävä voima (Crane, Pascarello & James 2006, 36).

Uusinta tekniikka AJAX-kokonaisuudessa edustaa tiedonsiirtoon käytetty XMLHttpRequest-rajapintaolio, joka on ollut käytettävissä Microsoftin Internet Explorer 5 -internetselaimen myötä vuodesta 1999 lähtien. Ongelmana on kuitenkin ollut se, että tekniikkaa tukeva selainkanta oli pitkään rajoittunut juuri mainittuun selaimen ja sen seuraajiin. Nytemmin ratkaisu on saanut uutta tuulta purjeisiinsa nimenomaan laajentuneen selaintuen ansiosta. (Asleson & Schutta 2006, 13) XMLHttpRequest ei ole yleisyydestään huolimatta päässyt W3C:n suositukseksi. W3C:n piirissä on kehitetty vastaava mekanismi nimeltään Load and Save, joka kuvataan DOM3-spesifikaatiossa. Standardoimaton XMLHttpRequest on kuitenkin kehittäjäyhteisölle luonteva valinta, sillä Load and Save -suositus on tällä hetkellä tuettuna vain marginaalisesti käytetyssä Opera-selaimessa. XMLHttpRequestin standardoimattomuus aiheuttaa ongelmia ja lisätyötä sovelluskehittäjille, sillä sen toteuttamiseksi ei ole olemassa yhtä määriteltyä tapaa. Kysymyksiä aiheuttaa myös se, taipuuko W3C suosituksiinsa luopumaan omasta Load and Save -määrityksestään tai lakkaavatko selainvalmistajat tukemasta nykyistä käytäntöä, mikä rikkoisi suuren joukon RIA-verkkosovelluksia selainpäivitysten myötä. (Weiss 2006)

Perinteisessä verkkosovellusten toimintomallissa selain lähettää palvelimelle kutsun ja hoitaa myös vastauksen käsittelyn. AJAX-toteutuksissa selaimen ja palvelimen välissä on välitaso, niin kutsuttu AJAX-kone, joka huolehtii tapah-

tumien kulusta (KUVIO 4). AJAX-kone on pohjimmiltaan vain JavaScript-olio tai -funktio, jota kutsutaan, kun tarvitaan toimintaa palvelimen ja asiakkaan välillä. Kun AJAX-sovelluksen toiminto aktivoidaan, menee JavaScript-kutsu ensin AJAX-koneelle, jonka kautta pyynnöt kulkevat sopivassa järjestyksessä verkkopalvelimelle perinteisistä verkkosovelluksista tutulla HTTP-kutsulla. Kutsu tapahtuu kuitenkin asynkronisesti taustalla eikä AJAX-koneen toiminta jähmetä muun koodin suoritusta. Näin sovelluksen käyttäminen voi tiedonsiirto-operaatioista huolimatta jatkua ilman katkoksia. Kutsuihin vastaava palvelin on konfiguroitu lähettämään perinteisten sivuelementtien, kuten kuvien ja html-tiedostojen, sijaan yleensä XML-muotoista dataa, jonka AJAX-kone tulkaa. Tuloksin tulokset esitetään esimerkiksi HTML- ja CSS-muotoisina ja käyttöliittymänäkymään aiheutetaan pyydetyn toiminnon mukaisia muutoksia. Etuina ratkaisussa ovat siirrettävän tiedon vähäinen määrä ja saavutettu nopeus. (Zakas, McPeak & Fawcett 2006)



KUVIO 4. AJAX-verkkosovelluksen toimintomalli Zakasia, McPeakia ja Fawcettia (2006) mukailten.

Koska AJAX-verkkosovelluksen toiminta ei ole sidottu käyttäjän nimenomaisiin pyyntöihin, voi sovellus reagoida käyttäjän liikkeisiin joustavasti. Esimerkiksi hakukenttään kirjoittaminen voi aktivoida tapahtuman ja käynnistää tiedonsiirron taustalla, samalla kun käyttäjä edelleen kirjoittaa tekstiä. Tässä tapauksessa

käyttäjä voisi esimerkiksi kirjoitettuaan ainoastaan kirjaimet k, a ja t saada palvelimen vastauksen perusteella valittavakseen joukon sopivia kat-alkuisia sanoja, kuten katti, kattila ja katto.

### 3.5 Yhteenveto

Web 2.0 on käsite, jonka kautta pyritään erottamaan uudet WWW-toteutustekniikat ja toimintatavat aikaisemmin WWW-ympäristössä käytetyistä toimintamalleista ja tekniikoista. Uuteen Web 2.0 -kategoriaan luokiteltavat sovellukset voivat muistuttaa enemmän työpöytäsovelluksia kuin verkkosivuja, paitsi ulkoasultaan myös toiminnoiltaan. Useita tällaisia toisen aallon verkkosovelluksia kutsutaan nimellä RIA-verkkosovellukset. Nimitys RIA tulee englanninkielisistä sanoista Rich Internet Application. Näille ”rikkaille internetsovelluksille” ei ole olemassa vakiintunutta suomenkielistä vastinetta. RIA-verkkosovelluksilla pyritään hyvään tekniseen ja toiminnalliseen laatuun. Tällöin sovelluksen käytön tulisi olla miellyttävämpää ja lopputuloksen parempi kilpaileviin sovelluksiin verrattaessa.

RIA-verkkosovelluksille on keskeistä tiettyjen teknologioiden hyödyntäminen ja sitä kautta saavutetut edut, kuten sisällön lataaminen asynkronisesti internet-selaimen välimuistiin ja asiakaspään laskentatehon hyödyntäminen sekä näistä seuraavat nopeat vasteajat ja sulavasti päivittyvät dynaamiset grafiikat. RIA-verkkosovellukset tukevat myös perinteisiä verkkosovelluksia paremmin multimedialla ja informaation visuaalista esittämistä.

RIA-verkkosovelluksissa edes useampivaiheista kokonaisuutta ei ole välttämätöntä jakaa useille erillisille sivuille, mikä tekee verkkosovelluksista yhtenäisempiä ja helpompia käyttää. RIA-verkkosovellusten etuna on sivunäkymien dynaaminen päivitysmahdollisuus, jolloin vain tarvittavat osat päivittyvät tar-

peita vastaavaksi. Palvelimelle lähetetyt kutsut eivät jähmetä myöskään koko selainta, sillä tiedonsiirto tapahtuu taustalla ja kohdistuu vain pieneen osaan kokonaisuudesta, jolloin käyttäjän ei tarvitse olla tietoinen asiakas-palvelin-arkkitehtuurista johtuvasta liikennöinnin mekaniikasta.

Yleisiä RIA-verkkosovellusten toteutusteknologioita ovat esimerkiksi AJAX, Flash ja Java. Jokaisella toteutusteknologialla on omat vahvuutensa ja heikkoutensa RIA-verkkosovellusten toteuttamisessa, joskin on myös mahdollista yhdistellä edellä mainittuja tekniikoita parhaan lopputuloksen saavuttamiseksi. AJAX on noussut erityisen keskeiseksi toteutusteknologiaksi, koska se ei vaadi erillistä selainlaajennosta tai suoritusympäristöä, sen käyttöönotossa on alhaiset kustannukset ja sovelluskehitystä pidetään suhteellisen helppona. Puhtaasti AJAX-pohjaisissa RIA-verkkosovelluksissa videon ja äänen käytön ominaisuudet kuitenkin häviävät muihin esiteltyihin toteutustekniikoihin vertailtaessa.

AJAX on pohjimmiltaan kokoelma useita yksittäisiä teknologioita. Nimitys tulee sanoista Asynchronous JavaScript and XML, joskin se sisältää XML-merkkaukielen ja JavaScript-ohjelmointikielen lisäksi myös muita teknologiakomponentteja. AJAX:in käyttämiä muita teknologioita ovat CSS, DOM, XSLT ja tiedonsiirrosta huolehtiva XMLHttpRequest -olio. JavaScript on AJAX-verkkosovellusten ydin, jonka avulla ohjelmoidaan myös niin kutsuttu AJAX-kone, joka sitoo esiteltyt teknologiat AJAX-kokonaisuudeksi palvelinpuolen ohjelmointiratkaisuja lukuun ottamatta

## 4 VERKKOSOVELLUKSIEN KÄYTETTÄVYYDEN ARVIOINTI

Käyttäjästävällisyyteen panostava suunnittelu on tärkeää kaikissa ihmisen käyttöön tarkoitetuissa tuotteissa ja erityisen tärkeää se on, kun käyttäjän hyväksyntä tuotetta kohtaan näyttelee olennaista roolia (SFS-EN ISO 9241-1 1998). Se on tärkeää myös siitä syystä, että ihmiset käyttävät koko ajan kaikenlaisia hyödykkeitä pelkän näppituntuman varassa ymmärtämättä, miten ne oikeasti toimivat. Vain harvat viitsivät lukea käyttöohjeita. (Krug 2006, 26–28) Jos verkkosivu ei tunnu satunnaisesta selailijasta käytettävältä tai muutoin miellyttävältä, ovat seuraavalle sivustolle siirtymisestä aiheutuvat kustannukset erittäin vähäiset (Nielsen 2000, 161).

Verkkosovellusten käytettävyyttä ei voida pohtia irrallaan käyttöliittymästä, koska käyttöliittymä on ihmisen ja koneen kohtaamispaikka ja usein se näkyvä ja kosketeltava osa käytettävyyttä, johon kohdistuu sekä järjestelmän suunnittelijan huomio että käyttäjän kritiikki. (Tarkka 2000, 106; Mielonen & Hintikka 1998) Tässä työssä käytettävyyttä käsitelläänkin verkkosovellusten käyttöliittymän arviointiin soveltuvien osien.

Seuraavassa alaluvussa esitellään termi käytettävyys ja sitä seuraavassa alaluvussa käytettävyyteen läheisesti liittyvä termi käyttäjäkokemus. Lisäksi näissä alaluvuissa esitellään verkkosovellusten käytettävyyteen liittyviä tekijöitä, joita on havaittu verkkosovellusten ja käyttöliittymätutkimuksen yhteydessä. Alaluvussa kolme esitellään käytettävyyden ja käyttöliittymän arviointiin kehitettyjä asiantuntija-arvioinnin menetelmiä. Neljännessä alaluvussa esitellään soveltavan esimerkin avulla Nielsenin Kymmenen heuristiikan listan käyttöä. Viides alaluku on yhteenveto.



## 4.1 Käytettävyys

Tässä alaluvussa tarkastellaan moniulotteista käytettävyyden käsitettä, jota on määritelty useiden tutkijoiden voimin ja useista lähtökohdista ilman konsensusta yksikäsitteisestä määritelmästä (Ovaska ym. 2005, 3). Sinkkosen ym. (2002, 19) mukaan *käytettävyys* on kokonaisuudessaan menetelmä- ja teoriakenttä, jonka kautta käyttäjän ja laitteen yhteistoiminta pyritään saamaan tehokkaammaksi ja miellyttävämmäksi. Käytettävyys hyödyntää kognitiivisen psykologian sekä ihmisen ja koneen vuorovaikutuksen tutkimusta.

Käytettävyyttä tutkittaessa pääosassa on käyttäjä ominaisuuksineen. Käyttäjällä on synnynnäisiä fysiologisia ja psyykkisiä ominaisuuksia kuten aistit, muistirakenteet ja perustarpeet. Näiden päälle ovat rakentuneet suhteellisen pysyvät kulttuurisidonnaiset asiat, esimerkiksi kieli, normit ja tavat. Käyttäjältä odotetaan ja hänellä on erilaisia konventioita liittyen teknisiin toimintaympäristöihin. (Sinkkonen ym. 2002, 24–27) Ihmisen toimintaan vaikuttavia kulttuurielementtejä ovat Sinkkosen ym. (2002, 27–29) mukaan myös muoti ja erilaiset alakulttuurit. Alakulttuuria edustavat osaltaan myös käyttöliittymä- ja WWW-kulttuuri. Lisäksi toiminnan vaikuttimina ovat tehtävät, yksilölliset kyvyt ja käyttötilanne. Nämä vaihtelevat seikat ovat sellaisia, joita on pohdittava jokaisessa tuotteen suunnitteluprosessissa erikseen.

Vaikka käytettävyyden tutkimusperinne painottaa kognitiivisia seikkoja, käyttöliittymä ja sitä ympäröivä verkkosovellusympäristö sisältävät kuitenkin enemmän ulottuvuuksia. Verkkosovellus- ja uusmediasuunnittelussa puhasoppisen käytettävyyden näkökulman ohi ajaa usein ajatus käyttäjäkokemuksesta. Suunnittelun kohteeksi nousee tuotteen tai palvelun välittämä kokonaiselämyksellisyys. Verkkosovellusten käyttöliittymäsuunnittelussa voidaankin usein sanoa edettävän käyttäjälähtöisyyden sijasta asiakaslähtöisyyden ehdoilla.

(Tarkka 2000, 106–107) Constantinen ja Lockwoodin (2002) mukaan verkkosovelluksien kehitystyössä on huomioitu useammin loppukäyttäjän kokemus perinteisiin tietojärjestelmiin verrattuna.

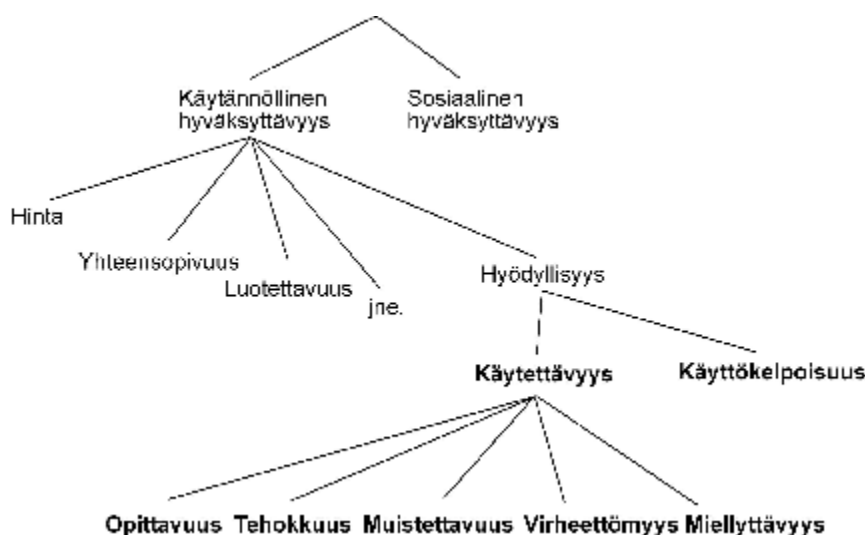
Useissa julkaisuissa ja erityisesti puhekielessä *ihmisen ja koneen vuorovaikutus* (*engl. Human-Computer-Interaction, HCI*) ja käytettävyys nähdään pitkälti samana asiana. Teoriassa voidaan todeta, että kun tarkastelunäkökulmana on ihmisen ja koneen vuorovaikutus, ei ihmistä käsitellä organisaation osana, työntekijänä eikä tahtovana toimijana. (Sinkkonen ym. 2002, 20) Ihmisen ja koneen vuorovaikutus ei kuitenkaan ole vain mekaanista toimintaa. Ihmisen oma tahto ja inhimilliset piirteet ovat käytettävyyden kokemuksen muodostumisessa tärkeitä asioita.

Krug (2006, 5, 11) pelkistää määrittelyssään käytettävyyden tarkoittavan sitä, että jokin toimii hyvin ja että myös vähemmän kokenut henkilö pystyy käyttämään jotain – vaikkapa verkkosovellusta – sen suunniteltuun käyttötarkoitukseen ilman toivottomia turhautumisen tunteita. Hänen johtoajatukseensa onkin, että käyttäjää ei saisi pakottaa ratkaisemaan ylimääräisiä käyttöliittymän käyttöön liittyviä pulmia, vaan että käyttöön liittyvän toiminnan tulisi olla ilmi selvää. Myös Lauesen (2005, 3) pohjustaa käytettävyydemäärittelyä yleistäen: jotkin tietojärjestelmät ovat helppoja käyttää, joten niiden käytettävyyden taso on korkea. Toisten järjestelmien käyttö on painajaismaista ja niiden käytettävyyden taso on matala. Kyseiset määritelmät eivät ole analyttisiä tai kattavia, mutta ne kuvaavat käytettävyyttä varsin kansantajuisesti.

Nielsenin (1993, 25-26) mukaan käytettävyys jaetaan viiteen osa-alueeseen: opittavuuteen, tehokkuuteen, muistettavuuteen, virheettömyyteen ja miellyttävyyteen. Nielsen muistuttaa, että käytettävyys ei ole käyttöliittymästä havaittava yksiulotteinen ominaisuus, vaan se koostuu useista tekijöistä. ISO-standardissa

9241-11 (ISO 1998) ollaan samaa mieltä tehokkuudesta ja miellyttävyydestä, mutta lisäksi siihen on otettu mukaan myös tuottavuuden näkökulma. Käytettävyys syntyy tällöin siitä, kuinka hyvin käyttäjä voi tuotteen avulla saavuttaa tavoitteensa tuloksellisesti, tehokkaasti ja käyttäjää tyydyttävällä tavalla tietyssä käyttökoneksissa. Käytön kontekstin muodostavat käyttäjän omien ominaisuuksien lisäksi tehtävän laatu, laitteisto ja ympäristö.

Nielsenin määritelmässä (1993, 25) sekä käytettävyys että käyttökelpoisuus muodostavat yhdessä tuotteen hyödyllisyyden, joka on puolestaan yksi hyväksyttävyyden osatekijöistä (KUVIO 5). Käyttökelpoisuus voidaan selvittää kuitenkin vasta käytännön toimissa, eikä sitä pystytä juurikaan havainnoimaan käytettävyystutkimuksen menetelmillä. Tuote voi olla toisaalta äärimmäisen käytettävä, mutta ei kokonaisominaisuuksiltaan käyttökelpoinen tehtävän suorittamiseen. Dicksin (2002) mukaan olisikin syytä pohtia aina myös kunkin tuotteesta löytyvän toiminnon käyttökelpoisuutta.



KUVIO 5. Käytettävyyden osatekijät Nielsenin mukaan (Nielsen 1993, 25)

Käytettävyyden osatekijät voivat olla myös ristiriitaisia vaatimuksiltaan käyttöliittymäsuunnittelussa. (Nielsen 1993, 42) Esimerkiksi helposti opittava käyttöliittymä voi lukuisine käyttäjää ohjaavine ohjeistuksineen ja välivaiheineen olla kaukana tehokkaasta. Käyttöliittymäsuunnittelun alkuvaiheessa tulisikin määrittellä käyttäjien tarpeet ja ohjelmistolle suunnitellut tyypilliset käyttötavat (Nielsen 1993, 73). Käytettävyyttä voidaan tutkia myös kaksijakoisella tavalla, jossa tarkastellaan erikseen tuotteen lähestyttävyyttä sen käyttöönoton yhteydessä ja erikseen toiminnallisessa käyttötilanteessa (Keinonen 2000, 104).

Käytettävyyden yhteydessä puhutaan usein myös *saavutettavuudesta* eli *esteettömyydestä* (*engl. accessibility*). Käytössä on myös jokseenkin samaa tarkoittava englanninkielinen termi Design for All, eli ”kaikille tarkoitettu suunnittelu”. Esteettömässä suunnittelussa on lähtökohtana huomioida erityisryhmiä, joihin kuuluu aistiensa, motoriikkansa tai kognitiivisten kykyjensä suhteen poikkeavia ihmisiä. (Ovaska ym. 2005, 4-5) Esteettömyyden ja käytettävyyden suhde ei ole kuitenkaan tarkkarajainen ja niistä löytyy hyvin paljon yhteistä leikkauspintaa. Erityisryhmille suunniteltu tuote toimii yleensä suotuisalla tavalla myös tavalliselle käyttäjälle. (Hyppönen 2000, 107-108) Kaikille tarkoitettu suunnittelu ei tarkoita kuitenkaan sitä, että tehtäisiin sananmukaisesti jokaiselle mahdolliselle käyttäjälle sopiva tuote tai palvelu. Tuotteen suunnittelu optimaaliseksi koko kuluttajakunnalle ei ole järkevää taloudellisesti eikä myöskään käytettävyyšnäkökulmasta. (Hyppönen 2000, 109)

## 4.2 Käyttäjäkokemus

*Käyttäjäkokemus* on monesta muuttujasta riippuvainen ja subjektiivinen tutkimuskohde, eikä sitä voida mitata samoin kvantitatiivisin menetelmin kuin esimerkiksi joitain käytettävyyteen liittyviä tekijöitä käyttäjätestauksessa. Käyttäjäkokemuksen subjektiivisuudesta huolimatta tiettyjä siihen liittyviä lainalai-

suuksia on pystytty nostamaan esiin, kuten esimerkiksi esteettisen suunnittelun tuoma tuki käytettävyyden kokemukselle sekä tämän heijastusvaikutukset käyttäjän kokemukseen ja kokonaistyytyväisyyteen – joskin tähän liittyvät asiat on perinteisessä käytettävyydestutkimuksessa jostain syystä kyetty enimmäkseen sivuuttamaan. (Tractinsky, Katz & Ikar 2000)

McNamara ja Kirakowski (2006) ehdottavat teknologiatuotteiden arviointiin kolmijakoa, jonka mukaan arvioinnin kohteena oleva objekti jaetaan kolmeen eri tekijään: varsinaiseksi tuotteeksi, johon toiminnallisuus liittyy, tuotteen käytettävyyteen ja tuotteen käytöstä syntyvään käyttäjäkokemukseen. Jokainen kolmesta tekijästä on ainutkertainen, mutta toisista riippuvainen ja toisiin vaikuttava asia tuotetta käytettäessä ja arvioitaessa.

Tuotetta ja sen toiminnallisuutta arvioitaessa tutkitaan pelkistetyimmillään mitä tuote tekee. Tuotteen käytettävyyttä tutkittaessa katsotaan pääsääntöisesti sitä, suorittaako tuote käyttäjän tahdon mukaisesti siihen suunniteltuja toimintoja. Käyttäjäkokemusta kartoitettaessa joudutaan sen sijaan puntaroimaan hieman hankalampia ja moniulotteisempia asioita, kuten esimerkiksi käyttäjän tuntemuksia ja kokemuksia, mitä tuotteen käyttö merkitsee käyttäjälle, onko tuotteen käyttö tärkeää käyttäjälle ja sopiiko se käyttäjän arvoihin ja tavoitteisiin. (McNamara & Kirakowski 2006)

Käyttäjäkokemuksen keskiössä ovat käyttäjä ja tuote, jotka sijaitsevat aina jossakin tietyssä ympäristössä ja edelleen sosiaalisessa ja kulttuurisessa tilassa (Forlizzi & Ford 2000). Käyttäjäkokemus voi olla myös sosiaalisessa kontekstissa koettu ja jaettu, kuten esimerkiksi usean käyttäjän kesken pelattavassa pelissä (Forlizzi & Battarbee 2004). Tuote vaikuttaa käyttäjäkokemukseen ominaisuuksillaan, kuten toiminnallisuudella, muotokielellä ja estetiikalla (Forlizzi & Ford 2000). Mutkikkaat käyttäjäkokemukseen liittyvät tunteet vaativat kognitii-

visia prosesseja syntyäkseen ja vastaavasti kognitio tarvitsee tunteita ohjaamaan toimintaa, kuten esimerkiksi päätöksenteossa. Keskeisimmät tunteet tuotteisiin liittyen koskevat sen vetovoimaisuutta ja viehättävyyttä, eli pidetäänkö tuotteesta vai inhotaanko sitä. (Hassenzahl 2004)

Epämiellyttävien tuotteiden yhteydessä epämiellyttävyiden syiksi mainitaan usein huonoksi koettu estetiikka sekä puutteet käytettävyydessä ja toimivuudessa. Seurauksena on monenlaisia tunteita ärsyyntymisestä suoranaiseen aggressioon ja petetyksi tulemisen tunteeseen. (Mattelmäki & Battarbee 2000, 145) Perustunteilla, kuten vihalla, rakkaudella, pelolla ja ilolla on lajinkehityksellinen perusta. Esteettinen ja älyllinen mielihyvä ovat puolestaan konstruoituja tunteita. Konstruoidut tunteen vaihtelevat ihmisestä ja kulttuurista riippuen ollen sosiaalisia ja ideologisia pikemmin kuin biologisia. (Koskinen 2000, 133) Esteettisen suunnittelun onnistuminen onkin hyvin riippuvainen siitä, että pystytään huomioimaan minne ja mille yleisölle tuote on suunnattu (Mattelmäki & Battarbee 2000, 145).

Yksittäinen kokemus muodostuu lukemattomista pienemmistä kokemuksista, jotka ovat riippuvaisia ilmenemisyhteydestä, tilanteeseen liittyvistä ihmisistä ja käytettävistä artefakteista. Esimerkiksi retkeilyyn liittyvä kokemus muodostuu patikoinnista, vuorovaikutuksesta patikointiseurana kanssa sekä erilaisten tuotteiden, esimerkiksi teltan tai retkikeittimen, käytöstä. Käyttäjäkokemuksen elementtejä ei voida niiden moninaisuuden takia täysin säädellä, joten myös sattunnaistekijöillä on aina merkittävä asema. (Forlizzi & Ford 2000)

Forlizzin ja Fordin (2000) mukaan käyttäjäkokemus voi olla tiedostamaton (*engl. subconscious*), kognitiivinen (*engl. cognitive*), kerronnallinen (*engl. narrative*) tai tarinallinen (*engl. storytelling*). Tiedostamaton käyttäjäkokemus on automaattista ja sujuvaa. Näitä ovat rutiininomaiset tavat toimia tuttujen tai helppokäyt-

töisten tuotteiden kanssa. Kognitiivinen kokemus vaatii puolestaan korkeampaa ajattelutyötä. Kognitiivinen luovaa ongelmanratkaisukykyä vaativa tilanne on esimerkiksi vuorovaikutus uusien tuotteiden tai ympäristöjen kanssa. Kerronnallinen kokemus on tiedostettu ja sitä voidaan kuvailla sanallisesti. Tarinallinen kokemus on kokijalleen henkilökohtainen ja siihen liittyvät esimerkiksi konteksti, aikaisemmat kokemukset sekä vallitseva tunnetila.

Käyttäjäkokemuksen tyyppi voi myös vaihtua tasolta toiselle. Esimerkiksi kerronnallinen kokemus, mainoksen katselu, voi aktivoida ajattelua ja pakottaa kognitiiviselle tasolle kyseenalaistamaan mainos tai omat asenteet. Henkilökohtaisesti koskettava kokemus taas voi muuttua tarinalliseksi. Esimerkiksi hyvä tuote, vaikkapa onnistunut verkkosovellus, tarjoaa muistettavan tarinan, joka on helppo omaksua ja josta kerrotaan mielellään eteenpäin. (Forlizzi & Ford 2000)

Käyttäjäkokemusta ei voida suunnitella, mutta suunnittelutyötä sen edesauttamiseksi voidaan tehdä. Suunnittelija voi luoda siis vain mahdollisuuksia positiivisen käyttäjäkokemuksen syntymiseen. Jos tuote tietyssä tilanteessa täyttää käyttäjän sen hetkiset tarpeet, luo se edellytykset positiivisiin tuntemuksiin ja positiiviseen käyttäjäkokemukseen. Ihmiset jakavat joukon perustarpeita, joita voi käyttää suunnittelussa lähtökohtana. Usein tarve liittyy tiettyjen asioiden manipuloimiseen eli haluun saada jokin asia hoidetuksi. Tarve voi liittyä myös esimerkiksi stimulaatioon ja itsensä kehittämiseen tai itsensä ilmaisemiseen. (Hassenzahl 2004)

Vaikka tuote olisi helppokäyttöinen ja käytettävyydestein osoitettu hyväksyttäväksi, se ei takaa tuotteen menestystä eikä anna takuuta siitä, että tuote olisi merkityksellinen käyttäjälle ja koskettaisi tunnetasolla (Mattelmäki & Battarbee 2000, 161). Tutkittaessa tuotteiden vastaanottoa markkinoilla on suoritettava

erityinen markkinatutkimus, esim. kyselyitä ja käyttäjätestauksia hyödyntäen. (Brooks 1994, 256)

### 4.3 Käytettävyyden ja käyttöliittymän arviointiin kehitettyjä menetelmiä

Tässä alaluvussa tutustutaan käytettävyyden asiantuntijälähtöiseen arviointiin. *Asiantuntija-arviota* kutsutaan monella eri nimellä: esimerkiksi heuristista arviointia (*engl. heuristic evaluation*) ja asiantuntija-arviointia (*engl. expert evaluation*) käytetään monesti synonyyminä (Korvenranta 2005, 111). Asiantuntija-arviot ovat analyyttisiä menetelmiä, joissa käytettävyyсарviointi on ennakoivaa. Tällöin tutkitaan, mikä sovelluksen ominaisuus tai toiminto tulee olemaan mahdollinen käytettävyysoongelma. (Perälä 2005, 301) Käytettävyyden asiantuntija-arviointi pitää sisällään erilaisia menetelmiä, joita käyttäen ja yhdistellen on mahdollista arvioida käyttöliittymien käytettävyyttä ja löytää niistä ongelmia sekä tarjota parannusehdotuksia (Ivory & Hearst 2001). Pelkkä ongelmien löytäminen ei vielä sinällään paranna käytettävyyttä ja tuloksista olisikin syytä johtaa suunnitteluvirheitä korjaavat suunnittelumuutokset.

Käytettävyyteen vaikuttavat suunnitteluvirheet aiheuttavat katkoksen käyttäjän ja sovelluksen välillä ja johtavat negatiivisiin muutoksiin käyttäjän käyttäytymisessä ja suoriutumisessa. (Lavery, Cockton & Atkinson 1997) Käytettävyyden arviointi liittyy usein laajempaan viitekehykseen eli käyttöliittymän suunnitteluprosessiin. Suunnitteluprosessi on yleensä luonteeltaan iteratiivinen ja koostuu suunnittelusta, prototyyppien kehityksestä sekä arvioinnista. (Ivory & Hearst 2001)

Aina ei ole mahdollista eikä kannattavaa ottaa testikäyttäjää mukaan arviointityöhön. Tuote voi olla esimerkiksi niin keskeneräinen, ettei testikäyttäjän mukaan ottaminen ole mielekästä. Testikäyttäjän kanssa tehtävä arviointi voi olla



joissain tilanteissa myös liian aikaa vievää. Arviointiin onkin kehitetty menetelmiä, joiden avulla käytettävyyttä voidaan arvioida nopeasti ilman ulkopuolisia henkilöitä. (Korvenranta 2005, 111)

Testattavaan järjestelmään voidaan kehitystyön eri vaiheissa soveltaa myös useampaa eri testausmenetelmätyyppiä, joilla kullakin löydetään hieman erilaisia käytettävyysongelmia. Erilaisten ongelmien löytyminen ei ole sikäli yllättävää, että myös yhtä arviointimenetelmää käyttävät eri testajat löytävät usein toisistaan poikkeavia ongelmia. (Ivory & Hearst 2001) Toisistaan poikkeavat ongelmat voivat olla myös väärää hälytyksiä ja toisaalta arvioijalta voi jäädä osa ongelmista havaitsematta (Perälä 2005, 302). Käytettiinpä mitä käytettävyyden arvioinnin menetelmää hyvänsä, tosielämän tiukasti aikataulutetuissa ohjelmistotuotannoissa osa sovelluksesta jää helposti toisia kohtia huonommalle testaukselle (Brooks 1994, 257).

Wixon, Jones, Tse ja Casady (1994, 79) käyttävät asiantuntija-arviointien yhteydessä linssistä johdettua metaforaa. Käytettävyyden läpikäynti asiantuntija-arvioinnin menetelmin toimii kuin linssi, jonka kautta käyttöliittymiä tarkastellaan. Linssin säädöistä riippuen vain osa kohteesta on mahdollista pitää tarkassa polttopisteessä. Linssiä säädetään sen mukaan, mitä piirteitä kohteena olevasta kokonaisuudesta halutaan tarkastella. Metafora kärjistää myös kysymykset käytettävyyksmenetelmien keskinäisestä vertailusta ja käytettävyyksmenetelmien käyttömahdollisuuksien rajoista toteamalla, ettei tähtitaivastakaan katsota mikroskoopilla, vaikka mikroskooppikin on optinen väline. Arvioinnin tavoitteet tulisi siis olla selvillä ennen arviointivälineiden valintaa. Käytettävyyden asiantuntija-arvioinnin menetelmien tehokkuuksia vertailtaessa on otettava mukaan laskuihin myös ne tavoitteet, joita varten menetelmä on kehitetty (Wixon ym. 1994, 99). Seuraavissa ala-alaluvuissa tutustutaan kolmeen käytettä-

vyyden asiantuntija-arvioinnin menetelmään, jotka sisältävät jossain määrin myös yhteisiä piirteitä.

#### 4.3.1 Kognitiivinen läpikäynti

*Kognitiivinen läpikäynti* on arviointimenetelmä, jolla selvitetään tuotteen käytettävyyttä ilman loppukäyttäjää joko yhden henkilön voimin tai ryhmässä. Menetelmä soveltuu käytettäväksi erityisesti ohjelmiston kehitysprosessin alkuvaiheissa. (Wharton, Rieman, Lewis & Polson 1994, 108) Kognitiivisessa läpikäynnissä yritetään mallintaa käyttäjän ajatuksia ja toimintaa, kun hän käyttää käyttöliittymää ensimmäistä kertaa. Tavoitteena on jäljitellä, kuinka käyttäjä suorittaa tietyn tehtävän ja arvioida, onko käyttäjän helppo ymmärtää ja oppia käyttämään käyttöliittymää. Menetelmä on käytännöllinen kriittisten kohtien sekä pienten sovellusten läpikäymisessä. (Ranne 2005, 125) Kognitiiviseen läpikäyntiin sisältyvä kokeilevan oppimisen näkökulma on tärkeä, koska aiempaa tietokoneen käyttökokemusta omaavat käyttäjät opettelevat sovellusten käyttöä ennemmin itse tutkien käsikirjojen lukemisen ja muodollisen koulutuksen sijasta (Rieman 1996).

Kognitiivinen läpikäynti on jalostunut ajan myötä useammaksi versioksi. Ensimmäisen version ongelmana oli se, että vain kognitiivisen psykologian asiantuntijat pystyivät hyödyntämään menetelmää, vaikka tavoitteena olikin ollut yleiskäyttöisempi työkalu käytettävyyden arviointiin. Tilannetta yritettiin paikata yksityiskohtaisilla ja runsailla lomakkeistoilla, jotka kuitenkin tekivät menetelmästä muodollisen ja raskaan. Nykyversio kognitiivisesta läpikäynnistä perustuu suppeampaan lomakkeistoon ja vapaampaan keskusteluun. Siinä keskitytään käyttäjän motivaatioon valita tietty toiminto ja suorittaa toiminnot käyttöprosessin kannalta oikeassa järjestyksessä. (Ranne 2005, 126-127)

Ranteen (2005, 127) mukaan kognitiivinen läpikäynti toimii samankaltaisesti Normanin (1991, 83) ihmisen toiminnan seitsenvaiheisen mallin kanssa. Normanin mallista on johdettavissa suunnittelukysymykset, joilla testausta tehdään. Käyttäjän pitäisi kyetä esimerkiksi ymmärtämään laitteen toiminta ja toteamaan mahdolliset toiminnot, toteamaan käyttöliittymän tila, määrittämään aikomuksen ja konkreettisen toiminnan välisen kytkennät sekä lopulta suorittamaan toimenpide (Norman 1991, 83).

Kognitiivinen läpikäynti selvittää siis yksinkertaistaen sitä, ovatko kohteen käyttömahdollisuudet riittävän selkeät ja onko käyttöliittymä helposti opittavissa. Esimerkiksi lattian käyttömahdollisuuksia ovat seisominen ja käveleminen. Kaikki ympäristöstä havaitut käyttömahdollisuudet eivät kuitenkaan ole välttämättä käyttökelpoisia tai ainakaan hyödyllisiä ja voivat olla myös piilossa. Näyttöä voi koskettaa, mutta ellei kyseessä ole nimenomaan kosketusnäyttö, ei siitä ole mitään hyötyä. (Sinkkonen ym. 2002, 95) Luonnollisesti myös verkkosovellusten sisältämien käyttömahdollisuuksien tulisi olla helposti havaittavissa ja käytettävissä.

Kognitiivinen läpikäynti perustuu neljään arviointikysymykseen (TAULUKKO 3), joita läpikäynnin kuluessa toistetaan kaikissa työvaiheissa ja jotka muodostavat näin iteratiivisen arviointisyklin (Wharton ym. 1994, 112).

**TAULUKKO 3. Kognitiivisen läpikäynnin neljä arviointikysymystä (Wharton ym. 1994)**

1. Tavoitteleeko käyttäjä toimillaan oikeaa toimintoa? (Jakaako käyttäjä päätavoitteen samalla tavalla alitehtäviin kuin mitä suunnittelija on olettanut?)
2. Huomaako käyttäjä onko oikea toiminto saatavilla? (Voiko käyttäjä havaita tarvittavan toiminnon tai voiko sen tietää mahdollisesti kokemuksen perusteella?)
3. Yhdistyvätkö käyttäjällä oikeat toiminnot käyttötavoitteisiin? (Ovatko valikot, kuvakkeet ja termistö ymmärrettäviä?)
4. Kun oikea toiminto on suoritettu, voiko palautteesta päätellä tehtävän oikean etenemisen? (Onko palaute riittävä?)

Ennen varsinaista testausta suoritetaan läpikäynnin esiselvitys. Esiselvityksessä hahmotellaan tiedot käyttäjäryhmistä ja laaditaan läpikäynnissä käytettävät tehtävät, joiden etukäteen määritelty suoritusjärjestys pidetään koko istunnon ajan voimassa. Lisäksi taustalle luodaan skenaarioita, jotka muodostuvat käyttäjien tavoitteista ja tehtäväkokonaisuuksista. (Ranne 2005, 129)

Kognitiivisen läpikäynnin aikana tarkastellaan toimintasarjoja kohteena olevasta käyttöliittymästä ja yritetään kertoa uskottava tarina. Tarinassa perustellaan käyttäjän toimia: miksi valita juuri tietty toiminto ja mitkä toiminnot olisivat edes mahdollisia. Osatehtävien mahdollinen onnistuminen tai epäonnistuminen perustellaan käyttäjäryhmän ominaisuuksiin peilaten. (Riihiaho 2000, 42) Epäonnistumisen jälkeen testattava käyttöliittymä palautetaan tilaan, josta testi-suoritusta on mahdollista taas jatkaa (Wharton 1994, 123). Istunnossa kirjataan kriittinen informaatio käyttäjistä ominaisuuksiineen, käyttäjän tarvitsemista taustatiedoista, tehtävän aikana vaadittavasta oppimisesta sekä esiin tulleet

ideat ja parannusehdotukset. Asettumalla käyttäjän asemaan voidaan löytää selitykset löydetyille ongelmille. (Riihiaho 2000, 42-43) Kognitiivisen läpikäynnin avulla löydetään esimerkiksi poikkeavat käsittemallit käyttäjien ja suunnittelijoiden välillä, epäonnistuneet sanavalinnat valikoissa ja painikkeissa sekä puutteellinen palaute (Wharton ym. 1994, 124).

#### 4.3.2 Automaattinen arviointi

Käytettävyyden *automaattinen arviointi* tarkoittaa automatisoitujen työkalujen hyödyntämistä käytettävyyden arviointiin ja testaukseen liittyvän aineiston keräämisessä ja analysoinnissa. Keskeinen automaation muoto on myös korjausehdotusten automaattinen esittäminen. (Heimonen 2005, 169) Automaattisessa arvioinnissa kerättävä aineisto pitää sisällään esimerkiksi tehtävien suoritusai-koja, käyttäjien ja suunnittelijoiden tekemiä virheitä sekä subjektiivisia arvioin-teja kohteesta. Kerätyn aineiston analyysillä puolestaan löydetään käytettä-vyysongelmat, joita mahdolliset parannusehdotukset koskevat. (Ivory & Hearst 2001)

Automatisoidut tarkastusmenetelmät esimerkiksi havaitsevat ja raportoivat an-nettuja käytettävyyssääntöjä rikkovia osa-alueita käyttöliittymästä ja parhaim-millaan pystyvät myös esittämään valmiita korjausehdotuksia (Heimonen 2005, 175). Automaattiseen arviointiin käytettyjen tietokoneohjelmistojen automaati-on aste vaihtelee, ja ne voivat toimia joko perinteisen, aitoja käyttötilanteita muistuttavia tehtäviä hyödyntävän, käytettävyydestauksen tukena tai itsenäi-sesti. (Heimonen 2005, 169) Lähimpänä asiantuntija-arviointia ovat ohjelmat, joiden käyttöön vaaditaan vain asiantuntijakäyttäjä tai -käyttäjiä eikä lainkaan erillisiä testihenkilöitä.

Käytettävyyden automaattisessa arvioinnissa käyttäjän toimet tallennetaan usein lokitiedostoihin erillisen tallentavan ohjelmiston avulla, joka voi sijaita yksittäisellä koneella tai välityspalvelimella (Heimonen 2005, 175; Hong, Heer, Waterson & Landay 2001). Jotkin ohjelmat keräävät tietoja kohteesta myös ilman varsinaisia testikäyttäjiä. (Ivory & Hearst 2001) Ohjeistukset, joita vasten aineistoa analysoidaan, on myös operationalisoitava, jolloin niistä tulee konkreettisia ja mitattavia. Operationalisointi tehdään esimerkiksi laskien määrällisiä suureita, kuten käyttöliittymäelementtien kokotietoja, ruudun käyttöä ja elementtien sijoittelua. (Heimonen 2005, 175)

Sampson (2006) torjuu laadullisten tekijöiden numeerisen mittaamisen liki mahdottomana tehtävänä, jonka tulokset ovat useimmiten vain harhaan johtavia. Hän myös muistuttaa, että se mitä kyetään mittaamaan, ei välttämättä ole loppukäyttäjälle oikeasti merkityksellistä. Nielsenin (2004) mukaan epätäydellisenkin laadullinen käytettävyydetutkimus tuottaa usein päteviä tuloksia, kun taas kvantitatiivisessa tutkimuksessa validin tuloksen tuottaminen vaatii erityistä huolellisuutta ja huippuosaamista.

Ivory ja Hearst (2001) ovat luokitelleet automaattiset arviointityökalut neljän muuttujan mukaan, joita ovat työkalun luokka, työkalun tyyppi, automaation tyyppi ja vaadittava työpanos. Työkalun luokka kuvaa käytetyn arviointityypin korkealla tasolla ja se jakaantuu testaustyökaluihin, tarkistustyökaluihin, kyselytyökaluihin, mallinnustyökaluihin ja simulointityökaluihin. Työkalun tyyppi kuvaa, kuinka käytettävyyden arviointi suoritetaan kunkin luokan sisältämällä työkaluilla, esimerkiksi tarkistuslistaa tai kognitiivista läpikäyntiä käyttäen. Automaation tyyppi kertoo, mikä käytettävyyden arvioinnin osa-alue on automaattista, ja vaadittava työpanos muodostuu työkalun käyttöön liittyvien toimenpiteiden vaatimasta työmäärästä.

Testaus-, tarkastus- ja kyselytyökaluilla voidaan kerätä sekä laadullista että määrällistä aineistoa. Näiden työkaluluokkien sisällä käytettyjä varsinaisia työkaluja ovat usein kognitiivinen läpikäynti, lokitiedostojen analyysi, heuristinen arviointi, kyselylomakkeet tai käytettävyydestaus. Mallinnus- ja simulointityökalut ovat puolestaan lähempänä järjestelmien suorituskykyanalyysiä. (Heimonen 2005, 172-173)

Suurimmat ongelmat automaattisessa käytettävyyden arvioinnissa liittyvät työkalujen tarkkuuden puutteisiin verrattaessa niitä perinteisempiin menetelmiin. (Heimonen 2005, 169) Työkalut eivät esimerkiksi osaa analysoida laadullisia suunnitteluohjeita. Työkalut eivät myöskään pääsääntöisesti tue WWW:n edistyneitä ominaisuuksia, kuten Flash-laajenteita tai Java-sovelmia. (Heimonen 2005, 176) Nielsen (1999) on ollut jopa sitä mieltä, että WWW-ympäristössä automaattisin menetelmin voidaan tarkistaa ainoastaan vasteaikoja, HTML:n standardinmukaisuus, linkitysten toimivuus ja sivujen saavutettavuus. Toisaalta taas HTML-koodia joudutaan tekemään joskus tahallaan epävalidilla tavalla, jotta selaimissa olevia ohjelmointivirheitä saadaan kierrettyä.

Esimerkkejä yksittäisistä automaattisen arvioinnin verkkosovelluksille sopivista ratkaisuista ovat esimerkiksi Cognitive Walkthrough for the Web (Blackmon, Polson, Kitajima & Lewis 2002), eli WWW-sivujen kognitiivinen läpikäynti, ja InfoScout™ Bloodhound Simulator (Chi, Pirolli & Pitkow 2000). Molemmat ovat erityisesti navigointiin ja tiedonhakuun keskittyviä työkaluja. Automaattiset menetelmät sopivatkin paremmin suoraviivaisten tiedonhakutehtävien kuin monimutkaisten verkkosovellusten testaamiseen. Verkkosovellusten arviointi automaattisesti voi olla hankalaa, sillä sovelluksessa on usein suuret määrät erilaisia toisiinsa vaikuttavia valintoja, toimintoja ja tiloja, joiden hyödyllisyys tai oikeellisuus ovat kulloinkin suoritettavasta tehtävästä riippuvia.

Automaattiseen arviointiin laadittu WWW-sivujen kognitiivinen läpikäynti vertailee tavoitekuvauksien ja yksittäisten verkkosivujen sisältöjen vastaavuuksia. WWW-sivujen kognitiivisessa läpikäynnissä kuvitellun käyttäjän tavoitteet kuvataan tarkasti, kuitenkin enintään parinsadan sanan pituisella tekstillä ja käyttäjälle ominaista termistöä käyttäen. Alkuperäisen kognitiivisen läpikäynnin kysymyssarjan lisäksi esillä on avainkysymys: yhdistääkö käyttäjä sivulla olevan oikean alakohdan tavoitteisiinsa eli kohtaavatko käyttäjän tavoite ja oikea painike tai valikko toisensa. Tavoitteena on paljastaa umpikujaan tai harhaan johtavat otsikot ja termit. (Blackmon ym. 2002) Työvälineen uusin versio on automaation asteeltaan pidemmälle viety, tarkempi ja sisältää myös ongelmien vakavuusluokittelun (Blackmon, Kitajima & Polson 2005).

InfoScent™ Bloodhound Simulatorissa tuotetaan automaattisesti käytettävyyssraportti asiantuntijoiden määrittelemien informaatiotarpeiden pohjalta. Työkälulle annetaan aloituspiste verkossa sekä joukko avainsanoja ja kohdesivuja, joilta tietoa aiheesta löytyy. Tämän jälkeen työkalu simuloi käyttäjän toimintaa ja palauttaa käytettävyyssarvion, joka kertoo esimerkiksi, onnistuiko tiedonhaku sivustolla ja mihin käyttäjä todennäköisesti päätyy. (Chi ym. 2003)

#### 4.3.3 Heuristinen käytettävyyssarviointi

Käytettävyyden *heuristinen arviointi* perustuu heuristiikkoihin, jotka ovat käytettävyyssperiaatteita, sääntöjä ja ohjeistuslistoja. Heuristiikat ovat välineitä tuotteen tai käyttöliittymän suunnitteluun ja toimivat hyvin asiantuntija-arviointien ohjeistuksina. (Korvenranta 2005, 112) Heuristista arviointia mainostetaan yleisesti nopeana ja vähän resursseja vaativana menetelmänä (Korvenranta 2005, 115). Heuristiset menetelmät ovat saaneet osakseen myös kritiikkiä, ja onkin syytä muistaa, että heuristiikat tarjoavat ainoastaan yleiset lähtökohdat hyvälle



käyttöliittymälle. Heuristinen käytettävyystudkimus voi olla toisinaan myös testihenkilöiden avulla suoritettavaa käyttäjätestausta edeltävä kartoittava vaihe, jonka ansiosta käyttäjätestaukseen pystytään rajaamaan mukaan vain sovelluksen merkitykselliset ja mielenkiintoiset kohdat. Vastaavasti heuristisella testauksella löydettyihin ongelmiin laadittuja erilaisia ratkaisuita voidaan asettaa paremmuusjärjestykseen käyttäjätestein. (Brooks 1994, 256-257)

Lauesenin (2005, 443) mukaan heuristisella evaluaatiolla löydetty ongelmat ovat potentiaalisia ongelmia, sillä arvioijan lausunto perustuu aina olettamuksille arvioidun kohteen ongelmallisuudesta. Heuristista listaa käyttäen arvioidaan, voisiko jokin sovelluksen ominaisuus loukata jotakin listalta löytyvää heuristiikkaa. Muita käytettävyysongelmien kriteereitä ovat myös erilaiset suunnitteluohjeet ja standardit. (Perälä 2005, 301) Asiantuntija-arvioiden perustana olevia ohjelistauksia voidaan käyttää myös tapauskohtaisesti muokattuina kulloistenkin tarpeiden mukaan.

Heuristisessa käytettävyyssarvioinnissa voidaan hyödyntää myös yrityskohtaisia *tyyliohjeistuksia* (engl. *styleguide*), jotka kattavat esimerkiksi tietyn sovelluksen tai käyttöliittymän suunnitteluohjeistuksen. Tyyliohjeistuksia voidaan käyttää tällöin eräänlaisina heuristiikkalistoina. Tyyliohjeiden avulla pyritään varmistamaan, että sovellukset toimivat samalla tavalla kuin aiemmatkin samaan tuoteperheeseen kuuluvat tuotteet. Tyyliohjeilla varmistetaan myös tuotteen yhtenäinen termistö ja graafinen ulkoasu. Esimerkiksi tietotekniikkajätit Apple ja Microsoft ovat tuottaneet käyttöjärjestelmilleen omat tyyliohjeensa. (Korvenranta 2005, 117)

Blattin ja Knutsonin (1994, 357-358) mukaan tyyliohjeita pidetään ohjelmistokehittäjien keskuudessa usein huonosti suunnittelu- ja kehitystyöhön sopivina, vaikeakäyttöisinä, liian yleisluontoisina ja toisaalta luovuutta rajoittavina. Tyy-

liohjeet vaativatkin soveltamista ja sovittamista kulloiseenkin ongelmaan, sillä ne eivät varsinaisten heurististen listausten tavoin voi kuvata jokaista ongelmatilannetta erikseen.

Käytettävyyssalan asiantuntijat ovat laatineet runsaasti kilpailevia heuristiikkalistoja yleiseen käytettävyyden arviointiin. Keinonen (1998) on vertaillut eri heuristiikkalistoja ja käytettävyyden ohjeistuslistoja keskenään ja yleistänyt niistä seitsemän keskeistä ja myös pitkälle yhteistä käytettävyyden osatekijää (TAULUKKO 4).

TAULUKKO 4. Keinosen (1998) yleistämät seitsemän keskeistä käytettävyyden osatekijää.

1. Johdonmukaisuus - Eri tapaukset ja tilanteet noudattavat yhdenmukaisia periaatteita.
2. Hallittavuus - Käyttäjä ohjaa suoraan toimintaa sen sijaan, että antaisi vain ohjeita siitä, miten kohteen pitäisi toimia.
3. Sopiva esitystapa - Käyttäjä saa kaikissa tilanteissa selkeän tiedon siitä, mitä toimintaa on tapahtumassa.
4. Virheiden sieto - Asialliset virheilmoitukset ja käskyjen peruuttamisen mahdollisuus.
5. Muistettavien asioiden määrä - Järjestelmä tarjoaa tarvittavat vaihtoehdot, joista käyttäjä saa valita haluamansa.
6. Tehtävään sopivuus - Järjestelmä esittää käyttäjälle vain ne asiat, joita tämä tarvitsee.
7. Opastus - Käyttäjä saa opastusta näyttöruudulta tai ohjekirjoista.

Nielsen (1992) on tutkinut heuristisen menetelmän käyttöä kokemuspohjaltaan ja osaamiseltaan erilaisilla arvioijilla ja vertaillut näiden arvioijien suorituksia keskenään. Nielsen jakoi arvioijat noviisiarvioijiin, joilla ei ollut koulutusta tai kokemusta käytettävyydestä, käytettävyyssalan asiantuntijoihin, joilla oli käytettävyysosaamista paljon, muttei kuitenkaan tuote- tai toimialakohtaista osaamista, sekä kahden alan asiantuntijoihin, joilla oli käytettävyyssalan osaamista ja käyttökokemusta arvioitavasta tuotteesta. Noviisit löysivät 22 % ja käytettävyyssalan asiantuntijat 41 % käytettävyysongelmista. Kahden alan asiantuntijat pääsivät 60 % suoritustasoon. Nielsen arvioi tilastollisia menetelmiä käyttäen, että viiden noviisiarvioijan ryhmä löytää puolet käytettävyysongelmista, kun puolestaan käytettävyyden asiantuntijat löytävät vastaavalla kokoonpanolla 80 % ongelmista. Käyttämällä arvioinnissa viittä kaksoisasiantuntijaa, käytettävyysongelmista jäisi löytymättä ainoastaan 2 %.

Heuristinen käytettävyyssarvio voidaan suorittaa yleisen läpikäymisen sijaan myös tehtäväpohjaisesti. Tällöin arvioija koettaa suorittaa tuotteella ennalta asetettuja tehtäviä tai arvioija itse keksii suoritettavat tehtäväkokonaisuudet. Tehtäväpohjainen heuristinen arviointi lähestyy ominaisuuksiltaan testihenkilöiden avulla suoritettavaa käytettävyydestä. Keskeisin ero heuristiikkalistaan tukeutumisen lisäksi on se, että arvioija tallentaa itse kohtaamansa ongelmat. Oletuksena tällaisessa lähestymistavassa on, että järjestelmän kehitys on edennyt jo pitkälle ja se sisältää tarvittavan toiminnallisuuden. (Lauesen 2005, 444)

Nielsenin (1994) kymmenen heuristiikan lista (TAULUKKO 5) on yleisin heuristiikkalista käytettävyyden arviointiin. Vaikka alkuperäisen, vain yhdeksän kohtaa sisältäneen, listan kehittivät alun perin Nielsen ja Molich yhdessä (Nielsen & Molich 1990), on kyseinen heuristiikkalista opittu tuntemaan yleisesti Nielsenin nimellä hänen kehitettyään sitä eteenpäin. (Korvenranta 2005, 113)

TAULUKKO 5. Nielsenin Kymmenen heuristiikan lista (Nielsen 1994, 30)

1. Palvelun tilan näkyvyys ( <i>engl. Visibility of the system status</i> )
2. Palvelun ja tosielämän vastaavuus ( <i>engl. Match between system and the real world</i> )
3. Käyttäjän kontrolli ja vapaus ( <i>engl. User control and freedom</i> )
4. Yhdenmukaisuus ja standardit ( <i>engl. Consistency and standards</i> )
5. Virheiden estäminen ( <i>engl. Error prevention</i> )
6. Tunnistaminen mieluummin kuin muistaminen ( <i>engl. Recognition rather than recall</i> )
7. Käytön joustavuus ja tehokkuus ( <i>engl. Flexibility and efficiency of use</i> )
8. Esteettinen ja minimalistinen suunnittelu ( <i>engl. Aesthetic and minimalist design</i> )
9. Virhetilanteiden tunnistaminen, diagnosointi ja korjaaminen ( <i>engl. Helping users recognize, diagnose and recover from errors</i> )
10. Opastus ja ohjeistus ( <i>engl. Help and documentation</i> )

Palvelun tilan pitäisi olla käyttäjälle näkyvä ja käyttäjälle tulisi informoida mahdollisimman tarkoin siitä, mitä se kullakin hetkellä tekee. Esimerkiksi pitkäkestoisissa hauissa tämä on erityisen merkittävää. Samoin verkkoympäristössä latausajat ja sovelluspalvelimella tapahtuva tiedonkäsittely voivat aiheuttaa katkoksia käyttöön, jolloin käyttäjälle ei saisi tulla tarpeetonta mielikuvaa jumiutuneesta järjestelmästä.

Käyttäjällä tulisi olla kontrolli tai ainakin vahva kontrollin tunne ja vapaus kokeilla erilaisia toimintoja. Käyttäjät valitsevat usein toimintoja vahingossa tai kokeilemalla ja käyttäjillä tulisikin aina olla mahdollisuus perua suorittamansa

toiminnot tarvittaessa. Tähän kytkeytyy myös vaatimus siitä, että järjestelmä ja sen toiminnot tulisi suunnitella mieluiten niin, että käyttäjän virheet pystyttäisiin minimoimaan ja mahdollisimman hyvin ennaltaehkäisemään. Mikäli virheitä kuitenkin tapahtuu, virheilmoitusten tulisi sisältää käyttäjille selkeää kieltä ja kertoa, kuinka tapahtuneen virheen jälkeen voidaan edetä. Käyttäjän tekemä virhe voi olla epähuomiossa tapahtunut lipsahdus tai vakavampi virhe, jossa käyttäjä toimii tietoisesti väärin. Kun järjestelmän tila on näkyvä, voi käyttäjä helpommin havaita toimiensa seuraukset ja myös itse arvioida, onko toiminta ollut virheellistä.

Palvelussa käytetyn termistön, kielen, symbolien ja käsitteiden tulisi olla käyttäjälle ymmärrettävää ja selkeää. Tätä tavoitetta tuetaan palvelun ja tosielämän vastaavuudella, jolloin tosielämästä tuttu tietämys voidaan valjastaa käyttöön myös virtuaalisessa ympäristössä. Järjestelmän on seurattava alan, sovellusalueen ja alustan yleisiä standardeja. Käyttäjä olisi aina uuden sovelluksen tai sovelluksen osa-alueen myötä myös suuren opettelu- ja omaksumishaasteen edessä, jos järjestelmä ei käyttäisi johdonmukaisesti samoista asioista samoja termejä ja mikäli samankaltaiset toiminnot eri näytöillä eivät vastaisi toisiaan.

Muistaminen on käyttäjän henkiselle suorituskyvyllä raskaampi operaatio kuin tuttujen asioiden tunnistaminen. Toimintojen ja järjestelmän objektien tulisikin olla aina näkyvissä siten, ettei käyttäjän tarvitsisi muistaa asioita esimerkiksi käyttöliittymänäkymästä toiseen siirryttäessä. Näin toimimalla vähennetään pitkäkestoisen muistin kuormitusta. Käytön kuormittavuutta voidaan vähentää edelleen siten, että dialogit ja näytöt eivät sisällä käyttäjälle tarpeetonta informaatiota. Jos näytöllä on paljon tehtävän suorittamiseen liittymättömiä elementtejä, on oleellisen löytäminen kaiken seasta hitaampaa ja vaikeampaa.

Järjestelmän tulisi tukea vaihtoehtoisia työtapoja käytön tehostamiseksi. Tärkeimpien ja useimmiten suoritettavien toimintojen tulisi olla muokattavissa käyttäjän tarpeiden mukaisiksi. Esimerkiksi noviisi- ja eksperttikäyttäjälle voidaan tarjota eri tavalla toimivia käyttöliittymiä, joissa järjestelmän läpikotaisin tunteva tehokäyttäjä voisi suorittaa toimintoja hyvin suoraviivaisesti ja tehokkaasti ja vastaavasti noviisikäyttäjälle tarjottaisiin moniportaista ohjeistettua etenemistapaa. Yleisesti ottaen avusteiden tulisi olla helposti käytettävissä mistä tahansa järjestelmän tilasta. Avusteiden tulisi olla yksinkertaisia ja käyttäjälle mahdollisimman informatiivisia.

Toinen erittäin tunnettu lista, josta voi löytää hyvin paljon yhteistä Nielsenin heuristiikkalistan kanssa, on Shneidermanin (1992) Kahdeksan kultaista sääntöä dialogin suunnitteluun (TAULUKKO 6) (Korvenranta 2005, 116-117). Nyttämmin vuorovaikutteisia dialogisysteemeitä tosin suositellaan kutsuttavan yksinkertaisemmin interaktiivisiksi systeemeiksi, sillä vuorovaikutteiset esineet ja asiat toimivat nykyisin yhä useammin perinteisistä valikkodialogeista poikkeavilla tavoilla. Näin olleen voidaan puhua ennemmin interaktiosuunnittelusta dialogien suunnittelun sijaan. (Geis, Dzida & Redtenbacher 2004, 59)

TAULUKKO 6. Shneidermanin Kahdeksan kultaista sääntöä dialogin suunnitteluun (Shneiderman 1992, 72-73)

1. Noudata yhteneväisyyttä toimintatavoissa ( <i>engl. Strive for consistency</i> )
2. Anna mahdollisuus oikopolkujen käyttöön ( <i>engl. Enable frequent users to use shortcuts</i> )
3. Tarjoa selkeää palautetta ( <i>engl. Offer informative feedback</i> )
4. Suunnittele dialogit niin, että ne johtavat lopputulokseen ( <i>engl. Design dialog to yield closure</i> )
5. Tarjoa helppo virheiden käsittely ( <i>engl. Offer simple error handling</i> )
6. Anna mahdollisuus toimintojen peruutukseen ( <i>engl. Permit easy reversal of actions</i> )
7. Anna käyttäjälle kontrolli (käyttökokemus) ( <i>engl. Support internal locus of control</i> )
8. Vähennä lyhytkestoisen muistin kuormitusta ( <i>engl. Reduce short term memory load</i> )

Hyvin monet käytettävyyssalan asiantuntijat ovat kehittäneet omia heuristiikkalistojaan, joista varsin monet ovat vain muokanneet Nielsenin listaa paremmin tilannekohtaisiin tarpeisiinsa sopivammaksi (Korvenranta 2005, 116).

Koska yksi arvioija ei kykene löytämään heuristiikkalistojen avulla kaikkia käytettävyysoongelmia, suosituksena olisi tehdä arviointi useamman henkilön voimin. Toisaalta suurikaan osallistujajoukko ei takaa täydellistä tulosta. Mikäli heuristinen arviointi tehdään ryhmässä, käyvät arvioijat läpi kohteena olevan käyttöliittymän tai sovelluksen kuitenkin ensin yksinään. Näin menettelemällä arvioijat eivät saa vaikutteita, ennakkotietoja tai ennakkoasenteita toisiltaan. (Nielsen 1994, 26)

Kunakin arvioijan kannattaa käydä kohde läpi ainakin kahteen kertaan, edeten yleiskatsauksesta yksityiskohtien perinpohjaisempaan tarkasteluun. Arvioija voi itsenäisesti päättää, miten hän arvioi tuotetta ja kuinka monta ja minkä tasoista arviointikierrosta suorittaa – heuristiikkalista on kuitenkin koko ajan arvioinnin tukena. Arvioinnissa voidaan hyödyntää tarvittaessa myös muita ohjeistuksia Kymmenen heuristiikan listan lisäksi, esimerkiksi Shneidermanin ohjeistusta interaktiosuunnitteluun tai tuotekohtaisia ohjeistuksia. (Nielsen 1994, 28-29) Mikäli arvioijia on monta, kokoonnutaan yksilösuoritusten jälkeen tekemään synteesi ryhmätyönä. Kun arvioinnissa on saavutettu yhtenäinen lista käytettävyyssongelmista, voidaan tämän jälkeen luokitella löydösten vakavuus (TAULUKKO 7). (Nielsen 1994, 48)

TAULUKKO 7. Nielsenin viisiportainen vakavuusluokitus havaituille käytettävyyssongelmille (Nielsen 1994, 49)

0. Kyseessä ei ole käytettävyyssongelma. ( <i>engl. I don't agree that this is a usability problem at all</i> )
1. Kosmeettinen käytettävyyssongelma, joka korjataan ajan salliessa. ( <i>engl. Cosmetic problem only – need not to be fixed unless extra time is available on project</i> )
2. Vähäinen käytettävyyssongelma, jonka korjauksella on alhainen prioriteetti. ( <i>engl. Minor usability problem – fixing this should be given low priority</i> )
3. Suuri käytettävyyssongelma, joka haittaa selvästi tuotteen käyttöä. Ongelman korjaamisella on korkea prioriteetti. ( <i>engl. Major usability problem – important to fix, so should be given high priority</i> )
4. Katastrofaalinen käytettävyyssongelma. Tuotetta ei voida julkaista ennen korjausta. ( <i>engl. Usability catastrophe – imperative to fix this before product can be released</i> )



Arvioijien tekemä vakavuusluokittelu on hyvin subjektiivista toimintaa ja näin ollen yhden arvioijan tekemään luokitukseen ei tulisi luottaa sokeasti (Nielsen 1993, 103). Suosituksena onkin, että vähintään kolme arvioijaa esittää arvionsa ja näistä johdetaan yhteinen arvo joko neuvotellen tai keskiarvoon tukeutuen. (Nielsen 1993, 156-157) Käytettävyysongelman vakavuuteen vaikuttavat ongelman yleisyys, vaikutus ja pysyvyys. Asiantuntijan tehtävänä on siis havainnoida ja arvioida, esiintyykö ongelma usein, onko ongelma vaikeasti ohitettava ja häiritseekö sama ongelma joka kerta siihen törmätessä. (Nielsen 2005) Ongelmassa voivat yhdistyä myös kaikki edelliset seikat. Mitä enemmän päällekkäisyyttä ongelma-alueiden kesken löytyy, sitä vakavammasta ongelmasta on kysymys. Toisaalta vakavankin tuntuisen ongelman luokitusta voidaan tarvittaessa harkita alennettavan, jos se esiintyy äärimmäisen harvoin (Nielsen 1993, 104). Vakavuusluokittelu on samalla myös prioriteettalista korjattavia kohtia koskien (Nielsen 2005).

Heuristisen arvioinnin tuloksena on raportti, joka sisältää listauksen käytettävyysongelmissa. Jokaisesta käytettävyysongelmasta tulisi olla viittaus heuristiikkaan tai heuristiikkoihin, joita kyseinen ongelma rikkoo ja lisäksi ongelman ratkaisemiseksi tulisi tuottaa korjausehdotus mahdollisuuksien mukaan. Heuristisen arvioinnin avulla ei voida systemaattisesti löytää ratkaisuehdotuksia ongelmakohtiin, mutta ratkaisu on usein varsin ilmeinen ja ratkaisuehdotus on helppo lisätä raporttiin. (Nielsen 1993, 159)

#### 4.4 Esimerkki heuristisen käytettävyysohjeiston käytöstä

Tässä alaluvussa havainnollistetaan, kuinka Nielsenin (1994, 30) Kymmenen heuristiikan listaa käyttäen voidaan arvioida verkkosovellusta ja millainen on tästä syntyvä raportti. Tätä työtä varten laadittu arvio on tutkielman kirjoittajan tekemä subjektiivinen näkemys, eikä siihen tulisi luottaa ehdottomana totuute-

na (Nielsen 1993, 103). Toisaalta heuristisessa arvioinnissa henkilökohtainen asiantuntijuus on keskeistä. Arvioitavaksi kohteeksi on valittu pieni Suunnitelmakorit-sovellus (KUVIO 6) Työterveyslaitoksen julkaisemasta Duunitalkoot-sivustosta (Työterveyslaitos 2007).

The screenshot shows a web browser window with the URL [http://www.ttl.fi/internet/duunitalkoot/tyoyhteisot\\_p3.html](http://www.ttl.fi/internet/duunitalkoot/tyoyhteisot_p3.html). The page title is "Duunitalkoot - Tehtävä: Suunnitelmakorit". The navigation menu includes "Minä ja työkaverit", "Esimiehet", "Kehittäjät", and "Johtajat". The main content area shows a breadcrumb trail: "Duunitalkoot > Minä ja työkaverit > Työyhteisötarina > Yksi työ - monta näkemystä > Tehtävä: Suunnitelmakorit". Below this, a paragraph states: "Tämä on työyhteisön tehtäväpolun viimeinen, kehittämistoimenpiteisiin keskittyvä osio C, Suunnitelmakorit." The main task card, titled "Tehtävä: Suunnitelmakorit", features a cartoon character and the following text: "Edellisessä tehtävässä keskustelitte erilaisista toimintatavoista ja näkemyksistä. Tässä osassa on esimerkkejä erilaisista käytännöistä, joilla voidaan edistää työhyvinvointia ja työn sujumista." Below the text are three reflection questions: "Miettikää tehtävää tehdessänne: • Mitä tehtävässä esiintyviä käytäntöjä työpaikallanne on? • Mitä uusia käytäntöjä tarvittaisiin, ja olisivatko ne myös toteutettavissa? • Mitä haasteita näette eri käytäntöjen kehittämisessä työn ja työyhteisön kannalta?". At the bottom of the card are buttons for "Avaa ohje" and "Tulosta ohje", and a "SEURAAVA" button. To the right, a sidebar titled "Työyhteisön tehtäväpolku" lists three steps: A. Työyhteisötarina - Työyhteisön moninaisuus, B. Yksi työ - monta näkemystä - Työyhteisön toimintatavat, and C. Suunnitelmakorit - Työyhteisön kehittämis-toimenpiteet. Below the sidebar is a form to share the page, with fields for "Lähetäjän nimi" (filled with "lähettäjä") and "Vastaanottajan sähköpostiosoite" (filled with "sähköpostiosoite"), and a "Lähetä viesti" button. The footer of the page contains "Valmis" and navigation links "Tulosta sivu" and "Yhteystiedot ja palaute".

KUVIO 6. Duunitalkoot-sivustoon upotettu Suunnitelmakorit-verkkosovellus (Työterveyslaitos 2007)

Arvioitava Suunnitelmakorit-sovellus on viimeinen kolmen pienen verkkosovelluksen sarjasta, jotka muodostavat yhtenäisen loogisesti jatkuvan kokonaisuuden. Kunkin sovelluksen käytettävyys voidaan kuitenkin arvioida itsenäisenä kokonaisuutena. Sovellusta arvioitaessa on otettava jossain määrin kantaa myös ympäröivään sivuun, johon se on upotettu. Verkkosovellusten ja verkkosivujen välinen raja on usein häilyvä ja toisaalta selkeärajaissakin tapauksissa verkkosivu ja sovellus ovat vaikutussuhteessa keskenään. Tutkittava Suunnitelmakorit-sovellus on Nielsenin (2000, 258-260) luokittelun mukaan sisältösovellus eli sovellus on kiinteästi kytköksissä verkkosivun sisältöön ja ulkoasuun. Ginigen ja Murugesanin (2001) mukaan kyseinen sovellus lukeutuu käyttäjän syöttämää tietoa keräävänä interaktiivisten ja työnkulkua edistävien sovellusten kategorioihin.

Vaikka toteutus on tekniikaltaan Flash-pohjainen, se ei toteuta täysin RIA-verkkosovelluksen kriteereitä, huolimatta sovelluksen visuaalisesta rikkaudesta ja useita verkkosovelluksia joustavammasta toiminnallisuudesta (Perfetti & Spool 2002; Noda & Helwig 2005). Multimediaelementtejä sisältävä Suunnitelmakorit-miniatyyrisovellus toimii asiakaskoneen varassa, jolloin puhutaan asiakas-palvelin -arkkitehtuurista ja lihavasta asiakkaasta (Polvinen 1999, 22).

Sovellusta on tarkasteltu tutkielman kohdassa 4.3.3 esitettyjen suuntaviivojen mukaisesti. Nielsenin heuristiikkojen juuret ovat ajalta ennen WWW:n läpimurtoa, mutta ne soveltuvat joustavuutensa ansiosta myös nykyisin moneen käyttöön. Nielsenin (2004) mukaan käytettävyys on vakaa tieteenala ja käyttäjien toiminta pysyy melko samanlaisena vuodesta toiseen.

Oheiseen taulukkoon (TAULUKKO 8) on kerätty läpikäynnissä löydetyt käytettävyysongelmat ja mahdolliset korjausehdotukset Huomautukset-sarakkeeseen. Joissain kohdissa korjausehdotus sisältyy implisiittisesti ongelmakuvaukseen,

eikä siitä ole katsottu tarpeelliseksi enää erikseen mainita. Käytettävyysoongelmasta on kerrottu sen sijainti sovelluksen sisällä ja mitä heuristiikkaa tai heuristiikkoja ongelma rikkoo sekä vakavuusaste.

**TAULUKKO 8. Yhteenveto Suunnitelmakorit-verkkosovelluksen heuristisesta arviosta.**

<b>Heuristiikat</b>				<b>Vakavuusluokittelu</b>
1. Palvelun tilan näkyvyys 2. Palvelun ja tosielämän vastaavuus 3. Käyttäjän kontrolli ja vapaus 4. Yhdenmukaisuus ja standardit 5. Virheiden estäminen 6. Tunnistaminen mieluummin kuin muistaminen 7. Käytön joustavuus ja tehokkuus 8. Esteettinen ja minimalistinen suunnittelu 9. Virhetilanteiden tunnistaminen, diagnosointi ja korjaaminen 10. Opastus ja ohjeistus				0. Kyseessä ei ole käytettävyysongelma. 1. Kosmeettinen käytettävyysongelma, joka korjataan ajan salliessa. 2. Vähäinen käytettävyysongelma, jonka korjauksella on alhainen prioriteetti. 3. Suuri käytettävyysongelma, joka haittaa selvästi tuotteen käyttöä. Ongelman korjaamisella on korkea prioriteetti. 4. Katastrofaalinen käytettävyysongelma. Tuotetta ei voida julkaista ennen korjausta.
<b>Ongelma</b>	<b>Ongelman sijainti sovelluksessa</b>	<b>Heuristiikat</b>	<b>Ongelman vakavuus</b>	<b>Huomautukset</b>
Ohjetekstissä viitataan Tietotori-osioon, vaikka sellaista ei ole missään näkyvissä ja saatavilla. Sana itsessään ei ole myöskään linkki.	1/9	2, 10	2	
Avautuvassa Ohjeikkunassa ei ole omaa sulkemismekanismia. Sulje-painike ja ikkuna ovat erilliset.	1/9	3	1	Konvention mukaisesti Ohjeikkunassa voisi olla yläkulmassa x-painike sulkemista varten.

(jatkuu)

TAULUKKO 8. (jatkuu)

Toiminnallisten i-symbolien sijoittelu on sekavaa samalla sivulla. Lisäksi i-symbolien paikat vaihtuvat eri välilehdillä.	2/9	8, 4	1	
Raahattavat objektit eivät sisällä varsinaista graafista vihjettä siitä, että niihin on mahdollista tarttua ja että niitä voi liikuttaa.	3/9	3, 10	1	
Valittuja kohteita koskeva lisäinfo on saatavilla vain sivulla 2/9. Muistia virkistääkseen on palattava sovelluksessa taaksepäin.	4/9	6	2	Myös sivulla 4/9 voisi olla käytävissä samanlainen i-symboli valittua kohtaa koskevina lisäinfoineen kuin sivulla 2/9.
Lomakekenttien välillä ei voi liikkua näppäimistön avulla (tabnäppäin)	4/9, 5/9 ja 6/9	7	2	
Sovelluksessa laaditut muistiinpanot voidaan vain tulostaa eikä hyödyntää sähköisessä muodossa, vaikka ne kirjoitetaan tietokoneella.	7/9	7	3	Muistiinpanot pitäisi pystyä tallentamaan tiedostoon, viemään leikepöydälle tai lähettämään sähköpostiin.
Hypertekstilinkeissä voisi olla alleviivaus tai tehoste osoittimen osuessa kohdalle.	9/9	4	1	
Linkitys tekstissä mainittuun Ohjaajan oppaaseen puuttuu. Opas pitää etsiä itse muualta sivustolta.	9/9	3, 7	2	

(jatkuu)

TAULUKKO 8. (jatkuu)

Jos selaimen sulkee vahingossa tai navigoi selaimen Takaisin-painikkeella toiselle sivulle, häviävät kaikki sovellukseen syötetyt tiedot.	koko sovellus	5, 9	4	Takaisin-painikkeen käyttöä ei voi estää, koska se on keskeisimmin käytettyjä toimintoja selaimessa. Koska virhettä on vaikea estää, tulisi siitä toipuminen olla nopeaa: ts. tiedot olisivat tallessa kun sovellukseen palataan virhetoiminnon jälkeen
Edellinen- ja Seuraa-painikkeet eivät erotu parhaalla mahdollisella tavalla taustasta, koska molemmat ovat vihreitä.	koko sovellus	3	1	

Korjausta vaativien kohtien vastapainoksi positiivisia huomioita saivat kokonaisuudessaan esteettinen ja muun sivuston kanssa yhtenäinen ulkoasu sekä sovellukseen upotettu ohjeistava metateksti. Sovellus on etenemiseltään melko lineaarinen ja varsinainen eksyminen on lähes mahdotonta sovelluksen sisällä. Käyttäjä saanee lisäksi riittävästi informaatiota missä päin milloinkin on, koskien sekä sovellusta että laajempaa sivustokontekstia.

#### 4.5 Yhteenveto

Käytettävyys on moniulotteinen käsite, jota on määritelty useiden tutkijoiden voimin ja useista lähtökohdista ilman konsensusta yksikäsitteisestä määritelmästä. Käytettävyyttä voidaan tarkastella suppeasti yhtenä laatuominaisuutena esimerkiksi käytön helppouden ja tehokkuuden näkökulmasta, tai ajatella sitä laajemmin tuotteen käytöstä seuraavana kokonaisvaltaisena laatukokemuksena. Tieteellisen käytettävyysnäkökulman ohi ajaa usein ajatus käyttäjäkokemuksesta. Tällöin suunnittelun kohteeksi nousee tuotteen tai palvelun välittämä kokonaiselämyksellisyys.

Tuotetta ja sen toiminnallisuutta arvioitaessa tutkitaan pelkistetyimmillään mitä tuote tekee. Tuotteen käytettävyyttä tutkittaessa katsotaan pääsääntöisesti sitä, suorittaako tuote käyttäjän tahdon mukaisesti siihen suunniteltuja toimintoja. Käyttäjäkokeemusta kartoitettaessa joudutaan sen sijaan puntaroimaan hieman hankalampia ja moniulotteisempia asioita, kuten esimerkiksi käyttäjän tunteuksia ja kokemuksia. Käyttäjäkokeemuksen keskiössä ovat yksilöllinen käyttäjä ja tuote, jotka sijaitsevat aina jossakin tietyssä ympäristössä ja edelleen sosiaalisessa ja kulttuurisessa tilassa. Tuote vaikuttaa käyttäjään ominaisuuksillaan, kuten toiminnallisuudella, muotokielellä ja estetiikalla. Käyttäjäkokeemusta ei voida suunnitella, mutta suunnittelutyötä positiivisten käyttäjäkokeemusten edesauttamiseksi voidaan tehdä.

Käytettävyyden asiantuntija-arviointia kutsutaan monella eri nimellä. Esimerkiksi heuristista arviointia ja asiantuntija-arviointia käytetään monesti synonyyminä. Asiantuntija-arviot ovat analyttisiä menetelmiä, joissa käytettävyydsarviointi on ennakoivaa. Tällöin tutkitaan, mikä sovelluksen ominaisuus tai toiminto tulee olemaan mahdollinen käytettävyydsongelma. Asiantuntija-arvioinnin menetelmiä ovat esimerkiksi kognitiivinen läpikäynti, käytettävyyden automaattinen arviointi ja heuristinen käytettävyydsarviointi. Arviointi suoritetaan tyypillisesti ilman loppukäyttäjiä yhden tai useamman asiantuntijatoimijan voimin.

Kognitiivisessa läpikäynnissä yritetään mallintaa käyttäjän ajatuksia ja toimintaa, kun käyttöliittymä kohdataan ensimmäistä kertaa. Tavoitteena on jäljitellä, kuinka käyttäjä suorittaa tietyn tehtävän ja arvioida, onko käyttäjän helppo ymmärtää ja oppia käyttämään käyttöliittymää. Kognitiivinen läpikäynti selvittää yksinkertaistaen sitä, ovatko kohteen käyttömahdollisuudet riittävän selkeät. Kognitiivisen läpikäynnin avulla löydetään esimerkiksi poikkeavat käsite-

mallit käyttäjien ja suunnittelijoiden välillä, epäonnistuneet sanavalinnat valikoissa ja painikkeissa sekä puutteellinen palaute.

Käytettävyyden automaattinen arviointi tarkoittaa automatisoitujen työkalujen hyödyntämistä käytettävyyden arviointiin ja testaukseen liittyvän aineiston keräämisessä ja analysoinnissa. Automaattiseen arviointiin käytettyjen tietokoneohjelmistojen automaation aste vaihtelee, ja ne voivat toimia joko perinteisen käytettävyydestauksen tukena tai itsenäisesti. Parhaimmillaan automaattisen asiantuntija-arvioinnin välineet pystyvät myös esittämään valmiita korjausehdotuksia arvioitavasta kohteesta. Automaattinen arviointi hyödyntää olemassa olevia asiantuntija-arvioinnin menetelmiä, mistä hyvänä esimerkkinä toimii WWW-sivujen kognitiivinen läpikäynti. Suurimmat ongelmat automaattisessa käytettävyyden arvioinnissa liittyvät työkalujen tarkkuuden puutteisiin verrattaessa niitä perinteisempiin menetelmiin. Työkalut eivät esimerkiksi osaa analysoida laadullisia suunnitteluohjeita. Automaattiset menetelmät sopivatkin paremmin esimerkiksi suoraviivaisten tiedonhakutehtävien kuin monimutkaisten verkkosovellusten testaamiseen.

Käytettävyyden heuristinen arviointi perustuu heuristiikkoihin, jotka ovat käytettävyydsperiaatteita, sääntöjä ja ohjeistuslistoja. Heuristisella evaluaatiolla löydettyt ongelmat ovat potentiaalisia ongelmia, sillä arvioijan lausunto perustuu aina olettamuksille arvioidun kohteen ongelmallisuudesta. Koska yksi arvioija ei kykene löytämään heuristiikkalistoja avulla kaikkia käytettävyyso ongelmia, suosituksena olisi tehdä arviointi useamman henkilön voimin. Käytettävyydsalan asiantuntijat ovat laatineet runsaasti kilpailevia heuristiikkalistoja yleiseen käytettävyyden arviointiin. Nielsenin kymmenen heuristiikan lista on yleisin heuristiikkalista käytettävyyden arviointiin. Toinen erittäin tunnettu lista, josta voi löytää hyvin paljon yhteistä Nielsenin heuristiikkalistan kanssa, on Shneidermanin Kahdeksan kultaista sääntöä dialogin suunnitteluun. Heuristi-



sessä arvioinnissa etsitään käytettävyysongelmia, luokitellaan ongelmia vakavuusasteen mukaan sekä mahdollisuuksien mukaan esitetään korjausehdotukset.

## 5 ASIANTUNTIJA-ARVIOINNIN SOVELTAMINEN RIA-VERKKOSOVELLUKSIIN

Käytettävyyden läpikäynti asiantuntija-arvioinnin menetelmin toimii kuin linssi, jonka kautta käyttöliittymiä tarkastellaan. Linssin keskipisteessä on tarkastelun kohde, jota käsitellään kohdealueen rajauksen ja syvyyden suhteen. Kohdealue voidaan rajata käsittämään esimerkiksi tyyliohjeiden noudattamista, yleistä käytettävyyttä tai graafista toteutusta. Kohdetta voidaan tarkastella myös suoranaisten käytettävyyssasioiden ulkopuolelta. Fokuksessa voi olla esimerkiksi suunniteltujen käyttöliittymien tekninen toteuttamiskelpoisuus. Valittu syvyys vaikuttaa siihen, onko arvioinnin kohteena käyttöliittymän tietty osa-alue vai yritetäänkö käsitellä kaikki mahdolliset käyttöliittymästä löytyvät piirteet. Käytettävyyden asiantuntija-arvioinnin tavoitteiden määrittely helpottuu puntaroidulla etukäteen tarkastelun syvyyttä ja rajausta. (Wixon ym. 1994, 80-82)

Tässä tutkielmassa tarkastelun polttopisteessä ovat RIA-verkkosovellusten ominaisuudet, jotka ovat perinteisiin verkkosovelluksiin nähden uusia tai muuttuneita ja jotka vaikuttavat käytettävyyteen. RIA-verkkosovellukset ja perinteiset verkkosovellukset omaavat pohjimmiltaan yhteisen taustan ja paljon yhteisiä piirteitä, mutta RIA-verkkosovelluksilla on uudempana ja kehittyneempänä ilmiönä joukko käytettävyyteen vaikuttavia ominaisuuksia, joita ei ole mahdollista käsitellä täysin kattavasti samoilla välineillä kuin perinteisiä verkkosovelluksia (Asleson & Schutta 2006, 16).

Ensimmäisessä alaluvussa käsitellään käyttäjäkokemusta RIA-verkkosovellusten kontekstissa ja pohditaan käyttäjäkokemuksen yhteyksiä käytettävyyteen. Toisessa alaluvussa paneudutaan RIA-verkkosovellusten käytettävyyteen vaikuttaviin piirteisiin. Kolmannessa alaluvussa esitellään tarkastuslista RIA-verkkosovellusten käytettävyyden asiantuntija-arvioinnin tueksi

yhteenvedonomaaisesti. Tarkastuslista pohjautuu tutkielmassa läpikäytyyn kirjallisuuteen. Tarkastuslistan tavoitteena on täydentää sinällään pätevien perinteisten käytettävyyden asiantuntija-arvioinnin menetelmien soveltuvuutta RIA-verkkosovellusten arviointiin. Lista voi toimia myös ohjeistuksena ja muistilistana RIA-verkkosovellusten kehittäjille.

### 5.1 RIA-verkkosovellukset ja käyttäjäkokemus

Käyttäjäkokemuksen saama suuri huomio RIA-verkkosovelluksissa juontaa juurensa ensisijaisesti uuden tekniikan mukanaan tuomista mahdollisuuksista. Positiivisia käyttäjäkokemuksia haettaessa voidaan luoda samalla myös uusia ulottuvuuksia sovelluksen käytettävyyteen, mutta toisaalta sovellus voidaan myös pilata käytettävyydeltään tuomalla siihen liikaa häiritseviä tekijöitä mukaan.

RIA-verkkosovelluksissa sekä toiminnallisuus että visuaalisuus ovat vahvassa vuorovaikutussuhteessa keskenään. Nämä tekijät vaikuttavat edelleen rinnakkain toisistaan riippuvaisiin käytettävyyteen ja käyttäjäkokemukseen. Kuten Nielsen (1993, 26) on todennut, käytettävyys ei ole käyttöliittymästä havaittava yksiulotteinen ominaisuus, vaan se koostuu useista tekijöistä. RIA-verkkosovellukset sisältävät useita tuoreita ja edellisen sukupolven verkkosovellustoteutuksista poikkeavia teknisiä ratkaisuja. Lisäksi niille asetetaan usein runsaasti ei-toiminnallisia vaatimuksia esimerkiksi käyttäjäkokemuksen suhteen.

Nielsen (2002) toteaa käytön helppouden olevan ensiarvoista, mutta muistuttaa myös sovelluksen käyttämisen ilon tärkeydestä. Käyttämisen iloa on kuitenkin vaikeaa saavuttaa, jos tuote ei ole myös helppokäyttöinen. Nielsen (2002) peräänkuuluttaakin käytettävyyshetimitä, joilla voitaisiin käsitellä tuotteen käy-

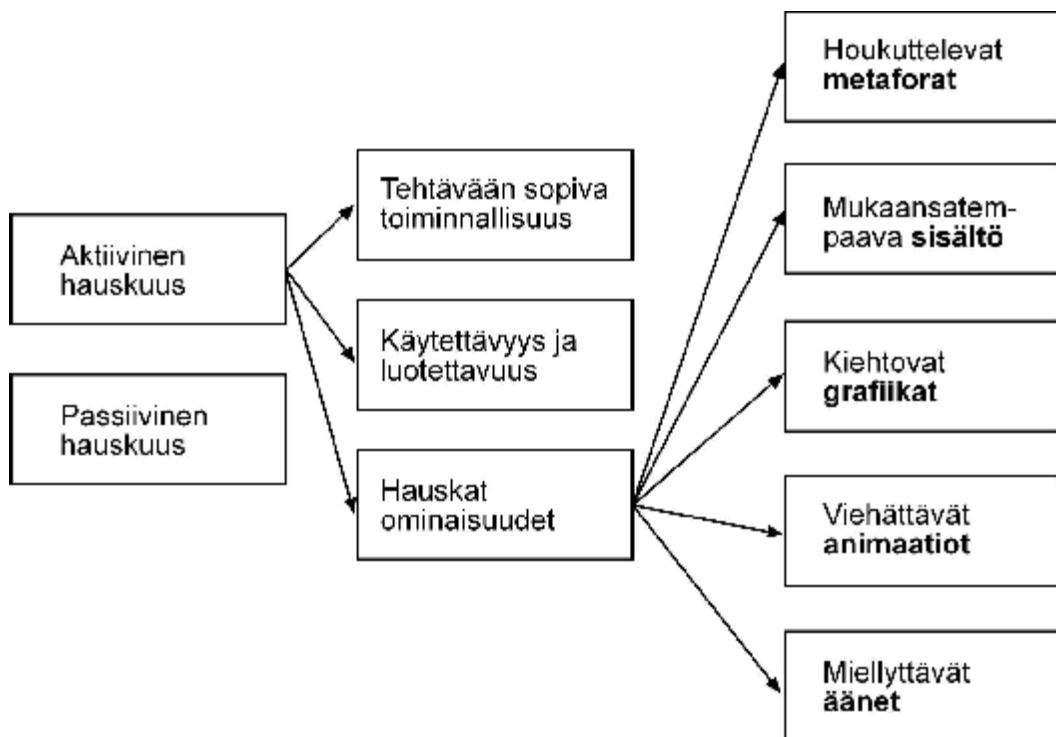
töstä saatavaa iloa ja vahvistaa tätä piirrettä. Koska käytettävyytutkimuksen tavoitteena on tuottaa tietokoneen käyttöelämyksistä myös nautittavampia, on hauskuuden lisääminen käytön kokemukseen keskeinen haaste (Pemberton 2004). RIA-verkkosovellusten toteutustekniikat tarjoavatkin aikaisempaa paremmat mahdollisuudet tällaisten ominaisuuksien toteuttamiseksi.

Olosuhteissa, joissa verkkosovelluksen käyttäminen on harkinnanvaraista ja jossa käyttäjältä vaaditaan jatkuvaluonteista interaktiota sovelluksen kanssa, helppokäyttöisyys ja käytön yksinkertaisuus eivät ole tarpeeksi kattava ominaisuusvalikoima. Ihmisten pitää haluta käyttää verkkosovellusta ja halun tulee säilyä käytön myötä. Jos verkkosovellusta on hauska käyttää ja käyttäjäkokemus on positiivinen, auttaa se osaltaan tämän tavoitteen saavuttamisessa. (Carroll 2004) Positiiviselle kokemukselle voidaan pedata hyvät olosuhteet, mutta viime kädessä käyttäjä tietyssä käytön kontekstissa ja olosuhteissa päättää suhtautumisellaan, toteutuvatko tekijöiden suunnitelmat (Hassenzahl 2004).

Shneiderman (2004) luokittelee hauskuuden kahteen kategoriaan, aktiiviseen tekemisen hauskuuteen (*engl. fun-in-doing*) ja passiiviseen tekemättömyyteen ja rentoutumisen liittyvään hauskuuteen (*engl. fun-in-not-doing*). Erityisesti aktiivinen tekemisen hauskuus voidaan liittää tietokoneohjelmien käyttämiseen. Mikäli kyse ei ole pelkästä viihdesovelluksesta, on sovelluksessa oltava tarpeelliset toiminnot, jotta sovellusta käyttämällä on mahdollista saada myös tuloksia aikaiseksi. Toiseksi sovelluksen on oltava käytettävyydeltään hyvä sekä toiminnaltaan luotettava, jotta pintaan nousevat turhaumat eivät veisi positiiviselta käyttäjäkokemukselta pohjaa.

Kun peruskäytettävyys ja perustoiminnallisuus ovat kunnossa, voidaan sovellukseen lisätä vielä ylimääräisiä ”mausteita”, joilla voi koettaa miellyttää käyttäjäkuntaa. Käyttäjän kokemaan hauskuuteen vaikuttavat houkuttelevat meta-

forat, mukaansatempaava sisältö, kiehtovat grafiikat, viehättävät animaatiot sekä miellyttävät äänet (KUVIO 7). Edellä luetellut ominaisuudet voivat ilmetä esimerkiksi äänimaisemana tai animoituina näkymävaihdoksina. (Shneiderman 2004) Tällaisten ominaisuuksien ilmeneminen on perinteisissä verkkosovelluksissa harvinaista, mutta uusissa RIA-verkkosovelluksissa ne ovat lähes oletusarvoisia.



KUVIO 7. Käyttöliittymän hauskuutta lisäävät elementit Shneidermania (2004) mukailten.

Animaatioiden ainoa arvo ei perustu pelkästään niiden viehättävyyteen. Animaatiota voidaan hyödyntää tehokkaasti seitsemään eri käyttötarkoitukseen, jotka ovat siirtymien välisen jatkuvuuden osoittaminen, siirtymän suunnan osoittaminen, ajan myötä tapahtuvan muutoksen esittäminen, näytön tehokas

hyödyntäminen, graafisten esitysten tehostaminen, kolmiulotteisten rakenteiden havainnollistaminen ja huomion kiinnittäminen. (Nielsen 2000, 145) Vastaavasti äänimaailmaan panostaminen on kannattavaa koko sovellusta koskien. Nielsenin (2000, 155) mukaan myös sovelluksen muut ominaisuudet koetaan helposti paremmiksi, kun äänet on tehty hyvin. Ääntä voi käyttää paitsi tehosteena, myös silloin, kun kaikkea ei ole mahdollista saada näkyviin. Ääni voi välittää tietoja, joita ei saada millään muulla tavalla esiin. Äänen perusteella voidaan esimerkiksi päätellä, että sovellus toimii kunnolla tai että se vaatii huomiota. (Norman 1991, 149)

RIA-verkkosovellusten tarjoamien prosessien kokemus on myös kontekstiriippuvainen siinä missä tosielämän palveluprosesseja arvioitaessa (Grönroos 2001, 103). Aivan kuten ravintolan nuhjuisuus vaikuttaa siihen, kuinka palvelu koetaan, vaikuttavat myös verkossa vastaavat mekanismit. Tökerösti toteutettu ja ulkoasultaan ”nuhruinen” sivusto tai sovellus herättää käyttäjässä kielteisiä mielteitä. Näin ollen RIA-verkkosovellusten ja niiden käytettävyyden suunnittelussa ja arvioinnissa ei voida keskittyä pelkästään tekniseen ja toiminnalliseen tarkasteluun. Tämä pätee myös perinteisiin verkkosovelluksiin, mutta RIA-verkkosovellusten perinteisiä verkkosovelluksia visuaalisemman luonteen takia esteettisen ulottuvuuden painoarvo on aiempaa suurempi. Miellyttävät ja esteettiset sovellukset, jotka luovat käyttäjissä positiivisia tuntemuksia, vähentävät mahdollisten vähäisten käytettävyysongelmien merkitystä (Norman 2002).

Sovelluskehittäjien tulisi kilpailla markkinoilla puntaroida, pitäisikö RIA-verkkosovellusten sisältää uusien toteutustekniikoiden mahdollistamia käytön hauskuutta ja miellyttävyyttä lisääviä ominaisuuksia. Lisäksi tulisi omaan asiantuntijuuteensa ja kokemuspohjaansa nojautuen kyetä arvioimaan, sopivatko ne muuhun sisältöön ja käyttöliittymään luontevasti, jotta ne eivät hyvistä tar-

koitusperistään huolimatta heikentäisi käytettävyyttä. Tämän ohella tulisi pitää huoli, että RIA-verkkosovelluksen ulkoasu on miellyttävä ja esteettinen.

## 5.2 RIA-verkkosovellukset ja käytettävyys

RIA-verkkosovellusten käytettävyyttä tarkasteltaessa voidaan huomioida paitsi suoraan sovelluksen toimintaan liittyvät seikat, myös sovellusta ympäröivän sivuston ominaisuudet – aivan samoin kuin muitakin verkkosovelluksia arvioitaessa. Tämän kokonaisuuden piiriin kuuluvat sekä kokonaiskäyttöliittymä että sivustolla vallitseva helppokäyttöisyyden aste navigointi- ja hakutoimintoihin.

Seuraavassa ala-alaluvussa johdetaan RIA-verkkosovellusten ominaispiirteistä käytettävyysvaatimuksia. Toisessa ala-alaluvussa käydään läpi RIA-verkkosovellusten käytettävyyteen vaikuttavia seikkoja konventioihin liittyen. Kolmannessa ala-alaluvussa käsitellään RIA-verkkosovellusten löydettävyyttä ja sijaintitietoa käytettävyyden näkökulmasta.

### 5.2.1 RIA-verkkosovellusten ominaispiirteet käytettävyysvaatimusten lähteenä

Palmerin (2002) mukaan sujuvasti toimiva navigaatio ja interaktiivisuus, lyhyet vasteajat sekä järjestelmän antaman palautteen korkea taso ovat tekijöitä, jotka olisi sisällytettävä verkkosovelluksiin hyvän käytettävyyden saavuttamiseksi. Perinteisiä verkkosovelluksen toteutustapoja noudattaen edellä luetellut vaatimukset ovat olleet haastavia toteuttaa. RIA-verkkosovellusten teknologiat ja toteutusmallit ovat helpottaneet tilannetta, mutta nekään eivät automaattisesti tuo ratkaisuja. Loogisen tason navigaatoratkaisut ja järjestelmän antaman palautteen sisällöllisesti korkea taso eivät ole toteutusteknologiasta riippuvia, ja niiden arviointiin ja suunnitteluun voidaan RIA-verkkosovelluskontekstista

huolimatta soveltaa hyvin olemassa olevia käytettävyyden asiantuntija-arvioinnin menetelmiä ja tietämystä.

Toiminnallisuudeltaan perinteisiä verkkosovelluksia rikkaampien RIA-verkkosovellusten käytettävyyksivaatimuksia voidaan johtaa muista sovellustyypeistä, joista useimmiten mainitaan työpöytäsovellukset. Työpöytäsovellukseksi voidaan laskea myös useat tietokonepelit. Sekä hyöty- että viihdesovellusten käyttöliittymät vaihtelevat runsaasti ulkoasultaan ja toiminnaltaan, mutta ne jakavat esimerkiksi yleiset vaatimukset tehokkuudesta ja nopeista vasteajoista. Koska kysymys on yleensä laajaa yleisöä havittelevista massatuotteista, on niiden käyttöliittymien oltava erittäin helppokäyttöisiä ja nopeasti opittavissa. Myös pelien käyttöliittymät soveltuvat esikuvaksi, koska tietokonepelien haastavuus perustuu yleensä muihin tekijöihin kuin käyttöliittymän vaikeaan hallintaan. Pelisuunnittelussa onkin ohjenuorana, että peli on helppo oppia, mutta muuten vaikea hallita. Hyötysovelluksissa myös hallinnan tulisi olla helppoa. (Jørgensen 2004) RIA-verkkosovelluksissa hallintaa helpottaa osaltaan jouhevasti toimiva käyttöliittymätaso.

RIA-verkkosovellusten tulisi ylittää nopeudessa perinteiset verkkosovellukset (esim. Paulson 2005; Smith 2006; Weiss 2006). Palmerin (2002) mukaan käyttäjän mielenkiinto alkaa herpaantua vajaan kymmenen sekunnin vasteajan jälkeen ratkaisevalla tavalla, eikä tilanne helpotu odotusajan kasvaessa. Mitä tutumpi käytetty sovellus on, sitä kriittisempää on suhtautuminen hitaaseen vasteaikaan. Oletettavasti laajakaistojen lisääntyminen on tehnyt ihmiset viime vuosina entistä kriittisemmiksi latausajoista riippuvaa viivettä kohtaan. Käyttäjät eivät viihdy tai vieraile yhtä usein sivuilla, jotka toimivat heidän mielestään hitaasti. Mikäli kyse on liiketoiminnasta, vähentyneet vierailukerrat ja lyhentyneet käyttöajat nakertavat myös verkon kautta saatavia voittoja. (Smith 2006)



RIA-verkkosovellusten nopeusedun tulisi näkyä heti käytön alkuhetkistä alkaen, eli käyttö pitäisi pystyä aloittamaan hyvin pian latauksen alettua. Käytettävyyden kannalta on myös kriittistä, ettei sovellus hyydy myöskään myöhemmissä vaiheissa. (Korhonen & Koivisto 2006) Näistä vaatimuksista tekee erityisen huomionarvoisia se, että RIA-verkkosovellukset lataavat usein sovellusta käynnistettäessä käyttäjän koneelle perinteisiin verkkosovelluksiin nähden runsaan määrän dataa. Pitemmällä aikavälillä tarkasteltuna yhteenlaskettu tiedonsiirron tarve on kuitenkin yleensä vähäisempää perinteisiin verkkosovelluksiin verrattaessa. (Crane, Pascarello & James 2006, 20) Verkkosovellusten dynaamisuus lisää myös osaltaan viivettä, jolloin vasteaika muodostuu tiedonsiirron lisäksi palvelimen toiminnasta sen generoidessa asiakkaan pyyntöjä vastaavia sisältöjä (Ghosh & Rau-Chaplin 2006). RIA-verkkosovellusten mahdollinen hitaus voi johtua myös asiakkaan koneella suoritettavan koodin runsaasta määrästä hitaan tai jatkuvan tiedon siirron sijaan (Holzner 2006, 328).

Palmer (2002) mainitsee interaktiivisuuden ja järjestelmän antaman palautteen tason merkityksen hyvän käytettävyyden saavuttamisessa. Desurvire, Caplan ja Toth (2004) esittävät, että käyttäjän suorittamien ensimmäisten toimenpiteiden pitäisi olla aina äärimmäisen ilmeisiä ja niistä pitäisi saada välitön positiivinen palaute. Interaktiota ja palautetta varten tarvitaan jouhevaa tiedon kulkua kumpaankin suuntaan käyttäjän ja sovelluksen välillä, mikä liittyy osin myös edellä käsiteltyyn vasteaikaan. Jos RIA-verkkosovelluksen tiedonsiirto tapahtuu taustalla sovelluksen muun käytön jatkuessa, voi suoritettava aktiviteetti jäädä käyttäjältä havaitsematta. RIA-verkkosovelluksen tulisikin antaa tietoa myös sellaisista tapahtumista, joita käyttäjä ei ole itse varta vasten aloittanut. Holzner (2006, 326) suositteleeekin tällaisiin tilanteisiin jonkinlaista graafista indikaattoria, joka ilmoittaa käyttäjälle sovelluksen tilasta ja toiminnasta. Palaute voi olla myös sanallista.

Jos verkkosovellus hyödyntää RIA-verkkosovelluksille ominaista tekniikkaa, mutta ei täytä muutoin RIA-verkkosovelluksille asetettuja yleisiä ja osin myös ei-toiminnallisia vaatimuksia, onko oikeastaan edes kyse RIA-verkkosovelluksesta? Termi RIA-verkkosovellus pitää mielestäni sisällään lupauksen perinteisiä verkkosovelluksia korkeammasta käytettävyyden tasosta, joka vain toteutuu tietyn teknologisen ratkaisun ja sen ominaisuuksien kautta.

### 5.2.2 Konventiot

Sovellusta tarjoavan tahon on tärkeää ymmärtää käyttäjien mahdollisia sumeita odotuksia, koska vaikka nämä eivät osaa ilmaista niitä tietoisesti, ne vaikuttavat kuitenkin siihen miten tyytyväisiä he ovat (Grönroos 2001, 136). Käyttäjät eivät käytä palveluita ja tuotteita tyhjiössä ja vertailua sovellusten kesken tapahtuu varmasti. Käyttäjät ovat tottuneet tiettyihin asioihin, esimerkiksi riittävän korkeatasoiseen käytettävyyteen ja huoliteltuun ulkoasuun sekä tiettyihin toimintatapoihin. Alun perin eksplisiittisistä vaatimuksista tulee ajan myötä implisiittisiä (Grönroos 2001, 138). Grönroosin (2001, 137) mukaan käyttäjien odotuksena on, että myös kaikki implisiittiset odotukset täytetään. Niiden oletetaan olevan niin ilmeisiä, ettei niitä ehkä koskaan vaadita erikseen. Odotusten olemassaolo käy kuitenkin ikävällä tavalla ilmi, kun niitä ei täytetä ja seuraa pettymys eikä sovelluksen käyttäjinä enää viihdytä.

RIA-verkkosovellukset hyödyntävät osin verkkosovelluksista tuttuja konventioita, mutta ne ovat imeneet vaikutteita lisäksi esimerkiksi työpöytäsovelluksista ja tietokonepeleistä. Dyckin, Pinellen, Brownin ja Gutwinin (2003) mukaan pelien ja viihdesovellusten käyttöliittymistä löytyy paljon elementtejä, joilla voidaan parantaa myös hyötysovellusten käytettävyyttä. Ongelmia voi kuitenkin ilmaantua, jos RIA-verkkosovelluksen käyttäjällä ei ole kokemusta peleistä ja

niiden käyttöliittymien logiikasta. Toisaalta, kun riittävän moni RIA-toteutus hyödyntää samaa ratkaisumallia, tulee siitä uusi konventio. Jos RIA-verkkosovellus muistuttaa hyvin paljon työpöytäsovellusta, voi käyttäjä helposti olettaa löytävänsä siitä työpöytäsovelluksista tutut yleiset toiminnot. Tällaisia toimintoja ovat esimerkiksi Undo- ja Redo- toiminnot, joilla on mahdollista kumota tai toistaa äskettäisiä toimenpiteitä ja komentoja. (Farrell & Nezelek 2007)

Hyvä esimerkki RIA-verkkosovelluksen konventioiden vastaisesta toiminnasta on verkkoselaimen Takaisin-painikkeen tekeminen hyödyttömäksi. Kun RIA-toteutukset eivät jätä jälkiä verkkoselaimen historiatietoihin, myöskään Takaisin-painike ei pysty esittämään RIA-verkkosovelluksen aiempia näkymiä. (Asleson & Schutta 2006, 17; Holzner 2006, 325) Verkkoselaimen Takaisin-painike on oleellinen osa navigointia, jonka varassa käyttäjät voivat tutkia verkkosisältöjä vapaasti ja lähes riskittömästi, navigaatiotoimenpiteiden peruutusmahdollisuuteen luottaen. Takaisin-painike on toiseksi käytetyin navigaatiotoiminto verkossa heti hyperlinkein tapahtuvan navigoinnin jälkeen. Jos tällainen toiminto otetaan yllättäen pois, on tuloksena katastrofaalinen käytettävyysongelma. (Nielsen 2005b)

Se että suurin osa ihmisistä toimii jollakin tavalla, on käytettävyyden näkökulmasta riittävän hyvä perustelu toimia myös jatkossa samalla tavalla. Käyttäjät tuntevat yleisen käytännön ja siihen pohjaavien järjestelmien käyttö on kaikkein helpointa. (Nielsen 2000, 246) Sovelluksen käytön oppiminen on nopeampaa, kun sovellus vastaa käyttäjän odotuksia ja kun käyttäjä voi soveltaa aiemmista ympäristöistä tuttuja toiminnan malleja (Desurvire, Caplan & Toth 2004). Samaa konventioita mukailevaa logiikkaa tulisi noudattaa myös RIA-verkkosovellusten teknisiä ratkaisuja miettiessä, ainakin jos on epävarma oman ratkaisun pätevyydestä (Asleson & Schutta 2006, 20).

Ongelmat RIA-verkkosovellusten navigaatioissa ja käytettävyydessä kumpuavat usein juuri konventioiden vastaisista ratkaisuista. RIA-verkkosovelluksen kehittäjän tulisikin uusia toteutustekniikoita soveltaessaan pysähtyä pohtimaan, rikkooko toteutus konventioita. Hyvin perusteltuna konventioita saa ja pitääkin rikkoa, mutta tarpeeton käytön hankaloittaminen on aina pahasta.

### 5.2.3 Sovelluksen sijainti ja löydettävyys

Kun tieto voidaan saada ulkomaailmasta, ilman että sitä tarvitsee muistaa tarkasti, on oppimisen tarve vähäistä (Norman 1991, 87-88). Verkkosovellusten kontekstissa ulkoisen tiedon lähde on yleensä näyttö. Kun tietoa on riittävästi ulkomaailmassa, ihmiset pystyvät toimimaan hyvin ympäristössään kykenemättä silti kuvailemaan mitä he tekevät. He voivat esimerkiksi suunnistaa oikein kaupungin läpi kykenemättä kuvailemaan reittiä täsmällisesti. (Norman 1991, 89) Sama pätee myös WWW-ympäristössä tapahtuvaan liikkumiseen ja löytämiseen. Kuitenkin huonolla sovellussuunnittelulla ja RIA-verkkosovellusten toteutustekniikoita varomattomasti käyttäen verkkoselainten perustoiminnallisuus sijaintitietoutta koskien voidaan rampauttaa. Esimerkiksi kirjanmerkit ja selaimen osoitekenttä eivät välttämättä toimikaan, kuten niiden pitäisi toimia.

RIA-verkkosovelluksilla voi olla ongelmia löydettävyyden suhteen myös hakukoneiden kanssa. Koska RIA-verkkosovellukset eivät useinkaan lataa koko sivua uusiksi sisällön päivittyessä, vaan muuttavat selainäkymää vain osittain tietyistä kohdista, voi tällainen aineisto jäädä hakukoneilta huomaamatta. Tällöin verkkosovellus tai sen osa ei näy hakukoneen hakemistoissa eikä myöskään linkkinä verkkoa selailevien käyttäjien hakutuloksissa. (Paulson 2005) Kun vain osa sivusta muuttuu, koko sivun päivittymiseen tottunut käyttäjä ei vält-

tämättä huomaa näkymässä tapahtunutta muutosta (Holzner 2006, 327). Tähän on kehitetty yhdeksi ratkaisuksi niin kutsuttu keltaisen huomioväriin tekniikka (*engl. Yellow Fade Technique*), jossa muuttunut osio osoitetaan värityksellä, esimerkiksi juuri keltaisella (Asleson & Schutta 2006, 16).

Toisin kuin perinteisten verkkosovellusten ollessa kyseessä, RIA-verkkosovellukset eivät useimmiten muuta selainohjelmiston osoiterivillä näkyvää osoitetta käytön aikana. Esimerkiksi kymmenen erilaista näkymää sisältävä RIA-verkkosovellus voi näyttää koko ajan samaa osoitetta selaimen osoiterivillä, riippumatta mikä kohta tai sovelluksen tila on esillä. Kun käyttäjä merkitsee mielenkiintoiseksi kokemansa kohdan selaimen kirjanmerkkeihin, ei kirjanmerkkikokoelmaan tallennukaan havaittu näkymä, vaan tyypillisesti RIA-verkkosovelluksen alkutilaan johtava osoitetieto. Vastaavalla tavalla oikeaan kohtaan osoittavan osoitetiedon lähettäminen esimerkiksi sähköpostilla epäonnistuu. Voi olla myös toivottava tilanne, että RIA-verkkosovellukseen tullaan aina tietyn aloituskohdan kautta sovelluksen jälkimmäisten osien ollessa ainoastaan tulos käyttäjän erityisistä toimenpiteistä. Esimerkiksi karttasovelluksen kohdalla olisi kuitenkin käyttäjäystävällistä päästää käyttäjä suoraan tiettyyn yksittäiseen karttanäkymään suoralla osoitteella, kulkematta ensin laajan maailmankartan kautta. (Asleson & Schutta 2006, 490)

Sovellus pitäisi pystyä sulkemaan välillä ja sen pariin palaamisen tulisi olla helppoa, mieluiten samaan sovelluksen osaan tai tilaan, josta poistuttiin (Desurvire, Caplan & Toth 2004). Jos verkkoselaimen kirjanmerkkeihin laitettu osoite johtaakin RIA-verkkosovelluksen alkutilaan eikä kohtaan, jossa aiemmin oltiin menossa, ei tämä vaatimus käytön jaksottamisesta ja helposta paluusta keskeneräisen tehtävän pariin toteudu.

### 5.3 Tarkastuslista RIA-verkkosovellusten käytettävyyden asiantuntija-arvioinnin tueksi

Tunnettuja ja yleisesti hyödynnettyjä käytettävyyden asiantuntija-arvioinnin menetelmiä, kuten Nielsenin (1994, 30) kymmenen heuristiikan listaa tai kognitiivista läpikäyntiä (Wharton ym. 1994), ei ole luotu RIA-verkkosovelluksia silmälläpitäen. Ne kattavat laajasti tekijöitä, jotka on syytä huomioida sovellustyyppistä riippumatta käytettävyyttä arvioitaessa tai käytettävää tuotetta suunniteltaessa. Luvussa 4 esitellyt asiantuntija-arvioinnin menetelmät muodostavat käytettävyyttä arvioitaessa ja käytettävyydeltään korkeatasoisia tuotteita laadittaessa onnistuneen toteutuksen kivijalan. Perinteisiä käytettävyyden asiantuntija-arvioinnin menetelmiä käyttäen ei ole kuitenkaan mahdollista sisällyttää suunnitteluun ja arviointiin kaikkia uusia RIA-verkkosovelluksissa ilmeneviä piirteitä ja ominaisuuksia (Asleson & Schutta 2006, 16).

Tässä luvussa on johdettu tutkielmassa esiteltyjä RIA-verkkosovellusten ominaisuuksia hyödyntäen käytettävyyteen liittyviä vaatimuksia. Havaitut käytettävyydsvaatimukset ovat suhteessa toisiinsa (TAULUKKO 9). Käytettävyyden suunnittelun ja asiantuntija-arvioinnin kivijalkana toimivat luvussa 4 esitellyt perinteiset asiantuntija-arvioinnin menetelmät. Tarkastuslista RIA-verkkosovellusten käytettävyyden asiantuntija-arvioinnin tueksi rakentuu tämän varaan. Ensimmäiseksi tulisi tarkastella, täyttääkö arvioitava verkkosovellus nimenomaan RIA-verkkosovellukselle ominaiset piirteet. Mikäli ehdot eivät täyty, riittävät käytettävyyden arviointiin todennäköisesti perinteiset asiantuntija-arvioinnin menetelmät sellaisenaan. Yleisten RIA-verkkosovellusten käytettävyyttä edistävien piirteiden muodostaman katon ja perinteisten käytettävyyden asiantuntija-arvioinnin menetelmien muodostaman kivijalan väliin voidaan paketoita edelleen yksityiskohtaisemmat RIA-verkkosovellusten käytettävyy-

vaatimukset, liittyivätpä ne käyttäjäkokemukseen tai perinteisemmin miellettyyn käytettävyyden käsitteeseen.

**TAULUKKO 9.** Tarkastuslista RIA-verkkosovellusten käytettävyyden asiantuntija-arvioinnin tueksi.

<p>Onko RIA-verkkosovelluksen toteutustekniikkaa pystytty hyödyntämään niin, että RIA-verkkosovellusten käytettävyyttä edistävät piirteet, nopeus, sujuva ja välitön reagointi käyttäjän toimiin sekä runsas palaute, toteutuvat? (Desurvire, Caplan &amp; Toth 2004; Holzner 2006; Korhonen &amp; Koivisto 2006; Paulson 2005; Smith 2006; Weiss 2006)</p>	
<p>Käyttäjäkokeusta tukevat RIA-verkkosovellusten käytettävyyden vaatimukset:</p>	<p>Käytettävyyttä tukevat RIA-verkkosovellusten käytettävyyden vaatimukset:</p>
<p>1) Onko RIA-verkkosovelluksen ulkoasu miellyttävä ja esteettinen? (Grönroos 2001; Mattelmäki &amp; Batarbee 2000; Norman 2002; Palmer 2002; Shneiderman 2004)</p>	<p>1) Noudattaako RIA-verkkosovellus konventioita? (Asleson &amp; Schutta 2006; Desurvire, Caplan &amp; Toth 2004; Grönroos 2001; Nielsen 2000)</p>

(jatkuu)

TAULUKKO 9. (jatkuu)

<p>2) Sisältääkö RIA-verkkosovellus käytettävyyttä edistäviä hauskoja ominaisuuksia, jotka sopivat luontevasti muuhun sisältöön ja käyttöliittymään? (Nielsen 2000; Nielsen 2002; Norman 2002; Pember-ton 2004; Shneiderman 2004)</p>	<p>2) Onko RIA-verkkosovellus ja tieto sen sijainnista helposti saatavilla ja löydettävissä? (Asleson &amp; Schutta 2006; Desurvire, Caplan &amp; Toth 2004; Norman 1991; Paulson 2005)</p>	
<p>Perinteisten käytettävyyden asiantuntija-arvioinnin menetelmien muodostama kivijalka.</p>		

Tarkastuslista RIA-verkkosovellusten käytettävyyden asiantuntija-arvioinnin tueksi on tarkoitettu käytettäväksi perinteisten käytettävyyden arvioinnin menetelmien kanssa. Tarkastuslistassa on pyritty huomioimaan RIA-verkkosovellusten laaja kirjo pitämällä periaatteet riittävän yleisellä tasolla. Esi-  
 tettyä mallia on kuitenkin mahdollista muokata hienojakoisempaan suuntaan, jossa yksittäisiä, esitettyihin periaatteisiin liittyviä tekijöitä voidaan nostaa esiin. Kuten muitakin heuristiikka- ja ohjelistoja, voidaan myös tätä muokata ja laa-  
 jentaa sekä syventää tai käyttää vain osin. Tällaisenaan listan on tarkoitus toi-  
 mia mahdollisimman kattavasti RIA-verkkosovellusten käytettävyyden asian-  
 tuntija-arvioinnin tukena sekä muistilistana sovellusten kehittäjille.

Tarkastuslista RIA-verkkosovellusten asiantuntija-arvioinnin tueksi sisältää yleisluontoisuutensa takia myös jossain määrin yhtymäkohtia tutkielmassa ai-  
 emmin esitettyihin heuristisiin listoihin. Periaatteessa osa uuden tarkastuslistan  
 kohdista voitaisiin katsoa katetuksi esimerkiksi Nielsenin Kymmenen heuristii-



kan listaa (Nielsen 1994, 30) ja Shneidermanin Kahdeksaa kultaista sääntöä dialogin suunnitteluun käyttäen (Shneiderman 1992, 72-73). Toisaalta Nielsenin ja Shneidermanin vaatimukset voivat aiheuttaa myös ristiriitoja RIA-verkkosovellusten luonteen takia. Tarkastuslistan tavoite on laajentaa ja tukea olemassa olevia käytettävyysvaatimuksia mutta myös poistaa tulkinnanvaraisuuksia ja tarkentaa vaatimuksia RIA-verkkosovelluksia koskien.

Esimerkiksi perusvaatimus esteettisestä suunnittelusta kattaa sovelluksen kuin sovelluksen, mutta minimalistisuus ei ole ominaista rikkaille verkkosovelluksille (Nielsen 1994, 30). Vastaavasti vaatimus palvelun tilan näkyvyydestä on helppo ymmärtää suppeimmalla mahdollisella tavalla koskemaan vain itse sovellusta, jolloin RIA-verkkosovellusten moninaiset ilmenemisulottuvuudet internetissä saattavat jäädä paitsioon (Nielsen 1994, 30). Shneiderman kehottaa noudattamaan toimintatavoissa yhteneväisyyttä ja Nielsen laajentaa vastaavaa vaatimusta myös standardien käytöllä (Nielsen 1994, 30; Shneiderman 1992, 72-73). RIA-verkkosovelluksissa käytetyt ratkaisut eivät kuitenkaan ole vielä virallisia standardeja kaikilta osin. Konventiot ovat hyvä käsite, jolla voidaan käsitellä RIA-verkkosovellusten johdonmukaisuutta ottamatta kantaa standardeihin ja laajentaa tarkastelukulma sovelluksen sisältä myös yleisemmälle tasolle. Toisaalta taas RIA-verkkosovelluksissa ilmenevät konventiot eivät ole välttämättä olleet konvention asemassa aiemmin. Näin ollen on perusteltua teroittaa sekä sovelluksien kehittäjille että niiden käytettävyyden arvioijille RIA-verkkosovellusten erityispiirteitä, joita ei välttämättä tule ajatelleeksi pelkästään aiempia arviointivälineitä käyttäen. Tätä kautta tarkastuslistan tukifunktio perinteisten menetelmien lisänä saa vahvistusta.

Onnistuneen konstruktion vaatimuksena on kohdeympäristön ja sen käsitteiden hyvä tuntemus. Muutoin lopputulos voi olla jopa virheellinen ja aiheuttaa ei-toivottuja sivuvaikutuksia käytäntöön sovellettaessa. Tässä tutkielmassa esi-

tetty tarkastuslista ei ole kuitenkaan vielä ollut todellisessa käytössä. Tarkastuslistan toimivuus voidaan kyseenalaistaa myös silloin, jos tarkastuslista ei toimi yksinään vaan vaatii käyttäjältään tarkempaa perehtymistä tarkastuslistan luonnissa käytettyyn aineistoon esitettyjen vaatimusten todellisen merkityksen ymmärtämiseksi. On syytä uskoa, että tarkastuslistan soveltaminen käytännössä RIA-verkkosovellusten arviointi- ja kehitystyöhön voisi paljastaa siitä puutteita ja ohjaisi jatkokehitystä suotuisaan suuntaan. Myös laajennettu jatkotutkimus voisi hyödyttää tarkastuslistan parantamistyössä. Kuten sovelluksia, myös teoreettista mallia voidaan parantaa toistuvaa rakenna ja arvioi -sykliä noudattaen.

## 6 YHTEENVETO

Tämän tutkielman tekemisen yhtenä kannustimena on se seikka, että verkkosovellusten määrä ja merkitys ovat kasvaneet valtavasti ja ne muodostavatkin tätä nykyä yhden suurimmista ja merkittävimmistä ohjelmistoteollisuuden osa-alueista. Myös verkkosovelluksia luotaessa voidaan hyödyntää vanhoja ja hyväksi koettuja ohjelmistojen tuotantomalleja sovittaen niitä uuteen ympäristöön. Vaikka tämä koskee myös käytettävyyssajattelua ja käytettävyyssuositusten soveltamista, on kehitykselle edelleen luonteenomaista käytettävyyssuositusten laiminlyönti. Tutkielmassa onkin noteerattu asiantuntevan suunnittelijan oman satsauksen käytettävyydestä huolehtimiseen olevan erityisen arvokas.

Tutkielman päätavoitteena on ollut selvittää, millä tavalla RIA-verkkosovellukset eroavat perinteisistä WWW-verkkosovelluksista ja kuinka RIA-verkkosovellusten käytettävyyttä ja käyttäjäkokemusta olisi mahdollista arvioida asiantuntija-arvioinnin menetelmin. Tutkielmassa on käyty läpi perinteisten asiakas-palvelin -arkkitehtuuria noudattavien WWW-verkkosovellusten ja RIA-verkkosovellusten ominaispiirteitä ja toiminnan mekaniikkaa. Edelleen on pohdittu, millaisia käytettävyyssuositteita tällaisille toimintaympäristöille voidaan ja tulee asettaa sekä kuinka käytettävyyttä voidaan arvioida asiantuntija-arvioinnin keinoin. Painopiste tarkastelussa on ollut käytettävyydessä ja verkkosovelluksen sekä käyttäjän välisessä vuorovaikutussuhteessa, jolloin päähuomio on ollut syytä suunnata erityisesti verkkosovellusten käyttöliittymätasolle.

Tutkielman perustana on toiminut moniselitteiseksi osoittautunut käsite käytettävyys, joka on esitelty ja määritelty monelta tasolta ja eri näkökulmista. Käytettävyys on kokonaisuudessaan menetelmä- ja teoriakenttä, jonka kautta käyttäjän ja laitteen yhteistoiminta pyritään saamaan tehokkaammaksi ja miellyttä-

vämmäksi. Tutkielmassa on myös käsitelty käyttäjäkokemusta, joka läheisesti liittyy käytettävyyteen. Käyttäjäkokemuksen keskiössä ovat käyttäjä ja tuote, jotka sijaitsevat aina jossakin tietyssä ympäristössä ja edelleen sosiaalisessa ja kulttuurisessa tilassa. Tuote vaikuttaa yksilön kokemukseen ominaisuuksillaan, kuten toiminnallisuudella, muotokielellä ja estetiikalla. Yksittäinen kokemus muodostuu aina lukemattomista pienemmistä kokemuksista ja myös satunnais-tekijöillä on sijansa syntyprosessissa.

Käytettävyyden asiantuntija-arvioinnin menetelminä on käyty läpi heuristinen arviointi, kognitiivinen läpikäynti sekä automaattiset arviointimenetelmät. Asiantuntija-arviota kutsutaan monella eri nimellä ja esimerkiksi heuristista arviointia ja asiantuntija-arviointia käytetään usein synonyyminä. Asiantuntija-arviot ovat analyttisiä menetelmiä, joissa käytettävyyсарviointi on ennakoivaa. Tällöin tutkitaan, mikä sovelluksen ominaisuus tai toiminto tulee olemaan mahdollinen käytettävyyssongelma.

Kognitiivisessa läpikäynnissä yritetään mallintaa käyttäjän ajatuksia ja toimintaa, kun käyttöliittymä kohdataan ensimmäistä kertaa. Menetelmä selvittää, onko käyttäjän helppo ymmärtää ja oppia käyttämään käyttöliittymää. Käytettävyyden automaattinen arviointi tarkoittaa automatisoitujen työkalujen hyödyntämistä käytettävyyden arviointiin ja testaukseen liittyvän aineiston keräämisessä ja analysoinnissa. Automaattiset menetelmät sopivat paremmin esimerkiksi suoraviivaisten tiedonhakutehtävien kuin monimutkaisten verkkosovellusten testaamiseen. Käytettävyyden heuristinen arviointi perustuu heuristiikkoihin, jotka ovat käytettävyyssperiaatteita, sääntöjä ja ohjeistuslistoja. Suurimman painoarvon käytettävyyden asiantuntija-arvioinnin menetelmistä on saanut Nielsenin (1994) kymmenen heuristiikan lista, josta on tehty myös soveltava esimerkkitapaus. Nielsenin heuristiikkojen juuret ovat ajalta ennen

WWW:n läpimurtoa, mutta ne soveltuvat joustavuutensa ansiosta myös nykyisin moneen käyttöön.

Perinteistä asiakas-palvelin -mallia toteuttavien verkkosovellusten rinnalle on ilmaantunut uudenlaisia sovelluksia, niin kutsutut RIA-verkkosovellukset. RIA-verkkosovelluksille on ominaista visuaalinen näyttävyys ja rikas sekä joustava toiminnallisuus. RIA-verkkosovelluksia käsittelevässä osassa on kartoitettu, kuinka verkkosovellukset ja niiden toimintaympäristö ovat muuttuneet viime aikoina. Uudenlaisten sovellusten ja muuttuneen toimintaympäristön on todettu asettavan uudenlaisia haasteita myös käytettävyyden arvioinnille.

Tunnettuja ja yleisesti käytettyjä asiantuntija-arvioinnin menetelmiä ei ole luotu RIA-verkkosovelluksia silmälläpitäen. Ne kattavat laajasti tekijöitä, jotka on syytä huomioida sovellustyypistä riippumatta käytettävyyttä arvioitaessa tai käytettävää tuotetta suunniteltaessa. Perinteiset asiantuntija-arvioinnin menetelmät muodostavat käytettävyyttä arvioitaessa ja käytettävyydeltään korkeatasoisia tuotteita laadittaessa onnistuneen toteutuksen kivijalan. Perinteisiä asiantuntija-arvioinnin menetelmiä käyttäen ei ole kuitenkaan mahdollista sisällyttää käytettävyyden suunnitteluun ja arviointiin kaikkia uusia RIA-verkkosovelluksissa ilmeneviä piirteitä ja ominaisuuksia.

Tutkielmassa esiteltyjä RIA-verkkosovellusten ominaisuuksista hyödyntäen johdettiin edelleen käytettävyyteen ja käyttäjäkokemukseen liittyviä vaatimuksia, joiden pohjalta on konstruoitu tutkielman keskeisenä tuloksena tarkastuslista RIA-verkkosovellusten asiantuntija-arvioinnin tueksi. Tarkastuslista kattaa RIA-verkkosovelluksista havaitut keskeiset uudet piirteet yleisellä tasolla. Tarkastuslista on samalla malli, jota voidaan kasvattaa tarpeen mukaan uusia RIA-verkkosovellusten piirteitä havaittaessa. Tarkastuslistaa ei ole koeteltu käytännön kehitystyössä, mikä mahdollisesti toisi esiin tarpeen listan laajentamisesta

määrällisesti tai hienojakoisemmaksi. Myös laajempi kartoittava jatkotutkimus voisi lisätä listan kattavuutta.

Aiheen käsittely perustui kriittiseen kirjallisuuden tutkimiseen. Perehtymällä artikkelien sisältämään tietoon oli mahdollista luoda yleiskäsitys käytettävyydestä, erilaisista verkkosovelluksista sekä verkkosovelluksien käytettävyyden asiantuntija-arvioinnin menetelmistä sekä soveltamismahdollisuuksista ja haasteista.

## LÄHDELUETTELO

- Adobe Systems Inc. 2007. Planning for RIA success [online]. Adobe Systems Inc. [viitattu 1.9.2007]. Saatavilla [www-osoitteessa <http://www.adobe.com/devnet/flex/articles/planning\\_ria/planning\\_ria.pdf>](http://www.adobe.com/devnet/flex/articles/planning_ria/planning_ria.pdf).
- Asleson, R. & Schutta N. 2006. Foundations of Ajax, New York: Apress.
- Blackmon M., Polson P., Kitajima M. & Lewis C. 2002. Cognitive walkthrough for the web. Teoksessa Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves, Minneapolis, Minnesota, USA, April 20 - 25. New York: ACM Press, 463-470.
- Blackmon M., Kitajima M. & Polson P. 2005. Tool for accurately predicting website navigation problems, non-problems, problem severity, and effectiveness of repairs. Teoksessa Proceedings of the SIGCHI conference on Human factors in computing systems, Portland, Oregon, USA, April 02-07. New York: ACM Press, 31-40.
- Blatt L. & Knutson J. 1994. Interface Design Guidance Systems. Teoksessa Nielsen J. & Mack R.L. (toim.) Usability Inspection Methods. New York, NY: John Wiley & Sons, 351-384.
- Boll S. 2007. MultiTube – Where Multimedia and Web 2.0 Could Meet. IEEE Multimedia, 14(1), 9-13.
- Brooks P. 1994. Adding Value to Usability Testing. Teoksessa Nielsen J. & Mack R.L. (toim.) Usability Inspection Methods. New York, NY: John Wiley & Sons, 255-271.
- Carroll J. 2004. Beyond Fun. Interactions 11(5), 38-40.
- Chao D. 2004. Computer Games as Interfaces. Interactions 11(5), 71-72.

- Chi E., Pirolli P. & Pitkow J. 2000. The scent of a site: a system for analyzing and predicting information scent, usage, and usability of a Web site. Teoksessa Proceedings of the SIGCHI conference on Human factors in computing systems, Hague, Netherlands, April 01-06. New York: ACM Press, 161-168.
- Chi E., Rosien A., Supattanasiri G., Williams A., Royer C., Chow C., Robles E., Dalal B., Chen J. & Cousins S. 2003. The Bloodhound Project: Automating Discovery of Web Usability Issues using the InfoScent™ Simulator. Teoksessa Proceedings of the SIGCHI conference on Human factors in computing systems, Ft. Lauderdale, Florida, USA, April 05-10. New York: ACM Press, 505-512.
- Constantine L. & Lockwood L. 2002. Usage-centered engineering for web applications. IEEE software 19(2), 42-50.
- Crane D., Pascarello E. & James D. 2006. Ajax In Action. Greenwich: Manning.
- Desurvire H., Caplan M. & Toth J. 2004. Using heuristics to evaluate the playability of games. Teoksessa CHI '04 extended abstracts on Human factors in computing systems, Vienna, Austria, April 24-29. New York: ACM Press, 1509-1512.
- Dicks S. 2002. Mis-Usability: On the Uses and Misuses of Usability Testing. Teoksessa Proceedings of the 20<sup>th</sup> annual international conference on Computer documentation, Toronto, Ontario, Canada, October 20-23. New York: ACM Press, 26-30.
- Dyck J., Pinelle D., Brown B. & Gutwin C. 2003. Learning from Games: HCI Design Innovations in Entertainment Software. Teoksessa Proceedings of Graphics Interface 2003, Halifax, Nova Scotia, Canada, June 11-13. Toronto: Canadian Computer-Human Communications Society, 237-246.
- Farrell J. & Nezelek G. 2007. Rich Internet Applications - The Next Stage of Application Development. Teoksessa 29th International Conference on Information



Technology Interfaces, Cavtat/Dubrovnik, Croatia, June 25-28. USA: IEEE Computer Society, 413-418.

Forlizzi J. & Ford S. 2000. The building blocks of experience: an early framework for interaction designers. Teoksessa Proceedings of the Conference on Designing interactive systems: processes, practices, methods, and techniques, New York City, New York, USA, August 17-19. New York: ACM Press, 419-423.

Forlizzi J. & Battarbee K. 2004. Understanding experience in interactive systems. Teoksessa Proceedings of the 2004 Conference on Designing interactive systems: processes, practices, methods, and techniques, Cambridge, MA, USA, August 01-04. New York: ACM Press, 261-268.

Fraternali P. 1999. Tools and approaches for developing data-intensive Web applications: a survey. ACM Computing Surveys. 31(3), 227-263.

Garfinkel S. & Spafford E.H. 1997. Web Security & Commerce. USA: O'Reilly Media, Inc.

Gehlertland, J., Galbraith, B & Almaer, D. 2005. Pragmatic Ajax: A Web 2.0 Primer, USA: The Pragmatic Programmers, LLC.

Geis T., Dzida W. & Redtenbacher W. 2004. Specifying usability requirements and test criteria for interactive systems. Federal Institute for Occupational Safety and Health, Research report Fb 1010.

Ginige A. & Murugesan S. 2001. Web Engineering: an Introduction. IEEE Multimedia, 8(1), 14 -18.

Ghosh P. & Rau-Chaplin A. 2006. Performance of Dynamic Web Page Generation for Database-driven Web Sites. Teoksessa International Conference on Next Generation Web Services Practices, Seoul, Korea, September 25-28. USA: IEEE Computer Society, 56-63.

- Greenspan J. & Bulger B. 2001. MySQL/PHP Database Applications. USA: M&T Books.
- Grönroos C. 2001. Palveluiden johtaminen ja markkinointi. Porvoo: WSOY ja Suomen Ekonomiliitto.
- Hassan A. & Holt R. 2001. Towards better understanding of web applications. Software Architecture Group (SWAG), Department of Computer Science, University of Waterloo.
- Hassenzahl M. 2004. Emotions Can Be Quite Ephemeral; We Cannot Design Them. Interactions 11(5), 46-48.
- Heimonen T. 2005. Käytettävyyden automaattinen arviointi. Teoksessa Ovaska S., Aula A. & Majaranta P. (toim.) Raportti B-2005-1: Käytettävyytutkimuksen menetelmät. Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, 169-186.
- Holzner S. 2006. Ajax for Dummies. Hoboken: Wiley Publishing, Inc.
- Hong J., Heer J., Waterson S. & Landay J. 2001. WebQuilt: A proxy-based approach to remote web usability testing. ACM Transactions on Information Systems 19(3), 263-285.
- Hyppönen H. 2000. Mainstreaming teknologiasuunnittelussa. Teoksessa Keinonen T. (toim.) Miten käytettävyys muotoillaan?, B 61. Helsinki: Taideteollinen korkeakoulu, 106-127.
- ISO/IEC 9126. 1991. Information technology – Software product evaluation – Quality characteristics and guidelines for their use. International Standards Organisation & Internal Electrotechnical Commission.
- Ivory M.Y. & Hearst M.A. 2001. The State of the Art in Automating Usability Evaluation of User Interfaces. ACM Computing Surveys 33(4), 470-516.

- Jørgensen A. 2004. Marrying HCI/Usability and computer games: a preliminary look. Teoksessa Proceedings of the third Nordic conference on Human-computer interaction, Tampere, Finland, October 23-27. New York: ACM Press, 393 - 396.
- Kangas P., Toivonen S. & Bäck A. 2007. Googlen mainokset ja muita sosiaalisen median liiketoimintamalleja. VTT, VTT tiedotteita 2369.
- Keinonen T. 1998. One-dimensional usability – Influence of usability on consumers' product preference. Publication Series of the University of Art and Design, Helsinki UIAH A21. Gummerus.
- Keinonen T. 2000. Yksiulotteinen käytettävyys. Teoksessa Keinonen T. (toim.) Miten käytettävyys muotoillaan?, B 61. Helsinki: Taideteollinen korkeakoulu, 92-105.
- Korhonen H. & Koivisto E. 2006. Playability heuristics for mobile games. Teoksessa Proceedings of the 8th conference on Human-computer interaction with mobile devices and services, Helsinki, Finland, September 12-15. New York: ACM Press, 9-16.
- Korvenranta H. 2005. Asiantuntija-arvioinnit. Teoksessa Ovaska S., Aula A. & Majaranta P. (toim.) Raportti B-2005-1: Käytettävyystutkimuksen menetelmät. Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, 111-124.
- Koskinen I. 2000. Vuorovaikutussuunnittelun teoriasta ja metodeista. Teoksessa Keinonen T. (toim.) Miten käytettävyys muotoillaan?, B 61. Helsinki: Taideteollinen korkeakoulu, 92-105.
- Krug S. 2006. Älä pakota minua ajattelemaan! Tervettä järkeä verkkosuunnitteluun. Jyväskylä: Readme.fi.
- Kurose J. & Ross K. 2005. Computer Networking - a Top-Down Approach Featuring the Internet (3<sup>rd</sup> edition). USA: Addison Wesley.

- Lang L. & Fitzgerald B. 2005. Hypermedia Systems Development Practices: A Survey. *IEEE Software* 22(2), 68-75.
- Lauesen S. 2005. *User Interface Design. A Software Engineering Perspective*. King's Lynn: Addison Wesley.
- Lavery D., Cockton G. & Atkinson M.P. 1997. Comparison of evaluation methods using structured usability problem reports. *Behaviour & Information Technology* 16(4/5), 246-266.
- Lewis D. 2006. What is web 2.0? *Crossroads*, 13(1), 3-3.
- Mattelmäki T. & Battarbee K. 2000. Elämykset muotoilun lähtökohtana. Teoksessa Keinonen T. (toim.) *Miten käytettävyys muotoillaan?*, B 61. Helsinki: Taideteollinen korkeakoulu, 142-163.
- McDonald M. 2006. *Beginning ASP.NET 2.0 in C# 2005: From Novice to Professional*. USA: Apress.
- McNamara N. & Kirakowski J. 2006. Functionality, Usability, and User Experience: Three Areas of Concern. *Interactions* 13(6), 26-28.
- Microsoft 2007. Introduction to Dynamic HTML [online]. Microsoft Corporation [viitattu 14.10.2007]. Saatavilla [www-osoitteessa <http://msdn.microsoft.com/workshop/author/dhtml/dhtml.asp>](http://msdn.microsoft.com/workshop/author/dhtml/dhtml.asp).
- Mielonen S. & Hintikka K. 1998. Web-palveluiden käytettävyys ja tuotanto [online]. Helsinki: Taideteollinen korkeakoulu [viitattu 25.3.2007]. Saatavilla [www-muodossa <http://www2.uiah.fi/mediastudio/survey4/>](http://www2.uiah.fi/mediastudio/survey4/).
- Müller K. 2004. *Aivokutinaa*. Vammala: Työterveyslaitos.
- Newcomer E. 2002. *Understanding Web Services: XML, WSDL, SOAP, and UDDI*. USA: Addison-Wesley.

- Nielsen J. & Molich R. 1990. Heuristic Evaluation of User Interfaces. Teoksessa Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people, Seattle, Washington, USA, April 1-5. New York: ACM Press, 249-256.
- Nielsen J. 1992. Finding usability problems through heuristic evaluation. Teoksessa Proceedings of the SIGCHI conference on Human factors in computing systems, Monterey, CA, USA, May 3-7. New York: ACM Press, 373-380.
- Nielsen J. 1993. Usability Engineering. Academic Press.
- Nielsen J. 1994. Heuristic evaluation. Teoksessa Nielsen J. & Mack R.L. (toim.) Usability Inspection Methods. New York, NY: John Wiley & Sons, 25-62.
- Nielsen J. 1999. Voodoo Usability [online]. Useit.com/alertbox 12.12.1999 [Viitattu 25.7.2007]. Saatavilla [www-muodossa <URL: http://www.useit.com/alertbox/991212.html>](http://www.useit.com/alertbox/991212.html).
- Nielsen J. 2000. WWW-suunnittelu. Jyväskylä: IT Press.
- Nielsen J. 2002. User Empowerment and the Fun Factor [online]. Useit.com/alertbox 7.7.2002 [Viitattu 17.8.2007]. Saatavilla [www-muodossa <URL: http://www.useit.com/alertbox/20020707.html>](http://www.useit.com/alertbox/20020707.html).
- Nielsen J. 2004. Risks of Quantitative Studies [online]. Useit.com/alertbox 1.3.2004 [viitattu 2.9.2007]. Saatavilla [www-muodossa <URL: http://www.useit.com/alertbox/20040301.html>](http://www.useit.com/alertbox/20040301.html).
- Nielsen J. 2005. Severity Ratings for Usability Problems [online]. Useit.com 30.4.2005 [viitattu 17.7.2007]. Saatavilla [www-muodossa <URL: http://www.useit.com/papers/heuristic/severityrating.html>](http://www.useit.com/papers/heuristic/severityrating.html).

- Nielsen J. 2005b. Why Ajax Sucks (Most of the Time) [online]. [viitattu 11.11.2006].  
Saatavilla [www-muodossa](http://www.muodossa.com) <URL:  
<http://www.usabilityviews.com/ajaxsucks.html>>.
- Noda T. & Helwig S. 2005. Rich Internet Applications - Technical Comparison and Case Studies of AJAX, Flash, and Java based RIA. University of Wisconsin-Madison, UW E-Business Consortium.
- Norman D. 1991. Miten avata mahdottomia ovia? Tuotesuunnittelun salakarit. Jyväskylä: Gummerus.
- Norman D. 2002. Emotion and Design: Attractive things work better. *Interactions*, 9(4), 36-42.
- Norman D. 2006. Emotionally Centered Design. *Interactions*, 13(3), 53-ff.
- Offutt J. 2002. Quality attributes of web software applications. *IEEE Software*, 19(2), 25-32.
- Ovaska S., Aula A. & Majaranta P. 2005. Johdatus käytettävyytutkimukseen. Teoksessa Ovaska S., Aula A. & Majaranta P. (toim.) Raportti B-2005-1: Käytettävyytutkimuksen menetelmät. Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, 1-16.
- Pagulayan R. & Steury K. 2004. Beyond Usability in Games. *Interactions* 11(5), 70-71.
- Palmer, J. 2002. Designing for Web site usability. *IEEE Computer*, 35(7), 102-103.
- Paulson L. 2005. Building Rich Web Applications with Ajax. *IEEE Computer* 38 (10), 14-17.
- Pemberton S. 2004. Emotion. *Interactions*, 11(5), 4.

- Perfetti C. & Spool J.M. 2002. Macromedia Flash: A New Hope for Web Applications – Creating Powerful Web Applications with Macromedia Flash. User Experience White Paper. User Interface Engineering.
- Perälä R. 2005. Arvioijan vaikutus. Teoksessa Ovaska S., Aula A. & Majaranta P. (toim.) Raportti B-2005-1: Käytettävyystutkimuksen menetelmät. Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, 299-312.
- Preciado, J.C., Linaje, M., Sanchez, F. & Comai, S. 2005. Necessity of methodologies to model rich Internet applications. Teoksessa Seventh IEEE International Symposium on Web Site Evolution, Budapest, Hungary, September 26. USA: IEEE Computer Society, 7-13.
- Pressman. R.S. 2000. What a tangled Web we weave. IEEE Software 17(1), 18-21.
- Polvinen T. 1999. Tietokannat käytännön työssä. Porvoo: Teknolit.
- Powell T. 2002. Web Design: The Complete Reference (2<sup>nd</sup> Edition). Blacklick, OH, USA: McGraw-Hill Professional.
- Ranne S. 2005. Kognitiivinen läpikäynti. Teoksessa Ovaska S., Aula A. & Majaranta P. (toim.) Raportti B-2005-1: Käytettävyystutkimuksen menetelmät. Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, 125-140.
- Rantala A. 2005. Web-ohjelmointi. Porvoo: Docendo.
- Riihiaho S. 2000. Experiences with usability evaluation methods. Helsinki University of Technology, Department of Computer Science and Engineering, Laboratory of Information Processing Science, Licentiate's thesis.
- Rieman J. 1996. A field study of exploratory learning strategies. ACM Transactions on Computer-Human Interaction 3(3), 189-218.
- Roe C. & Gonic C. 2002. Server-side design principles for scalable Internet systems. IEEE Software, 19(2), 34-41.

- Saariluoma P. 2004. Käyttäjäpsykologia. Ihmisen ja koneen vuorovaikutuksen uusi ajattelutapa. Vantaa: WSOY.
- Sampson F. 2006. I Give That Web Site an 11. *Interactions* 13(3), 10-11.
- Shneiderman B. 1992. *Designing the user interface: strategies for effective human-computer interaction*. Reading: Addison Wesley.
- Shneiderman B. 2004. Designing for Fun: How Can We Design User Interfaces to Be More Fun? *Interactions* 11(5), 48-50.
- Sinkkonen I., Kuoppala H., Parkkinen J. & Vastamäki R. 2002. *Käytettävyyden psykologia*. Helsinki: IT Press.
- Smith, K. 2006. Simplifying AJAX-Style Web Development. *IEEE Computer* 39(5), 98-101.
- SFS-EN ISO 9241-11. 1998. Näyttöpäätteillä tehtävän toimistotyön ergonomiset vaatimukset. Osa 11: Käytettävyyden määrittely ja arviointi. Helsinki: Suomen standardoimiskeskus SFS.
- Sun Microsystems, Inc. 2005. Java™ Web Start Overview White Paper [online]. Sun Microsystems, Inc. [viitattu 1.9.2007]. Saatavilla [www-muodossa <URL: http://java.sun.com/developer/technicalArticles/WebServices/JWS\\_2/JWS\\_White\\_Paper.pdf>](http://java.sun.com/developer/technicalArticles/WebServices/JWS_2/JWS_White_Paper.pdf).
- Tarkka M. 2000. Käyttöliittymä kuvana - digitaalisen kuvaviestinnän retoriikka. Teoksessa Koskinen J. (toim.) *Visuaalinen viestintä - monialainen tulevaisuus*. Juva: WSOY, 105-117.
- Tractinsky N., Katz A.S. & Ikar I. 2000. What is beautiful is usable. *Interacting with Computers* 13(2), 127-145.



- Työterveyslaitos 2007. Duunitalkoot - Tehtävä: Suunnitelmakorit [online]. Työterveyslaitos [viitattu 3.8.2007]. Saatavilla www-muodossa <URL: [http://www.ttl.fi/internet/duunitalkoot/tyoyhteisot\\_p3.html](http://www.ttl.fi/internet/duunitalkoot/tyoyhteisot_p3.html)>.
- Wainewright P. 2007. What to expect from Web 3.0 | Software as Services [online]. CNET Networks, Inc. [viitattu 17.8.2007]. Saatavilla WWW-muodossa <URL: <http://blogs.zdnet.com/SAAS/?p=68>>.
- Weiss A. 2006. The Web Designers Dilemma: when standards and practice diverge. netWorker 10(1), 18-25.
- Wharton C., Rieman J., Lewis C. & Polson P. 1994. The Cognitive Walkthrough Method: A Practitioners Guide. Teoksessa Nielsen J. & Mack R.L. (toim.) Usability Inspection Methods. New York, NY: John Wiley & Sons, 105-140.
- Wixon D., Jones S., Tse L. & Casady G. 1994. Inspection and Design Reviews: Framework, History, and Reflection. Teoksessa Nielsen J. & Mack R.L. (toim.) Usability Inspection Methods. New York, NY: John Wiley & Sons, 77-104.
- Zakas N.C., McPeak J. & Fawcett J. 2006. Professional Ajax. Indianapolis: Wiley Publishing.