

Universaali käytettävyys

Käyttöliittymätutkimus-seminaari

Antti Martikainen (antti.martikainen@helsinki.fi)

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos
Helsinki 19.10.2001

Universaali käytettävyys

Antti Martikainen

Seminaariesitelmä

Tietojenkäsittelytieteen laitos

Helsingin yliopisto

19. lokakuuta 2001, 13 sivua

Monikanavaiset, useita päätelaitteita tukevat, www-palvelut ovat jatkossa internet-yhteensopivien mobiilien laitteiden yleistyessä yhä tärkeämpiä. Laitteiden teknisen variaansin tukeminen vaatii uudenlaisia arkkitehtuuriratkaisuja ja edellyttää käyttöliittymien kuvaamista uudenlaisten mekanismien avulla.

Uudenlaisten kuvausmekanismien käyttö vaatii myös nykyistä tiiviimpää yhteistyötä käytettävyysasiantuntijoiden ja sovellusarkkitehtuurisuunnittelijoiden kesken. Käytettävyys on yhä syvällisemmin yhteydessä sovellusarkkitehtuurien perusratkaisuihin, monikanavaisten arkkitehtuurien puutteet näkyvät yhä selkeämmin huonona käytettävyytenä.

Aiheluokat (ACM Computing Reviews 1998): C.5.3, D.2.2, D.2.11

Avainsanat: käyttöliittymät, mobiililaitteet, XML, arkkitehtuurit

Sisältö

1 Johdanto.....	1
2 Universaali käytettävyys.....	1
2.1 Käytettävyyslausunto	3
2.2 Teknologinen varianssi	3
3 Päätelaiteriippumattomat kielet sovellusten kuvaamiseen	6
3.1 XML.....	6
3.2 XML-pohjaiset käyttöliittymäkielet	7
4 Muutostarpeet sovellusarkkitehtuureissa	8
4.1 Käyttöliittymien kuvaus interaktioidina	8
4.2 Vaatimus kokonaisvaltaiselle käyttöliittymäsuunnittelulle.....	9
5 Kaksi esimerkkiratkaisua	9
5.1 FirstSearch-arkkitehtuuri.....	10
5.2 Aurora-arkkitehtuuri.....	10
7 Yhteenveto.....	11
Viitteet	12

1 Johdanto

Internet on tärkeä kanava sisällön ja palveluiden välittämiseksi eri tahoille. Useimmat yritykset käyttävät palveluidensa tarjoamiseen yksikanavaisia mekanismeja, joiden avulla palveluita tarjotaan käytettäväksi esim. www-selainten avulla. Nykyään, mobiililaitteiden lisääntyessä ja työtehtävien ollessa yhä vähemmän paikkaan sidottuja, on olennaista päästä käyttämään internet-palveluita erilaisten fyysisten laitteiden avulla, palveluiden halutaan olevan monikanavaisia. Nopean teknisen kehityksen vuoksi erityyppisiä ja kokoisia laitteita syntyy nopeassa tahdissa ja kaikki eivät noudata olemassaolevia standardeja.

Monikanavaisten internet-sovellusten toteuttaminen on vaikeaa ja työlästä. Erilaiset päätelaitteet vaativat tukea mm. erilaisille verkkoprotokollille, skriptikielille, ja eri kokoisille näytöille. Tällaisten teknisten asioiden toteuttaminen jokaiselle päätelaitteelle johtaa helposti ohjelmakoodin määrän moninkertaistumiseen ja käytännössä ongelmiin ohjelmakoodin ylläpidossa sekä sovellusarkkitehtuurin toimintavarmuudessa. Tulevaisuudessa vaaditaan standardeja mekanismeja monikanavaisten sovellusten kuvaamiseksi ja toteuttamiseksi. Kustannus- ja ylläpidettävyyssyistä ohjelmistotekniset ratkaisut tulisi alusta lähtien suunnitella tukemaan universaalia käytettävyyttä.

Universaali käytettävyys on parisen vuotta vanha tutkimusalue, jonka tavoitteena on internet-palveluiden tuominen kaikkien ulottuville. Tutkimus jakautuu kolmeen osa-alueeseen, joista tässä kirjoittelussa paneudutaan erityisesti selailuun käytettävien laitteiden tekniseen varianssiin. Muut osa-alueet käsittelevät käyttäjien erilaisia koulutuksellisia ja kulttuurillisia taustoja.

2 Universaali käytettävyys

Universaali käytettävyys pyrkii mahdollistamaan informaatio- ja kommunikaatiopalvelut mahdollisimman laajalle käyttäjäskaalalle [1]. Jotta myös kokemattomat käyttäjät kykenisivät palveluita käytännössä käyttämään, tähtäimeksi ei riitä pelkästään palveluiden saat-

taminen käyttäjien ulottuville teknisessä mielessä, vaikkakin tämä on oleellinen lähtökohta. Universaalien käytettävyyden on otettava huomioon myös käyttäjien erilaiset kulturelliset taustat, sekä eroavaisuudet tiedoissa ja taidoissa. Pääasiassa kaksi tekijää: ihmisten erilaiset toimintakyvyttömyydet, sekä mobiilit laitteet, ovat johtaneet viimeaikaiseen edistykseen universaalien käytettävyyden tutkimuksessa [2].

Universaali käytettävyys voidaan tutkimusalueena jakaa kolmeen osaan, jotka omalta osaltaan pyrkivät ratkaisemaan tutkimusalueen ongelmia [3]:

Teknologinen varianssi - Pyrkii takaamaan kaikille pääsyn kaikkiin www-pohjaisiin palveluihin riippumatta selailuun käytetyn päätelaitteen teknisistä ominaisuuksista.

Käyttäjien erilaisuus - Pyrkii mahdollistamaan palveluiden käytön erilaisille ihmisille riippumatta esimerkiksi käyttäjän tiedoista ja taidoista, iästä, sukupuolesta, kyvyttömyyksistä tai kulttuurista.

Aukot käyttäjien tietämyksessä - Pyrkii rakentamaan sillan käyttäjän tämän hetkisen tiedon ja tehtävien suorittamiseen vaadittavan tietotilan välille.

Esimerkiksi erilaisten kansalaispalveluiden kokonaisvaltainen siirtäminen verkossa hallittavaksi edellyttää universaalien käytettävyyttä. Tällaisia palveluita ovat esimerkiksi verotukseen, sekä eläkkeisiin liittyvät palvelut. Ihmiset käyttävät tällaisia palveluita erilaisista lähtökohdista. Erottavina tekijöinä ovat mm. käyttäjien taloudellinen tilanne, koulutustausta, asuinpaikka, sekä osin myös kulturelliset seikat (esim. maahanmuuttajien osalta). Universaalien käytettävyyden voidaan katsoa toteutuvan, kun 90 prosenttia potentiaalisesta käyttäjäryhmästä pystyy käyttämään palveluita riippumatta selailuun käytettävästä laitteesta ja kulttuuri- tai koulutustaustasta [1].

2.1 Käytettävyyslausunto

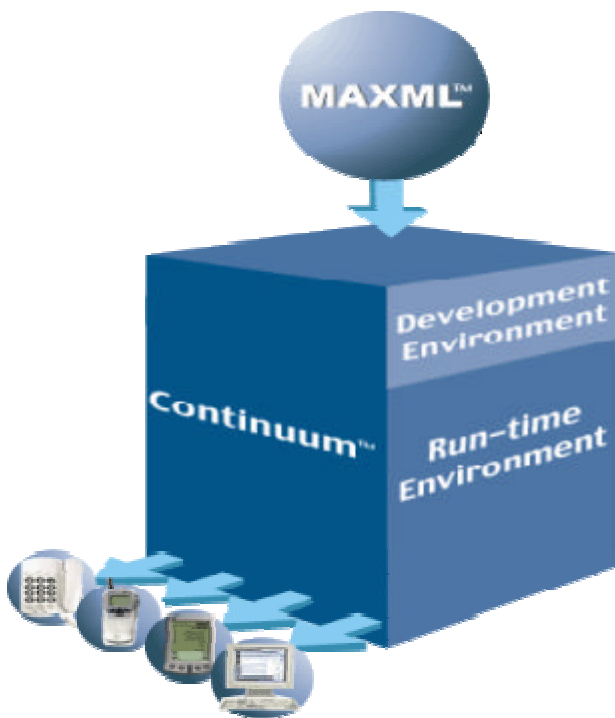
Schneiderman ja Hochheiser [1] ovat kehittäneet mallin käytettävyyslausunnolle (usability statement), jonka internet-palveluiden tarjoajat voivat liittää www-sivustolleen. Lausunnon tarkoituksena on informoida käyttäjää siitä, millä tasolla sivuston kokonaisvaltainen käytettävyys on, sekä myös motivoida kehittäjiä ja yrityksiä edistämään palveluidensa käytettävyyttä. Lausunnossa voidaan eritellä sivuston asettamat vaatimukset mm.

- käytettäville selaimille (esim. HTML-versio, JavaScript-tuki)
- näytön koolle (esim. vähintään 800x600), audio/video-ohjelmille, sekä
- käyttäjän kielitaidolle, kyvykkyydelle (esim. näkökyky) ja käyttökokemukselle.

Käytettävyyslausunto pyrkii kertomaan sivuston käytettävyydestä kaikkien universaalien käytettävyyden kolmen eri osa-alueen näkökulmista. Mallikkaasti toteutettujen sivustojen käytettävyyslausunnot voivat nostattaa tietoisuutta yleisistä käytettävyysaspekteista ja täten aiheuttaa positiivisia paineita yleiseen käytettävyyden kehittämiseen. Seuraavassa kappaleessa perehdytään päätelaitteiden teknisen varianssin aiheuttamiin ongelmiin, muut universaalien käytettävyyden tutkimuskohteet jätetään tämän kirjoitelman ulkopuolelle.

2.2 Teknologinen varianssi

Tyypillisiä internet-selailuun tarkoitettuja laitteita ovat esim. erilaiset PDA-laitteet, WAP-puhelimet, sekä WWW-selaimet. On selvää, että saman palvelun toteuttaminen esimerkiksi WAP-puhelimelle ja tavalliselle www-selaimelle, vaatii hyvää arkkitehtuurisuunnittelua, jotta palvelut saadaan toimimaan joustavasti molempien päätelaitetyyppien kannalta. Kuva 1 esittää mallin, jossa geneerinen sovelluskuvaus muunnetaan sovel-lusarkkitehtuurissa dynaamisesti eri päätelaitetyypeille sopivaan muotoon.



Kuva 1: Monikanavaisissa sovellusarkkitehtuureissa palveluiden geneerinen kuvaus konvertoidaan dynaamisesti eri päätelaitteelle sopivaan formaattiin. Kuvan MAXML-pohjainen kuvaus [4] voidaan muuntaa useille eri päätelaitteille sopivaksi käyttöliittymäksi.

Seuraavissa kappaleissa kuvataan olennaisimpia eroavaisuuksia laitteiden välillä.

2.2.1 Näytön koko

Erialaisten päätelaitteiden näyttöjen koot voidaan kategorisoida pikselikokojen mukaan esimerkiksi seuraavasti [5]

- | | |
|---------------------|----------|
| - Tietokoneet | 1024x768 |
| - Kämmentietokoneet | 256x364 |
| - Puhelimet | 48x48 |

Tarkat resoluutiot vaihtelevat toki laite- ja valmistajakohtaisesti, ylläoleva taulukko antaa kuitenkin kuvaa suuruusluokista ja koon vaihtelusta. Sinänsä saman palvelun tarjoaminen vaatii huomattavia ponnistuksia visuaalisuuden, sisällön koon ja käyttäjän interaktioiden muokkaamiseksi eri näyttökokoihin sopivaksi.

Kun sisältöä tuotetaan pienelle näytölle, sisällön koko muodostuu erityisen merkittäväksi. Palveluiden suunnittelijoiden tulee noudattaa periaatteita, joilla sisällölle saavutetaan mahdollisimman hyvä käytettävyys. On esimerkiksi olennaista, että esim. PDA-laitteella palvelua käyttävä henkilö pääsee suoraan käsiksi itse palvelun sisältöön, eikä hän turhaan joudu kahlaamaan erilaisten valikkojen läpi päästäkseen käyttämään palvelua. Sisällön joukosta tulisi karsia suuret kuvat tai käyttää pienempiä kuvia, kehyksien (frames), sekä useiden vierekkäisten sarakkeiden käyttöä tulisi välttää ja navigointi tulisi pitää mahdollisimman selkeänä ja yksinkertaisena [6].

2.2.2 Yhteysnopeuksien vaihtelu

Mikäli käyttäjä käyttää palvelua hitaan yhteysnopeuden kautta, on myös tässä tilanteessa kuvien käytön rajaamisella suuri merkitys. Kuvat tulee joko karsia tai niiden kokoa tulee pienentää. Myös multimedian käyttömahdollisuudet tulevat rajallisiksi yhteysnopeuden ollessa heikko. Osa ns. hitaista päätelaitteista pystyy käyttämään palveluita ainoastaan tekstipohjaisesti. [1]

2.2.3 Ohjelmistojen väliset erot

Kaikki päätelaitteet eivät noudata standardeja. Toisaalta jotkut selaimet ovat lähtökohtaisesti erilaisia (esim. www- ja wap-selaimet). Niinpä tarjottava palvelu joudutaan usein muuntamaan eri merkkauksielille ja sovellusarkkitehtuurin täytyy kyetä kommunikoi-
maan erilaisia verkkoprotokollia käyttävien laitteiden kanssa.

Jatkuva teknologinen kehitys aiheuttaa ongelmia yhteisten standardien soveltamiselle, siksi etenkin jatkossa sovellusarkkitehtuureilta vaaditaan kykyä toimia heterogeenisten päätelaitteiden kanssa. Tällaista monikanavaisuutta voidaan lähestyä ainakin kahdella eri tavalla. Toisaalta voidaan laatia sovellusarkkitehtuureja, jotka tukevat monikanavaisten sovellusten rakentamista. Toisaalta olemassa olevia palveluita voidaan käyttää väliohjelmiston kautta, joka muuntaa palvelut erilaisille laitteille sopiviksi (ks. luku 5).

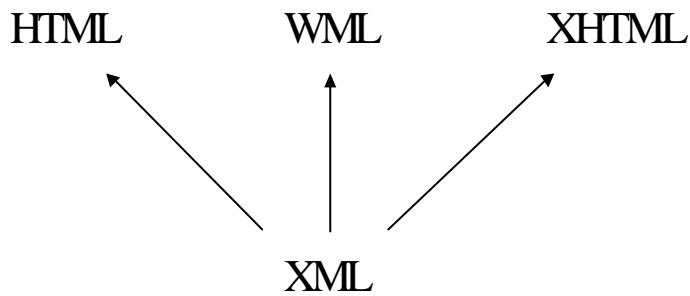
3 Päätelaite-riippumattomat kielet sovellusten kuvaamiseen

Kun tietokoneet olivat vielä kehittymättömiä, kaikki sovellukset kirjoitettiin binäärimuodossa. Ajan kuluessa laadittiin havaittujen ongelmien pohjalta kieliä, jotka nostivat ohjelmoinnin abstraktiotasoa, näin mahdollistettiin tietokoneiden käyttö eri tutkimusalueilla, ohjelmoijat saattoivat olla myös muiden alojen ammattilaisia. Myöhemmin käyttöliittymät muuttuivat graafisiksi ja sovelluksia päästiin rakentamaan sovelluskehittäjien avulla.

Samanlainen kehitys on nähtävillä myös yleensä ottaen käyttöliittymäsuunnittelussa. Tyypillinen teknisen varianssin skenaario on, että samaa palvelua voidaan käyttää esim. WAP-puhelimella, PDA-laitteella, sekä tavanomaisella WWW-selaimella. Tämänkaltaisen monikanavaisuuden tuki vaatii abstraktiotason nostamista alustakohtaisista käyttöliittymäkuvauksista. On tutkittu mahdollisuuksia mallintaa käyttöliittymiä käyttäjien semanttisesti merkittävien interaktioiden pohjalta [7]. Interaktioihin perustuva, tai muuten päätelaite-riippumaton käyttöliittymäkuvauksia voidaan muuntaa päätelaitekohtaiseksi käyttöliittymäkuvaukseksi erillisten tyylitiedostojen tai muunlaisten vastaavien kuvausmekanismien perusteella.

3.1 XML

XML (Extensible Markup Language) määrittää internetissä esiintyvien rakenteellisten dokumenttien ja tiedon yleispäteväksi formaatiksi [8]. XML:n avulla voidaan muodostaa itsekuvaavia, rakenteisia, sekä tekstimuotoisia dokumentteja. XML-dokumenttien käsittely on alustariippumatonta [9]. XML-dokumenttien tekstimuotoinen tieto voidaan eri tavoin toimivissa järjestelmissä helposti muuntaa tarkoituksenmukaiseen sisäiseen esitysmuotoon, tässä tapauksessa päätelaitekohtaisiksi käyttöliittymäkuvauksiksi. Käyttöliittymien kuvaukseen on olemassa useita XML-pohjaisia kieliä (ks. seuraava luku). Kielten kehityskaari on kuitenkin vielä nuori ja ongelmia esiintyy. Kuvassa 2 XML-pohjaisesta käyttöliittymäkuvauksesta muodostetaan sovellusprosessin avulla käyttäjän selaimelle sopiva merkkauskielinen esitys.



Kuva 2: XML-pohjainen käyttöliittymäkuvaus voidaan muuntaa eri merkkaukielille.

3.2 XML-pohjaiset käyttöliittymäkielet

3.2.1 XUL

XUL [10] on kieli käyttöliittymien kuvaamiseksi. Sen avulla voidaan kuvata useimpia moderneissa käyttöliittymissä tarvittavia elementtejä, kuten esim. painonappuloita, toolbar-komponentteja ja popup-valikoita. XUL ei itsessään ota kantaa käyttöliittymätapahtumiin (event) liittyvien toimintojen hallintaan. XUL on alunperin tehty Mozilla-selaimen käyttöliittymäkehityksen helpottamiseksi ja nopeuttamiseksi.

3.2.2 UIML

UIML:n [11] avulla voidaan käyttöliittymäkontrollien sijainti ja suunnittelu. Sen avulla voidaan myös määrittää toiminnallisuus käyttöliittymätapahtumien yhteydessä. Ohjelmamoduulia, jonka tehtävänä on konvertoida UIML-kuvaus päätelaitteelle sopivaan muotoon, ei kuitenkaan ole standardoitu. Näin UIML:n toiminnallisuus on käytännössä aina riippuvainen konversiomodulin toteutuksesta [7].

3.2.3 MAXML

Multi-Channel Access XML (MAXML) on osa kaupallista Continuum-sovellusalustatuotetta [4]. MAXML on kieli, jolla sovellusten käyttöliittymät ja toiminnallisuus kuvataan. Continuum on sovellusalusta, jonka avulla MAXML-pohjaisia sovelluksia voidaan ajaa. Continuum tukee MAXML-kuvauksen muuntamista HTML, Voice XML, XHTML, WML, sekä Palm PQA-muotoihin. Käyttäjän muodostaman palvelupyynnön yhteydessä Continuum-alusta tarkistaa käyttäjän selaintyypin ja palauttaa sisällön sopivassa muodossa.

4 Muutostarpeet sovellusarkkitehtuureissa

Monikanavaisten sovellusten kuvaaminen yhdelle kanavalle tarkoitetulla kielellä (esim. HTML) ei ole järkevää. Palvelun kuvaaminen erikseen kunkin päätelaitteen kielellä johtaa ohjelmakoodin määrän moninkertaistumiseen ja näin ongelmiin ohjelmakoodin ylläpidossa sekä sovellusarkkitehtuurin toimintavarmuudessa. Mikäli yleisiä mekanismeja ei käytetä, samat ongelmat toistuvat ohjelmistoprojektista toiseen. On tarpeen löytää sovellusten (ainakin käyttöliittymien) kuvaamiselle yleinen malli, joka voidaan konvertoida päätelaitteille erillisten päätelaiteriippuvaisten ohjeiden mukaisesti. Tällaisen järjestelmän geneerisen kuvauksen avulla pyritään pitämään järjestelmän kuvaamiseen tarvittavan ohjelmakoodin määrä kurissa ja näin vähentämään sovelluskehityksen riskejä.

4.1 Käyttöliittymien kuvaus interaktioina

On nähtävissä suuntaus, jossa Internet-palveluiden kuvaus on siirtymässä perinteisestä näyttöjen kuvauksesta kohti käyttäjän interaktioiden kuvausta [4, 7]. Sen sijaan, että käyttöliittymäkuvauksessa kuvattaisiin konkreettisia käyttöliittymäelementtejä, voidaankin määrittellä käyttäjän interaktioita, sekä abstrakteja interaktioon liittyviä tietotyyppisiä. Nämä abstraktit interaktiokuvaukset muodostavat sovelluslogiikassa uudelleen käytettävän osion, joka voidaan erikseen määriteltävien laitekuvausten, sekä mahdollisten tyyli-tiedostojen avulla muuntaa käytettävälle selaimelle soveltuvaan muotoon [7].

Esimerkki; XML-pohjainen kuvaus abstraktista syötekentästä [7]:

```
<!ELEMENT input_1-n (#PCDATA)>
<!ATTLIST input_1-n
    range_min CDATA #IMPLIED
    range_max CDATA #IMPLIED
>
```

Ylläoleva abstrakti syötekenttä voitaisiin tarpeen tullen muuntaa vaikka HTML-kielen SELECT-elementiksi.

4.2 Vaatimus kokonaisvaltaiselle käyttöliittymäsuunnittelulle

Tähän asti sovellusratkaisujen suunnittelussa on sekä tutkimuksessa, että yritysmaailmassa tyypillisesti noudatettu mallia, jossa käyttöliittymäsuunnittelijat suunnittelevat käytettävyyssratkaisut, jotka ohjelmoijien tulisi toteuttaa yleensä siinä vaiheessa, kun itse sovellusarkkitehtuuri on jo suunniteltu [12]. Tällainen toimintamalli johtaa usein järjestelmien suorituskykyongelmiin. Tyypillisesti käytettävyyss- ja arkkitehtuurisuunnittelijat eivät tee työtään yhteistoiminnassa ja tämä aiheuttaa myös molemminpuolisia uskottavuusongelmia.

Käyttöliittymäsuunnittelijat voivat olla koulutustaustaltaan psykologeja, kognitiotieteilijöitä, tai vaikkapa graafikoita. Monikanavaisuuden yleistyessä käyttöliittymien laadintaan vaadittavan teknisen tietämyksen tarve kasvaa jatkuvasti. On oleellista tähdätä yleiseen mekanismiin käyttöliittymien kuvauksessa, jotta eri ihmisryhmät voivat suunnitella käyttöliittymiä ilman liiallisia tekniseen tietämykseen liittyviä vaatimuksia.

Sovellusarkkitehtuurien monimutkaistuesssa on palveluiden vasteaikojen ja siten sovellusten käytettävyyden kannalta [13] hyvin olennaista, että käytettävyyssiantuntijat ja sovellusarkkitehtuurisuunnittelijat ovat alusta lähtien yhdessä pohtimassa käyttöliittymäpiirteiden toteutusta. Faulkner ja Culwin [12] ehdottavat käytettävyyssiantuntijoiden jakamista kahteen ammattiryhmään: käytettävyyssinsinööreihin, sekä käytettävyyden arvioijiin. Ideana on, että käytettävyyssinsinöörit olisivat teknisessä mielessä vastuussa siitä, että sovellusarkkitehtuurit mahdollistavat käytettävyyden toteuttamisen sovellusarkkitehtuuritasolla. Käytettävyyssarvioijat voisivat laatia käyttöliittymiä olemassaolevien määritysten rajoissa, sekä evaluoida olemassaolevia käyttöliittymiä.

5 Kaksi esimerkkiratkaisua

Tässä luvussa esitellään kaksi toteutettua arkkitehtuurimallia, jotka ratkovat universaalien käytettävyyden tekniseen varianssiin liittyviä ongelmia, kumpikin eri lähtökohdista.

5.1 FirstSearch-arkkitehtuuri

FirstSearch on järjestelmä, joka mahdollistaa kirjallisuushakujen tekemisen yli kahdeksaankymmeneen tietokantaan. Tämä www-pohjainen järjestelmä suunniteltiin mukautumaan odottamattomiin ja muuttuviin käyttäjätarpeisiin, erilaisille alustoille, eri kielille ja muuttuviin yleisiin vaatimuksiin [14]. Oleellista suunniteltaessa oli mahdollistaa järjestelmän jatkuva laajentuminen tukemaan erilaisia selaintyyppisiä ilman riskejä järjestelmän toimintavarmuuden heikkenemisestä.

Alunperin järjestelmä suunniteltiin tukemaan selaimia, joissa

- joko on, tai ei ole Javascript-tukea
- näytön koko vaihtelee (pieni, keskiuuri, suuri)
- ei ole graafisia toimintoja (esim. Lynx)

Kehittäjien mukaan kaikkein ongelmallisinta projektin läpiviennissä sen lisäksi, että vaurduttiin myös odottamattomiin teknisiin muutostarpeisiin, oli se, kuinka saada kehittäjät toteuttamaan kehitystyötä geneerisesti. Tässä tapauksessa geneerisyydellä tarkoitetaan sitoutumattomuutta tiettyyn kieleen, skriptikieleen, tai selaintyyppiin. Käyttöliittymiä suunniteltaessa oli olennaista ymmärtää toimintamahdollisuuksien erilaisuus erilaisten selaintyyppien välillä. Koko projektiryhmän tulikin siis ymmärtää, mitä asioita saadaan kirjoittaa sellaisenaan ohjelmakoodiin ja mitkä asiat tulee ratkaista ja toteuttaa ajonaikaisesti päätelaitteesta riippuen. Käytettävyyden toteuttaminen tällä tasolla vaatii siis paitsi suunnitteluyhteistyötä teknisten suunnittelijoiden ja käytettävyyssuunnittelijoiden välillä, myös tarkkaa yhteistyötä koko projektiryhmän kesken.

5.2 Aurora-arkkitehtuuri

Aurora-arkkitehtuuri lähestyy universaalien käytettävyyden ongelmaa eri näkökulmasta, kun aiemmin esitetyt mallit. Auroran perusajatus on muuntaa olemassaolevaa www-sisältöä eri selaintyypeille sopivaan muotoon semanttisin perustein [15]. Aurora voisi toimia esim. näkövammaisten portaalina erilaisiin www-kauppapaikkoihin. Aurora tarjoaisi linkin jokaiseen kauppapalveluun ja konvertoisi palveluiden sisällön yksinkertaiseen esitysmuotoon, jota voitaisiin helposti käyttää äänipohjaisella selaimella. Yleensä ottaen

käyttäjät siis pyytävätkin haluamansa palvelut Auroran kautta, jolloin käyttäjän selaintyyppistä riippuen Aurora konvertoi palvelun dynaamisesti käyttäjälle sopivaan formaattiin.

Sisällön muunnosprosessi mallinnetaan jokaiselle muunnettavalle sivulle erikseen laadittavan XML-kuvauksen perusteella. XML-elementtien sisältämän tiedon perusteella voidaan esim. tietty HTML-sivu muuntaa ainoastaan palvelun olennaiset piirteet sisältäväksi WML-sivuksi.

Aurora-arkkitehtuurin parhaana ominaisuutena voidaan pitää, että muunnettavia palveluja tarjoavia arkkitehtuureja ei itsessään tarvitse muuntaa. Lisäksi, koska jokaiselle sivustolle tehdään erilliset XML-pohjaiset muunnokset, on mahdollista kuvata muunnosprosessi eksaktisti, ilman geneeristen mallien potentiaalisesti pakottamia kompromissiratkaisuja.

Huonona puolena voidaan pitää, että muunnettavat palvelut ovat teknisessä mielessä riippuvaisia Aurora-arkkitehtuurista. Aina välttämättä ole haluttavaa tai mahdollista käyttää haluttua palvelua Auroran kautta. Käyttäjän ja varsinaisen palvelun välissä on aina yksi tarpeeton kerros, joka epäilemättä näkyy myös palvelun vasteajoissa.

7 Yhteenveto

Universaali käytettävyys pyrkii mahdollistamaan www-palveluiden käytön millä tahansa päätelaitteella, riippumatta käyttäjän taustatiedoista, kulttuuritaustasta ja käyttökokemuksesta. Tässä kirjoittelussa on erityisesti pohdittu monikanavaisuuden tukemisen aiheuttamia vaatimuksia www-palveluiden toteuttamiselle.

Jatkossa sovellusarkkitehtuureilta vaaditaan lähtökohtaisesti mukautuvuutta uusille ja erilaisille päätelaittekanaville. Sovelluksia kuvataan jatkossa usein päätelaiteriippumattomalla (XML-pohjaisella) kuvausmekanismilla. Tällainen yleinen kuvaus voidaan erilaisin mekanismein muuntaa päätelaittekohtaiseen formaattiin (esim. HTML- tai WML-muotoon).

Teknisten ratkaisujen tullessa yhä monimutkaisemmiksi ja päätelaiteriippumattomien sovelluskuvausten yleistyessä käyttöliittymä- ja sovellusarkkitehtuurisuunnittelijoilta vaaditaan parempaa yhteistyökykyä. Käyttöliittymien kuvaukseen liittyvä teknisen tietämyksen tarve motivoi käyttöliittymäsuunnittelijoiden jakoa käyttöliittymäinsinööreihin ja korkeamman tason käyttöliittymäsuunnittelijoihin.

Viitteet

- [1] Schneiderman, B., Universal Usability as a Stimulus to Advanced Interface Design, Behaviour & Information Technology 20th Anniversary Issue.
- [2] Vanderheiden, G., Fundamental principles and priority setting for Universal Usability, ACM 2000.
- [3] Schneiderman, B., Pushing human-computer interaction research to empower every citizen, UNIVERSAL USABILITY. Communications of the ACM 2000.
- [4] MAXML (Multi-channel Access XML), <http://www.workshop.curiousnetworks.com/maxml.jsp>, 15.10.2001.
- [5] Eisenstein, J., Vanderdonckt J., Puerta, A., Applying Model-Based Techniques to the Development of UIs for Mobile Computers, ACM 2001.
- [6] Nokia 9210 Communicator WWW-Browser Style Guide, Nokia Mobile Phones, 2001.
- [7] Mueller, A., Mundt, T., Lindner, W., Using XML to Semi-automatically Derive User Interfaces, IEEE 2001.
- [8] Nakhimovsky, A., Myers, T., *Professional Java XML Programming with Servlets and JSP*, s. 3. www.wrox.com 30.10.2000.
- [9] Royappa, A., Implementing catalog clearinghouses with XML and XSL. Teoksessa *Proceedings of the 1999 ACM symposium on Applied computing*.
- [10] XUL (XML-based User Interface Language), Introduction to a XUL-document, <http://www.mozilla.org/xpfe/xptoolkit/xulintro.html>, 15.10.2001.
- [11] UIML (User Interface Markup Language), <http://www.uiml.org>, 15.10.2001.
- [12] Faulkner, X., Culwin, F., Enter the Usability Engineer: Integrating HCI and Software Engineering, ACM 2000.

- [13] Nielsen, J., *Designing Web Usability: the practice of simplicity*, s. 42. www.newriders.com, 2000.
- [14] Perlman, G., The FirstSearch User Interface Architecture: Universal Access for any User, in many Languages, on any Platform, ACM 2000.
- [15] Huang, A., Sundaresan, N., Aurora: A Conceptual Model for Web-Content Adaption to Support the Universal Usability of Web-based Services, ACM 2000.