

T-121.5900: KÄYTTÖLIITTYMIEN JA KÄYTETTÄVYYDEN SEMINAARI JAVAN LOOK-AND-FEEL-SUUNNITTELUOHJENUORAT ESSEE

Antti Nummiahho

12.4.2006

ABSTRAKTI

Tässä esseessä käsitellään Javan Look-and-Feel-suunnitteluohjenuoria käyttöliittymälukutaidon kannalta. Javan Look-and-Feel-suunnitteluohjenuorat ovat käyttöliittymän näkymään (värit, muodot, aseteltu, kirjasinlajit ym.) ja tuntumaan (painikkeiden, asetusnappien, valikoiden ym. toimintatapa) liittyviä ohjeistuksia, joilla pyritään yhdenmukaisuuteen erilaisten käyttöliittymien välillä. Yhdenmukaisuuden tarkoituksena on helpottaa tuotteiden opittavuutta, sillä käyttäjät voivat hyödyntää toisen vastaavaa ohjenuoraa käyttävän tuotteen käyttökokemuksiaan uuden tuotteen käytön opettelussa. Javan Look-and-Feel-suunnitteluohjenuorat viittaavat Javan käyttöliittymäkirjaston Swing:n käyttöön liittyviin suunnitteluohjenuoriin. Esimerkkejä näistä ovat näppäimistöoikotiet usein käytetyille valikoiden toiminnoille sekä kontekstisidonnaisten valikoiden käyttö. Suunnitteluohjenuoria tarkastellaan W3C:n HTML-selain-editoria Amayaa esimerkkinä käyttäen. Look-and-Feel-suunnitteluohjenuorien käytöllä voidaan olennaisesti helpottaa käyttöliittymälukutaidon oppimisprosessia, mutta toisaalta joissain tapauksissa jopa haitata käyttöliittymien tehokkuutta. Olennaista on miettiä tapauskohtaisesti, mitkä käytettävyyssominaisuudet ovat tärkeimpiä.

1. JOHDANTO

Look-and-Feel-termi viittaa käyttöliittymän näkymään ja tuntumaan. Sitä voidaan käyttää niin tietokoneohjelmistojen kuin minkä tahansa tuotteen käyttöliittymistä puhuttaessa. Tässä esseessä keskitytään kuitenkin käsittelemään asiaa tietokoneohjelmistojen näkökulmasta. Siinä yhteydessä näkymän voidaan määrittellä muodostuvan käyttöliittymän väreistä, muodoista, asetelusta, kirjasinlajeista ym. ja tuntuman puolestaan koostuvan dynaamisten elementtien, kuten painikkeiden, asetusnappien ja valikoiden toimintatavasta. [1]

Suunnitteluohjenuoria (design guidelines) käytetään kuvaamaan minkälainen näkymä ja tuntuma pitäisi tietystä yhteydessä toteuttaa. Yhtäältä ne voivat kuvata miten käyttöliittymä tulee toteuttaa, jotta se näyttää yhdenmukaiselta tietystä tietokonejärjestelmässä, kuten Windowsissa, mutta toisaalta ne voivat myös määrittää käyttöliittymän toteutustavan, jonka myötä käyttöliittymä näyttää samalta eri tietokonejärjestelmissä. [1]

Look-and-Feel-suunnitteluohjenuorat auttavat tuotteen vahvistamisessa helpottamalla tietyn yrityksen tuotteiden tunnistamista. Toisaalta ne myös helpottavat tuotteiden opittavuutta, sillä käyttäjät voivat hyödyntää toisen vastaavaa Look-and-Feel-suunnitteluohjenuoraa käyttävän tuotteen käyttökokemuksiaan uuden tuotteen käytön opettelussa. Look-and-Feel-suunnitteluohjenuoria voidaan käyttää myös tukemaan saavutettavuuden, tietoturvallisuuden ym. laadullisten tekijöiden toteutumista. Esimerkiksi saavutettavuutta voidaan tukea mm. mahdollistamalla navigointi näppäimistön avulla sekä tekemällä fonttien ja värien vaihtaminen mahdolliseksi. [2, 3]

Javan Look-and-Feel-suunnitteluohjenuorat viittaavat Javan käyttöliittymäkirjaston Swing:n käyttöön liittyviin suunnitteluohjenuoriin. Javan Look-and-Feel-suunnitteluohjenuorien kolme keskeistä tavoitetta ovat: käyttöliittymäkomponenttien yhtenäinen esitystapa ja käyttäytyminen, yhteensopivuus yleisesti hyväksi havaittujen tyylien ja toimintatapojen kanssa sekä esteettinen vetovoima, joka ei estä keskittymästä sovelluksen sisältöön. Näistä nimenomaan viimeisintä eli esteettisen vetoiman tavoitteen toteutumista on voimakkaasti kritisoitu, sillä standardeja Javan Look-and-Feel-komponentteja pidetään yleisesti ottaen rumina. [4, 5]

Miellyttävyyden merkitystä tuotteen käytettävyyden ja markkina-arvon kannalta ei kannata väheksyä [6], joten Javan Look-and-Feel-komponenttien ulkoasun muokkamiseen onkin keskitytty eri tahoilla. Sinällään ulkoasun vaihtaminen onnistuu muutamalla rivillä koodia tai kommentorivillä sovellusta käynnistettäessä. Itse vaihtoehdoisen ulkoasun luominen vain ei ole niin helppoa. Erilaisia ulkoasuja on kuitenkin saatavilla niin ilmaiseksi kuin kaupallisinakin versioina. Esimerkiksi javootoo.com-sivusto ylläpitää listaa niin ilmaisista kuin kaupallisistakin Javan Look-and-Feel-komponenteista.

Oma taustani on sellainen, että olen ohjelmoanut Javalla aktiivisesti viimeiset viisi vuotta. Olen toteuttanut joitain Swing-käyttöliittymiä, jonka myötä olen myös törmännyt käsitteeseen Javan Look-and-Feel-suunnitteluohjenuorat. Näin minulla on jotain käsitystä siitä, mitä valitsemani näkökulma pitää sisällään.

Tässä esseessä keskityn tarkastelemaan Look-and-Feel-suunnitteluohjenuoria nimenomaan käyttöliittymälukutaidon kannalta. Yleisesti ottaen lukutaidon voidaan ymmärtää koostuvan sekä kyvystä lukea ja ymmärtää

tekstiä että taidosta tuottaa sitä. Vastaavasti käyttöliittymälukutaito ymmärretään tässä yhteydessä sekä kyvyksi käyttää tuotteita omien tavoitteidensa täyttämiseksi kuin taidoksi suunnitella ja kehittää käyttöliittymiä, jotka ottavat huomioon käyttäjien tavoitteet, kyvyt, tiedot ja taidot tavalla, joka mahdollistaa käyttäjien luontaisen vuorovaikutuksen tuotteen kanssa.

Erityisesti tarkastelen tässä esseessä Javan Look-and-Feel-suunnitteluohjenuoria. Suuri osa näistä ilmenee myös W3C:n HTML-selain-editorissa Amayassa [7], vaikka sitä ei olekaan suunniteltu Javan Look-and-Feel-suunnitteluohjenuorien avulla. Näin Amaya kuitenkin toimii hyvänä esimerkkinä. Mm. näppäimistöoikotiet usein käytetyille valikoiden toiminnoille, tarvittaessa disabloidut valikoiden toiminnot, kuvien käyttö valikon kohtien edessä, kolmen pisteen käyttö valikon toiminnoissa, jotka johtavat lisätiedon kysymiseen keskusteluikkunassa (dialog box), työkaluvihjeet (tooltip) nappuloissa, erotinviivat valikoissa ja työkalupalkissa (toolbar) sekä muistikkaiden (mnemonic) käyttö valikoiden kohdille ja nappuloille ovat Javan Look-and-Feel-suunnitteluohjenuoria, joita on käytetty myös Amayassa. Toisaalta on myös Javan Look-and-Feel-suunnitteluohjenuoria, joita Amayassa ei ole noudatettu, kuten kaikkien valikoiden kohtien alkaminen isolla kirjaimella, työkalupalkin kaikkien toimintojen saataisuus myös valikoista sekä kontekstisidonnaisten valikoiden (tyypillisesti hiiren oikealla napilla aukeavien) käyttö.

2. SUUNNITTELUOHJENUORISTA YLEENSÄ

Suunnitteluohjenuorilla on pitkät perinteet tietotekniikassa ja niitä on kehitelty vuosien varrella valtavia määriä eri käyttötarkoituksiin. Esimerkiksi Look-and-Feel-suunnitteluohjenuorien käyttötarkoitus on siis nimenomaan toimia tietynlaisina tyylioppaina, jotka ohjeistavat sitä, miten tietyn käyttöliittymän näkymä ja tuntuma tulisi toteuttaa. Javan Look-and-Feel-suunnitteluohjenuorien tarkoitus on siis saada käyttöliittymä näyttämään samalta eri tietokonejärjestelmissä. Toisaalta on olemassa myös ohjenuoria, jotka pyrkivät ohjeistamaan sen miten käyttöliittymä tulee toteuttaa, jotta se näyttää yhdenmukaiselta tietyssä tietokonejärjestelmässä. Esimerkiksi Windows XP:n ohjenuorat sovelluksille [8] ja Macintoshin käyttöliittymäohjenuorat [9] ovat tunnettuja esimerkkejä tällaisista käyttöjärjestelmäkohtaisista ohjenuorista. Eri ohjenuorien välillä on tyypillisesti monia samankaltaisuuksia, mutta luonnollisesti ne myös aina joissain mielessä eroavat toisistaan. Suunnitteluohjenuorat voivat perustua psykologian oppeihin tai käytännön kokemuksiin.

Suunnitteluohjenuorien keskeinen hyöty on siinä, että ne mahdollistavat käyttäjän opittujen taitojen siirtämisen tilanteesta toiseen ja näin esimerkiksi helpottavat uusien sovellusten käytön oppimista. Suunnitteluohjenuorilla voidaan siis ajankäytön näkökulmasta lisätä tuottavuutta ja vähentää uusien asioiden opetteluun kuluvaa aikaa ja kustannuksia. Myös käyttäjän kynnys vaihtaa sovelluksesta toiseen pienenee oleellisesti. Suunnitteluohjenuorilla voidaan kaiken kaikkiaan parantaa tuotteen laa-

tua ja esteettisiä ominaisuuksia sekä auttaa tuotekuvan vahvistamisessa helpottamalla tietyn yrityksen tuotteiden tunnistamista. Suunnitteluohjenuorilla voidaan myös parantaa tuotteiden saavutettavuutta. Lähtökohtaisesti käyttäjien tiedot ja taidot sekä käytössä olevat laitteistot ja ohjelmistot voivat nimittäin erota toisistaan valtavasti ja jos tuotteen käyttäjäryhmä on laaja, täytyy tuotteen olla käytettävä hyvin erilaisille käyttäjille.

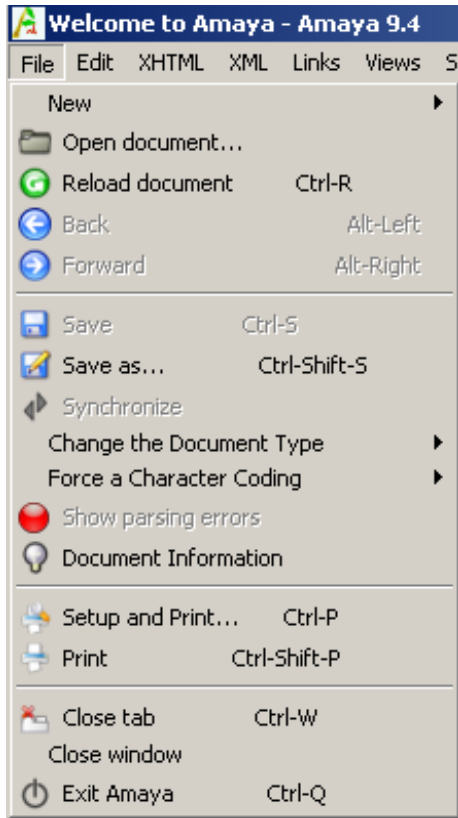
Suunnitteluohjenuorien käyttöön sisältyy myös erinäisiä ongelmia ja rajoituksia. Joskus voi olla vaikeaa valita, mitä suunnitteluohjenuoria tulisi noudattaa, jos tarjolla on useita vaihtoehtoja. Tällöin voi olla myös houkuttelevaa valita ne, mitä on helpointa noudattaa, vaikka ne eivät välttämättä olisikaan kaikkein parhaimmat tietyssä tilanteessa. Yleisesti ottaen sen päättäminen soveltuuko tietty suunnitteluohjenuora tiettyyn tilanteeseen jää useimmiten käyttöliittymän kehittäjän vastuulle, sillä suunnitteluohjenuorien määrittelyn yhteydessä asiaan ei yleensä oteta kovin kattavasti kantaa. Suunnitteluohjenuoria voi olla myös hankalaa tulkita oikein, koska ne on usein esitetty tiiviissä muodossa, jolloin ne eivät ole tarpeeksi yksikäsitteisiä ja kattavia. Eri suunnitteluohjenuorat saattavat myös olla keskenään osin ristiriitaisia.

Käyttöliittymän käytettävyyttä voidaan arvioida asiantuntija-arviointina tutkimalla kuinka hyvin käyttöliittymä noudattaa tiettyjä Look-and-Feel-suunnitteluohjenuoria. Suunnitteluohjenuorien noudattaminen ei kuitenkaan takaa hyvää käytettävyyttä, vaan käytettävyyttä on tärkeä arvioida myös käytettävyystesteillä käyttäjien kanssa.

3. ESIMERKKEJÄ JAVAN LOOK-AND-FEEL-SUUNNITTELUOHJENUORISTA

Tässä kappaleessa käsitellään joitain Javan Look-and-Feel-suunnitteluohjenuoria käyttäen esimerkkinä Windows-ympäristössä toimivaa Amayaa. Tarkoituksena on konkretisoida mitä Javan Look-and-Feel-suunnitteluohjenuorat oikein käytännössä tarkoittavat. Kuvassa 1 on esitetty Amayan File-valikko, jossa on näkyvissä useampi Javan Look-and-Feel-suunnitteluohjenuora.

Näppäimistöoikotiet usein käytetyille valikoiden toiminnoille on suunnitteluohjenuora, joka vaikuttaa ennen kaikkea järjestelmän käytön tehokkuuteen. Näppäimistöoikotiet esitetään valikoissa toimintojen nimien perässä. Järjestelmää käyttäessään käyttäjä oppii tarjolla olevat näppäimistöoikotiet ja voi ryhtyä käyttämään niitä valikosta toiminnon valitsemisen sijaan tehostaakseen toimintaansa. Näppäimistöoikoteinä on syytä käyttää yleisesti käytössä olevia näppäinyhdistelmiä, jotta eri ohjelmistoja käytettäessä käyttäjän ei aina tarvitsisi opetella tai muistella erilaisia näppäinyhdistelmiä. Yleisesti käytössä olevia näppäinyhdistelmiä ovat mm. CTRL-X leikkaustoiminnolle, CTRL-C kopioi-toiminnolle, CTRL-V liitä-toiminnolle, CTRL-Z undo-toiminnolle, CTRL-P tulosta-toiminnolle sekä CTRL-S tallenna-toiminnolle [5]. Nämä ovat käytössä myös Amayassa.



Kuva 1. Amayan File-valikko, jossa on näkyvissä useampi Javan suunnitteluohjenuora. Näppäimistöoikoteitä kuvassa ovat CTRL-R (päivitä), ALT-left (takaisin), ALT-right (eteenpäin), CTRL-S (tallenna), CTRL-SHIFT-S (tallenna nimellä), CTRL-P (asetukset ja tulosta), CTRL-SHIFT-P (tulosta), CTRL-W (sulje välilehti) ja CTRL-Q (poistu Amayasta). Järjestelmän senhetkiseen tilaan sopimattomat toiminnot on disabloitu: "Back" ja "Forward" (sivulla ei ole vielä liikuttu mihinkään suuntaan), "Save" ja "Synchronize" (sivua ei ole muokattu) sekä "Show parsing errors" (sivua ei ole parsittu). Valikon toimintojen edessä on käytetty samoja kuvia kuin työkalupalkissa (päivitä-, takaisin-, eteenpäin-, tallenna- ja tulosta-toimintojen yhteydessä). Kolmea pistettä on käytetty valikon toiminnoissa, jotka johtavat lisätiedon kysymiseen keskusteluikkunassa: "Open document...", "Save as..." ja "Setup and Print...". Valikko on ryhmitelty erotinviivoilla tiedoston avaukseen, tallentamiseen, tulostamiseen ja sulkemiseen liittyviin toimintoihin.

Toki ohjelmistossa voi käyttää myös sovelluskohtaisia näppäinyhdistelmiä toiminnoille, jotka ovat enemmän sovelluskohtaisia. Sovelluskohtaiselle toiminnolle voidaan käyttää myös yleisesti johonkin muuhun toimintoon käytössä olevaa näppäinyhdistelmää, jos on tiedossa että kyseistä yleistä toimintoa ei sovellukseen tulla koskaan toteuttamaan. Esimerkiksi Amayassa käytetään CTRL-R-yhdistelmää päivitä-toiminnolle, vaikka CTRL-R on yleisesti määritelty näppäimistöoikote oikealle tasaus -

toiminnolle.¹

Tarvittaessa disabloituilla valikoiden toiminnoilla parannetaan järjestelmän käytettävyyttä, sillä niillä estetään käyttäjiä tekemästä toimintoja, jotka eivät sovellu järjestelmän senhetkiseen tilaan. Disabloidut valikoiden toiminnot esitetään himmennettyinä ja käyttäjä ei voi valita niitä. Esimerkiksi Amayassa undo- ja redo-toiminnot ovat aluksi molemmat disabloituina. Kun käyttäjä tekee jotain, undo-toiminto tulee aktiiviseksi. Jos käyttäjä valitsee tällöin undo-toiminnon, redo-toiminto tulee aktiiviseksi ja undo-toiminto disabloituu, paitsi jos käyttäjä on tehnyt useampia peruttavissa olevia toimenpiteitä, jolloin sekä undo- että redo-toiminnot jäävät aktiiviksi. Suunnitteluohjenuoraan kuuluu myös sääntö siitä, että jos kaikki valikon toiminnot tulevat disabloituiksi, ei itse valikkoon pääsyä saa kuitenkaan estää, jotta käyttäjä voi kaikissa tapauksissa kuitenkin tarkastella valikon sisältöä. Amayassa ei tällaista tilannetta kuitenkaan pääse koskaan syntymään.

Samojen kuvien käyttö niin valikoiden kohtien edessä kuin työkalupalkin nappuloinakin auttaa käyttäjää miettimään työkalupalkin ja valikoiden vastaavat toiminnot samoiksi. Amayassa tätä on käytetty päivitä-, takaisin-, eteenpäin-, tallenna-, tulosta- ja etsi-toimintojen yhteydessä.

Kolmen pisteen (...) käyttö valikon toiminnoissa, jotka johtavat lisätiedon kysymiseen keskusteluikkunassa on analogista kolmen pisteen käyttöön kirjoitetussa tekstissä siinä mielessä, että tekstissä kolme pistettä osoittaa yhden tai useamman sanan puuttumista, jotta virke olisi täydellinen ja käyttöliittymän valikoiden toiminnoissa lisätiedon tarvetta, jotta toiminto olisi valmis suoritettavaksi. Esimerkiksi Amayassa on toiminto "Open document...", joka johtaa keskusteluikkunaan, jossa kysytään tarkemmin mikä dokumentti halutaan avata ja toiminto "Find...", joka johtaa keskusteluikkunaan, jossa kysytään tarkemmin mitä aiotaan hakea. Kolmea pistettä ei kuitenkaan ole tarpeen käyttää aina, kun toiminto johtaa keskusteluikkunan aukeamiseen, sillä keskusteluikkuna voi olla myös toiminnon tulos itsessään. Esimerkiksi Amayassa on toiminto "Preferences", jonka nimen perässä ei ole kolmea pistettä, vaikka se avaa keskusteluikkunan, sillä keskusteluikkuna itsessään on toiminnon tulos eli haluttujen asetusten määrittäminen.

Työkaluvihjeet nappuloissa ovat lyhyitä ohjetekstejä, jotka tulevat näkyviin, kun hiiren kursorin jättää hetkeksi nappulan päälle. Työkaluvihjeet auttavat käyttäjää ymmärtämään mitä mikin toiminto tekee. Tästä on hyötyä, sillä usein toiminnallisuuden päättely pelkästä nappulan kuvasta ei ole aivan helppoa. Työkaluvihjeitä onkin hyödyllistä tarjota kaikista työkalupalkin nappuloista. Amayassa muihin paitsi takaisin-, eteenpäin- ja pysäytä-nappuloihin on liitetty työkaluvihjeet.

¹ Itse asiassa Amayassa päivitä-toiminnolle on kaksi näppäimistöoikoteitä, sekä CTRL-R että yleisesti käytössä oleva F5, joista CTRL-R on laitettu valikkoon päivitä-toiminnon nimen perään. Se on yhdellä toiminnolla tarpeen olla useampi näppäimistöoikote ja erityisesti tässä tapauksessa onko CTRL-R tarpeen, kun yleisesti tunnettu näppäimistöoikote F5 on käytössä, on sitten asia erikseen.

Erotinviivat valikoissa ja työkalupalkissa ryhmittelevät toimintoja, mikä auttaa käyttäjää hahmottamaan mitkä toiminnot ovat keskenään jossain mielessä samankaltaisia. Erotinviivat myös parantavat valikoiden luettavuutta ja halutun toiminnon löytämistä. Esimerkiksi Amayassa File-valikko on jaettu erotinviivoilla tiedoston avaukseen, tallentamiseen, tulostamiseen ja sulkemiseen liittyviin toimintoihin ja Edit-valikko toimintojen perumiseen, tekstin muokkaamiseen, tiedon etsimiseen ja asetusten muokkaamiseen liittyviin toimintoihin. Pitkien valikoiden yhteydessä on mietittävä mitkä toiminnot on järkevää jakaa erotinviivoilla ja mitkä olisi tarpeen sijoittaa alavalikoihin.

Keskusteluikkunat (dialog boxes) voivat olla modaalisia tai modaalittomia. Modaaliset keskusteluikkunat esittävät käyttäjää käyttämästä muita sovelluksen toimintoja niin kauan kuin modaalinen keskusteluikkuna on auki, kun taas modaalittomat keskusteluikkunat mahdollistavat sovelluksen muiden toimintojen ja ikkunoiden käytön modaalittoman keskusteluikkunan ollessa auki. Sovelluksessa pitäisi pyrkiä käyttämään modaalittomia keskustelukunoita aina kun se on mahdollista. Näin käyttäjillä on mahdollisuus tehdä asioita haluamassaan järjestyksessä ja tarvittaessa tarkastaa asioita muualta sovelluksesta tarvitsematta jatkuvasti sulkea ja avata keskusteluikkunaa. Modaalisia keskustelukunoita tulisi käyttää vain tilanteissa, joissa sovelluksen muut toiminnot eivät voi edetä keskusteluikkunan aukiollessa. Esimerkiksi sovelluksen tietojen lataamisen etenemistä kuvaava keskusteluikkuna voisi olla modaalinen, jos sen aukiollessa ei ole järkevää käyttää mitään sovelluksen toimintoja. Esimerkkejä tyypillisistä modaalittomista keskustelukunoista ovat hakutoiminnon keskusteluikkuna (find) ja sovelluksen asetusten muuttamiseen käytettävä keskusteluikkuna (preferences). Nämä keskusteluikkunat ovat myös Amayassa modaalittomia.

Keskusteluikkunoiden pitää mahdollistaa navigointi hiiren lisäksi myös näppäimistöillä. Tällä voidaan parantaa niin sovelluksen saavutettavuutta kuin tehokkuuttakin. Tab-näppäimellä voidaan siirtyä eteenpäin ja Shift-Tab-näppäimillä taaksepäin kentästä toiseen loogisessa järjestyksessä ei-aktiivisten kenttien yli hyppien. Ctrl-Tab-näppäimillä voidaan vaihtaa eteenpäin ja Ctrl-Shift-Tab-näppäimillä taaksepäin välilehdestä toiseen. Enter-näppäimen painalluksella suoritetaan keskusteluikkunan oletustoiminto, kuten keskusteluikkunan sulkeminen tai hakutoiminnon suorittaminen. Jotkut käyttäjät ovat totuneet käyttämään Enter-näppäintä kenttien välillä siirtymiseen oletustoiminnon suorittamisen sijaan. Erityisesti tilanteissa, joissa keskusteluikkunan oletustoiminto on disabloitu ennen kuin keskusteluikkunan kentät on täytetty, voikin olla järkevää, että Enter-näppäintä käytetään Tab-näppäimen rinnalla keskusteluikkunan kentissä eteenpäin siirtymiseen, kunnes keskusteluikkunan oletustoiminto tulee aktiiviseksi. Navigoitaessa tab-näppäimellä tekstikenttään, tulee tekstikentän teksti maalata valituksi ja asettaa kursori tekstikentän loppuun. Näin käyttäjä voi korvata tekstikentän sisällön ryhtymällä vain kirjoittamaan uutta sisältöä, jatkaa edellistä sisältöä painamalla

ensin nuolta oikealle tai muuttaa edellistä sisältöä navigoimalla tekstikentässä nuolinäppäimillä. Amayassa keskusteluikkunoissa navigointi näppäimistöillä on toteutettu edellä kuvatulla tavalla. Enter-näppäintä käytetään ainoastaan keskusteluikkunan oletustoiminnon suorittamiseen.

Kytännappulat (toggle buttons) näyttävät muuten aivan tavallisilta nappuloilta, mutta niillä on kaksi tilaa: päällä ja pois päältä. Käyttäjä voi vaihtaa kytkinnappulan tilaa sitä klikkaamalla, jolloin kytkinnappulan kuvan tai tekstin ei tulisi muuttua, ainoastaan taustan korostua. Tyypillisiä kytkinnappuloiden käyttötarkoituksia ovat tekstin fontin (lihavoitu, kursivoitu, kiinteälevykyinen) ja tasauksen (vasemmalle, oikealle, keskitetty, reunat tasattu) valitsemiset. Näistä edellinen koskee yksittäisen kytkinnappulan käyttöä ja jälkimmäinen ryhmiteltyjen kytkinnappuloiden käyttöä. Esimerkiksi niin kauan kuin lihavoiminnin ja kursivoiminnin valitsevat kytkinnappulat ovat päällä, on pääikkunaan syötettävä teksti lihavoitua ja kursivoitua. Vastaavasti vain yksi tasaus voi olla valittuna kerrallaan. Fontin ja tasauksen asetukset ovat toimintoja, joihin kytkinnappuloita käytetään myös Amayassa. Tietyllä tapaa voidaan ajatella, että kytkinnappulat ja asetusnapit (check boxes) tarjoavat saman toiminnallisuuden. Molempia voidaan tarvittaessa käyttää useampia ja ryhmitellä mahdollistamaan valinnan tekemisen isommasta joukosta. Yleisenä nyrkkisääntönä onkin, että kytkinnappuloita tulee käyttää työkalupalkeissa ja asetusnappeja valikoissa ja keskusteluikkunoissa.

Hiiren käyttöön liittyy myös monia Javan Look-and-Feel-suunnitteluhjenuoria. Objektien valinta hiirellä klikkaamalla tapahtuu siten, että kun hiiren näppäin painetaan pohjaan jonkin objektin päällä, objekti yleensä korostuu ja kun hiiren näppäin päästetään irti, valitaan objekti tai suoritetaan sitä vastaava toiminto. Samalla aiemmin mahdollisesti valittuna olleen objektin valinta poistuu. Pitämällä Shift-näppäintä pohjassa hiirellä klikatessa voidaan jatkaa edellistä valintaa. Pitämällä Ctrl-näppäintä pohjassa hiirellä klikatessa voidaan vaihdella objektin tilaa valitun ja ei-valitun välillä muiden valintojen pysyessä ennallaan. Kaksoisklikkaamalla voidaan valita laajempia alueita, kuten kokonainen sana tekstikentässä tai valita ja suorittaa jokin objekti. Kolmoisklikkausta voidaan käyttää valitsemaan vielä laajempia alueita, kuten tekstirivi. Tätä ei käytetä Amayassa. Vetämällä (drag) voidaan valita joukko objekteja, kuten tietty osa tekstiä. Käyttäjälle voidaan antaa palautetta vaihtelemalla hiiren kursoria. Esimerkiksi tekstikentässä kursori voi olla I-muotoinen ja linkin kohdalla käsi. Kaikille hiirellä tehtävillä toiminnoille pitäisi olla vaihtoehtoinen suoritustapa näppäimistöillä.

Muistikkaiden käyttö valikoiden kohdille ja nappuloille auttaa käyttäjää navigoimaan valikoissa näppäimistön avulla. Muistikkaat näkyvät käyttöliittymässä valikoiden otsikoiden ja toimintojen nimissä alleviivattuina kirjaimina. Käyttäjä voi siirtyä tiettyyn valikkoon painamalla yhtäaikaan ALT-näppäintä ja alleviivattua valikon otsikon kirjainta. Vastaavasti käyttäjä voi valita jonkin avoimena olevan valikon toiminnon pitämällä ALT-näppäimen pohjassa ja painamalla haluamansa toiminnon nimen alleviiva-

tun kirjaimen pohjaan tai irrottamalla ALT-näppäimen ja painamalla pelkästään avoimen valikon halutun toiminnon nimen alleviivattua kirjainta. Toisin kuin näppäimistöikoteissa, kun jokin valikko on avoinna, koskee tietyn kirjaimen painallus vain avoinna olevaa valikkoa. Näin samoja kirjaimia voidaan käyttää muistikkaina eri valikoissa. Muistikkaita on hyvä käyttää kaikissa valikon komennoissa. Muistikkaita valitessa on ensisijaisesti käytettävä yleisesti käytössä olevia kirjaimia, jotta kuten näppäimistöikoteiden yhteydessä, käyttäjän ei eri ohjelmistoja käytettäessä tarvitsisi aina opetella eri muistikkaita. Esimerkiksi File-valikon yleisesti käytössä olevat muistikkaat ovat New, Open, Close, Save, Save As, Page Setup, Print ja Exit [5]. Jos toimintoa ei löydy yleisesti käytössä olevista muistikkaista, on pyrittävä käyttämään toiminnon ensimmäistä kirjainta. Jos valikossa on useampia toimintoja, jotka alkavat samalla kirjaimella, on ensisijaisesti käytettävä komennon keskeisintä konsonanttia ja toissijaisesti keskeisintä vokaalia. Esimerkiksi jos valikossa on komennot Style ja Size, voidaan muistikkaiksi valita S ja Z. Muistikkaat on luonnollisesti lokalisoitava käyttöliittymän kielen mukaan. Amayassa käytetään muistikkaita edellä kuvatulla tavalla ainoastaan sillä eroavaisuudella, että muistikkaat tulevat näkyviin vasta ALT-näppäintä painettaessa. Tämä on toisaalta hyvä asia, siinä mielessä että muistikkaat eivät hämää käyttäjiä, jotka navigoivat hiirellä, mutta toisaalta käyttäjien voi olla vaikea tajuta muistikkaiden olemassaolo, jos nämä eivät keksi kokeilla ALT-näppäimen painallusta.

Työkalupalkin kaikkien toimintojen saatavuus myös valikoista auttaa käyttäjää löytämään haluamansa toiminnon, sillä vaikka työkalupalkki on valikkoa nopeampi käyttää, siitä voi olla hankalaa löytää etsimäänsä komentoa varsinkin, jos siinä käytetään pelkkiä kuvia. Amayassa tätä sääntöä ei noudateta, sillä työkalupalkin Stop- ja Home-painikkeita ei löydy valikoista.

Kontekstisidonnaisten eli yleensä hiiren oikealla napilla aukeavien valikoiden käyttö tehostaa järjestelmän käyttöä. Se myös helpottaa juuri tiettyyn tilanteeseen sopivien komentojen löytämistä, sillä kontekstisidonnaisissa valikoissa esitetään vain siihen objektiin tai alueeseen liittyvät toiminnot, jossa hiiren kursori sijaitsee. Kontekstisidonnaisissa valikoissa ei ole otsikoita, mutta muut ne toimivat normaalien valikoiden tapaan eli niissä käytetään muistikkaita, näppäimistöikoteita ja alavalikoita, kuten muuallakin sovelluksessa. Kaikkien kontekstisidonnaisissa valikoissa olevien toimintojen pitää olla saatavilla myös tavallisista valikoista, sillä kontekstisidonnaisten valikoiden olemassaolo ei ole käyttäjälle niin ilmeistä kuin tavallisten valikoiden ainakaan, jos niitä ei käytetä systemaattisesti kaikkialla sovelluksessa. Kontekstisidonnaisissa valikoissa käytettävien muistikkaiden ja näppäimistöikoteiden pitää olla samoja kuin tavallisissa valikoissa käytettävien. Kontekstisidonnainen valikko liittyy tiettyyn objektiin, valintaan tai alueeseen seuraavalla tavalla.

- Jos jokin objekti tai valinta (esim. maalattu teksti) on valittuna ja käyttäjä avaa kontekstisidonnaisen valikon, liittyy kontekstisidonnainen valikko valit-

tuun objektiin tai valintaan.

- Jos mitään objektiä tai valintaa ei ole valittuna, mutta käyttäjä avaa kontekstisidonnaisen valikon sellaisen päällä, valitaan objekti tai valinta, jonka päällä ollaan ja liitetään kontekstisidonnainen valikko siihen.
- Jos mitään objektiä tai valintaa ei ole valittuna tai edes hiiren kursorin alla, poistetaan muualla mahdollisesti valittuna olevat objektit tai valinnat ja liitetään kontekstisidonnainen valikko alueeseen, jolla hiiren kursori on.

Amayassa ei käytetä kontekstisidonnaisia valikoita, vaikka niistä voisi olla siinä erityisesti sivuja editoidessa paljonkin hyötyä.

4. JAVAN LOOK-AND-FEEL-SUUNNITTELUOHJENUORAT KÄYTÄNNÖSSÄ

Standardeja Javan Look-and-Feel-komponentteja on kritisoitu ennen kaikkea niiden ulkoasun takia. Niiden ulkoasua pidetään toisaalta rumana ja toisaalta ei kovin uskollisena tietyn käyttöjärjestelmän yleisesti mielletylle ulkoasulle. [4]

Käyttöliittymän ulkoasu vaikuttaa ennen kaikkea käyttäjän ensivaikutelmaan ja miellyttävyyden tunteeseen järjestelmän käytön suhteen. Käyttäjät ovat alkaneet kiinnittää yhä enemmän huomiota myös tuotteiden esteettiseen puoleen, minkä vuoksi Javankin standardeille Look-and-Feel-komponenteille on kehitetty erilaisia ulkoasultaan miellyttävämpiä vaihtoehtoja. Javan Swing-käyttöliittymäkirjasto, jossa Javan Look-and-Feel-komponentteja siis käytetään, mahdollistaa tietyssä sovelluksessa käytössä olevien Look-and-Feel-komponenttien helpon määrittämisen ja vaihtamisen joko koodissa

```
UIManager.setLookAndFeel("className");
```

tai sovellusta käynnistettäessä

```
java -Dswing.defaultlaf=className MyApp
```

Kustomoiduilla Look-and-Feel-komponenteilla voidaan parantaa käyttöliittymän miellyttävyyttä monin tavoin, kuten ottamalla käyttöön antialiasoidut fontit tai miellyttävämmät väriskeemat. Look-and-Feel-komponenttien luominen ei ole aivan helppoa, mutta erilaisia ulkoasuja on kuitenkin saatavilla niin ilmaiseksi kuin kaupallisinkin versioina. Esimerkiksi javootoo.com-sivusto ylläpitää listaa niin ilmaisista kuin kaupallisistakin Javan Look-and-Feel-komponenteista. Yksi mielenkiintoisimmista on Skin Look And Feel -kirjasto, jonka avulla käyttäjä voi muunnella sovelluksen ulkoasua nykyään muodissa olevien skinien avulla suoraan käyttöliittymän kautta. Tulevaisuudessa omien Look-and-Feel-komponenttien kehittäminen saattaa helpottaa, sillä erilaisia generisiä Look-and-Feel-komponentteja, jotka mahdollistavat omien komponenttien määrittelyn esimerkiksi XML-tiedostojen avulla, ollaan kehittämässä. [4]

Standardien Javan Look-and-Feel-komponenttien uskollisuus tietyn käyttöjärjestelmän yleisesti mielletylle ulkoasulle on siis ollut toinen kritiikin kohde. Esimerkiksi on väitetty, että Windowsille sovelluksia kehitettäessä eivät Swingillä luodut käyttöliittymät näytä aidoilta Windows-sovelluksilta. Kannattaa kuitenkin miettiä, mikä oikein on Windows-sovellusten standardiulkoasu, kun jo Microsoftinkin tuotteet, kuten Office, Media Player ja Messenger, näyttävät aivan erilaisilta. Yleisesti ottaen Javan Look-and-Feel-komponenttien tämä puoli on herättänyt enemmän kritiikkiä sovellusten kehittäjien kuin käyttäjien keskuudessa. [4]

Javan käyttöliittymäkirjastoa Swingiä on myös itsessään kritisoitu ennen kaikkea hitaaksi, mitä se alunperin ilmestyessään kieltämättä olikin. Sitten Swing on kehittynyt paljonkin eivätkä sillä luotavat käyttöliittymät enää ole mitenkään erityisen hitaita, mutta vanha maine on jäänyt ja monilla on edelleen mielikuva hitaasta Swing:istä.

5. LOOK-AND-FEEL-SUUNNITTELUOHJENUORIEN MERKITYS KÄYTTÖLIITTYMÄLUKUTAIDON KANNALTA

Look-and-Feel-suunnitteluohjenuorien käytöllä voidaan olennaisesti helpottaa käyttöliittymälukutaidon oppimisprosessia. Erityisesti suunnitteluohjenuorilla voidaan helpottaa tuotteiden opittavuutta, sillä käyttäjät voivat hyödyntää toisen vastaavaa Look-and-Feel-suunnitteluohjenuoraa käyttävän tuotteen käyttökokemuksiaan uuden tuotteen käytön opettelussa. Look-and-Feel-suunnitteluohjenuorien merkitystä käyttöliittymälukutaidon helpottamisessa voidaan havainnollistaa analogialla tavalliseen kirjoitetun tekstin lukutaitoon. Lukeminen olisi olennaisesti hankalampaa, jos joka kirjassa kirjaimet hieman eroaisivat toisistaan. Yhtäläillä käyttöliittymien käyttö on hankalampaa, jos joka käyttöliittymässä käyttöliittymäkomponentit eroavat toisistaan hieman joko ulkoasun tai toimintatapojensa tai näiden molempien suhteen.

Kuitenkaan käytettävyyden ei ole pelkkää opittavuutta. Itse asiassa käytettävyyden voidaan määrittellä koostuvan viidestä osa-alueesta: opittavuus (learnability), tehokkuus (efficiency), muistettavuus (memorability), virheettömyys (error avoidance) ja miellyttävyys (subjective satisfaction) [10]. Käyttötarkoituksesta riippuen tietyt käytettävyyden osa-alueet ovat toisia tärkeämpiä yksittäisissä käyttöliittymissä. Voidaan todeta, että siinä missä Look-and-Feel-suunnitteluohjenuorien käytöllä voidaan helpottaa käyttöliittymien opittavuutta, voidaan niiden käytöllä toisaalta jopa haitata käyttöliittymien tehokkuutta. Yleiset suunnitteluohjenuorat ovat nimittäin yleisiä periaatteita, jotka eivät voi ottaa kantaa asioihin kovin yksityiskohtaisella tasolla. Niinpä yksittäisissä käyttöliittymissä voi olla mahdollista parantaa tehokkuutta toteuttamalla asioita suunnitteluohjenuorien vastaisesti. Tämä toki heikentää opittavuutta, mutta jos tehokkuus on kyseisissä käyttöliittymissä etusijalla, voi tämä olla hyväksyttävää. Tätäkin asiaa voidaan havainnollistaa analogial-

la. Yleisesti ymmärrettäviä lausahduksia voisivat olla esimerkiksi "Laittakaa jalat yhteen, kädet kiinni vartaloon ja seisokaa suorassa" tai "Kääntykää vasemman kautta ympäri, juoskaa noin kymmenen metriä taakse päin, kääntykää taas ympäri ja jääkää sille paikalle seisomaan". Kuitenkin armeijassa käytetään komentoja "Asento" ja "Taakse poistu", koska komentojen halutaan olevan tehokkaita. Samalla on toki heikennetty komentojen opittavuutta, sillä armeijaa käymättömän henkilön voisi olla vaikeaa toimia oikealla tavalla kyseiset komennot kuullessaan.

Muista käytettävyyden osa-alueista muistettavuus ja virheettömyys ovat myös sellaisia, joiden toteutumista hyvin laadituilla Look-and-Feel-suunnitteluohjenuorilla voidaan edesauttaa. Miellyttävyys taas on tämän suhteen hieman ristiriitainen osa-alue. Tuskin nimittäin voidaan määrittää sellaista käyttöliittymän ulkoasua, jonka kaikki kokisivat miellyttäväksi nyt ja tulevaisuudessa. Esimerkiksi standardien Javan Look-and-Feel-komponenttien kokeminen rumaksi ja halu muokata paremman näköisiä komponentteja on hyvä osoitus tästä. Toisaalta, lienee hyvin mahdollista, että suuri osa käyttäjistä kokee tietyn yksittäisen ratkaisun miellyttäväksi, jolloin sellaisen ratkaisun tukeminen suunnitteluohjenuorilla voi olla hyvinkin perusteltua.

Käyttöliittymälukutaidon kannalta Look-and-Feelsuunnitteluohjenuorilla vaikutetaan ennen kaikkea ulkoa oppimiseen. Jos käyttäjä esimerkiksi oppii, että CTRL-S suorittaa aina tallennustoiminnon, voi käyttäjä siirtää tämän tiedon sovelluksesta toiseen. Look-and-Feelsuunnitteluohjenuoria noudatettaessa on tärkeää toteuttaa sekä näkymä että tuntuma ohjenuorien mukaisesti. Jos ainoastaan näkymä toteutetaan ohjenuoria hyväksi käyttäen, sovellus voi kyllä näyttää käyttäjästä tutulta, mutta sen käyttäminen voi olla vielä vaikeampaa kuin täysin vieraalta näyttävän käyttöliittymän, kun tutulta näyttävät käyttöliittymäelementit toimivatkin totutusta poikkeavalta tavalla.

6. JOHTOPÄÄTÖKSET

Kehitettäessä sovelluksia kannattaa tuntee siihen liittyvät suunnitteluohjenuorat ja noudattaa niitä, sillä niiden olemassaololle on yleensä hyvät perusteet. Suunnitteluohjenuoria noudattamalla ei tarvitse keksiä pyörää uudestaan ja vielä erehtyä tekemään siitä neliönmallista, vaan voi käyttää hyväkseen hyväksi havaittuja ajatuksia ja sääntöjä. Suunnitteluohjenuoria ei kuitenkaan aina tarvitse noudattaa orjallisesti, mutta niistä ei kannata poiketa ilman hyvin perusteltua syytä.

Suunnitteluohjenuorilla voidaan korostaa tuotteen brändiä, jolla voi olla negatiivinen tai positiivinen vaikutus. Esimerkiksi MS Office -tuoteperheen tuotteilla on mielestäni sangen positiivinen brändi ainakin useimpien mielessä, mutta esim. Javalla (perus-Swing-käyttöliittymäkirjastolla toteutetuilla käyttöliittymillä) puolestaan varsin negatiivinen brändi, joka liittyy sen oletettuun hitauteen ja karuun ulkoasuun.

Opittavuus on keskeisin käytettävyyden osa-alue, jota suunnitteluohjenuorilla voidaan edesauttaa. Itsekin olen

huomannut tämän ennen kaikkea sellaisten sovellusten yhteydessä, jotka on portattu toisesta käyttöjärjestelmästä eikä käyttöliittymää ole juuri muutettu. Esimerkiksi GIMP-kuvankäsittelyohjelman, joka on portattu Linuxista Windowsiin, käytön oppiminen on varsin hankalaa, kun käyttöliittymä eroaa jossain määrin Windows-sovelluksen totutusta käyttöliittymästä.

Javan Look-and-Feel-suunnitteluohjenuorat soveltuvat erityisesti Javan Swing-sovelluksien kehitykseen, mutta voi niistä olla hyötyä muunkinlaisessa kehitystyössä. Swing:iä käytettäessä kannattaa panostaa myös käyttöliittymän miellyttävyyteen ja käyttää standardien Javan Look-and-Feel-komponenttien sijaan kustomoituja ratkaisuja, joita on jo hyvin saatavilla.

7. VIITTEET

- [1] Wikipedia, "Look and Feel," http://en.wikipedia.org/wiki/Look_and_feel, 2006.
- [2] E. Rosenzweig, "Design Guidelines for Software Products: A Common Look and Feel or a Fantasy?," <http://delivery.acm.org/10.1145/240000/234759/p21-rosenzweig.pdf?key1=2%34759&key2=4810440411&coll=GUIDE&dl=ACM&CFID=69263144&CFTOKEN=54149904>, 1996.
- [3] H. Thimbleby, "Computer Literacy and Usability Standards," <http://www.ucl.ac.uk/harold/srf/literacy.pdf>, 1993.
- [4] A. Roberts, "The Java Look-and-Feel Debate," http://www.osnews.com/story.php?news_id=10633&page=1, 2005.
- [5] Sun Microsystems, Inc., "Java Look and Feel Design Guidelines, Second Edition," <http://java.sun.com/products/jlf/ed2/book/>, 2001.
- [6] L. Brady and C. Phillips, "Aesthetics and Usability: A Look at Color and Balance," <http://www.hci.gr/files/surl/Aesthetics%20and%20Usability.doc>, 2003.
- [7] W3C, "Amaya," <http://www.w3.org/Amaya/>, 2006.
- [8] Microsoft Corporation, "Windows XP - Guidelines for Applications," <http://www.microsoft.com/whdc/Resources/windowsxp/default.aspx>, 2002.
- [9] Apple Computer Inc., *Macintosh Human Interface Guidelines*, Addison-Wesley Publishing Company, 1992.
- [10] J. Nielsen, *Usability engineering*, Academic Press Inc, 1994.