

10

Käyttöliittymä rakentaa ajatusmallia

Käyttöliittymiä ja vuorovaikutusta käsittelevässä kirjallisuudessa esiintyy keskeisesti termi ”mental model”. Suomeksi sanottuna käyttäjällä on jonkinlainen **ajatusmalli** siitä,

- mitä objekteja sovellus käsittelee, ja
- minkälaisia toimenpiteitä näille objekteille voidaan tehdä.

Tässä luvussa tarkastelemme sovelluskehittäjän keinoja auttaa käyttäjää ajatusmallin muodostamisessa.

Käyttäjän ja sovelluksen ensikohtaamisessa ajatusmalli nojautuu sovel- lusaluetta koskevaan tietoon. Kirjanpito-ohjelman käyttäjä odottaa pää- sevänsä käsittelemään tositteita ja tilejä. Tekstinkäsittelyohjelman käyttäjä tietää mm. sivuista, lauseista, kappaleista ja marginaaleista. Piirtämiso- hjelman käyttäjä tuntee esimerkiksi värit, piirustusarkit, pyyhkeumit ja kynät. Lentolippua webistä varaava käyttäjä odottaa keskustelua päivämää- ristä, lentoyhtiöistä, lentokentistä, matkakohteista, lentovuoroista ja hin- taluokista. Jne.

Käyttäjä yhdistää näihin sovellusalueen objekteihin tekemisen mahdol- lisuuksia. Esimerkiksi kirjanpidossa tilejä voidaan avata ja päättää, ja to- sitteita voidaan kirjata tileille. Piirrettäessä kynä jättää arkkiiin jäljen, joka voidaan pyyhkeumilla poistaa.

Joskus sovellus tai systeemi käsittelee objekteja, jotka eivät ole käyttäjil- le tuttuja. Tällaisessa tilanteessa voidaan ehkä lainata käsitteitä käyttäjille tutusta maailmasta. Esimerkiksi mikrotietokoneiden alkuaikoina kovale- vyty, tiedostot ja ohjelmat edustivat monille käyttäjille tuntematonta maa- ilmaa. **Työpöytä** tarjoaa käyttäjille tutummat käsitteet, jota muun muassa Macintoshin ja sittemmin myös Windowsin kehittäjät ovat hyödyntäneet suhteellisen onnistuneesti. Työpöytä on näiden systeemien keskeinen **vertauskuva** (metafora). Systeemit hyödyntävät vertauskuvina muitakin työpöytään liittyviä käsitteitä, kuten kansioita, dokumentteja sekä salkku- ja. Nämä vertauskuvat yhdessä muodostavat **käyttäjän käsitelmän** (englanninkielisessä kirjallisuudessa conceptual model). Kyseessä on siis se ajatusmalli, jonka suunnittelija yrittää välittää käyttäjän päähän.

Täsmennetään vielä: mitä eroa on käyttäjän käsitelmällä ja käyttäjän ajatusmallilla? Edellinen on käyttöliittymän *suunnittelijan* laatima käsi- temalli siitä, minkälaisia objekteja käyttäjä kohtaa. Jälkimmäinen taas *käyt- täjän* pääkoppaan syntyvä sovelluksen käyttämistä kuvaava tietorakenne.

Vertauskuvien käytössä ja vanhojen ajatusmallien lainaamisessa on omat vaaransa! Olen esimerkiksi tavannut mikrotietokoneiden alkuaikoi- na useampiakin itseoppineita tekstinkäsittelijöitä, jotka kirjoituskoneista tutulla tavalla painoivat Enteriä jokaisen rivin lopussa. Jotkut jopa sisen- sivät kappaleita sijoittamalla välilyöntejä jokaisen rivin alkuun.

Kun käyttäjä kohtaa uuden sovelluksen, hänellä on siis eväinään tietä- mystä varsinaisesta sovellusalueesta (esimerkiksi kirjanpidosta) tai verta- uskuvallisesta sovellusalueesta (esimerkiksi työpöydästä). Jos käyttäjät tie- tävät, minkälaisia objekteja sovellus käsittelee, he yleensä pystyvät ar- vaamaan ainakin perusasiat myös siitä, mitä näille objekteille voi tehdä. Esimerkiksi piirtämisohjelman käyttäjä varmaankin arvaa, että hän voi piirtää erivärisiä viivoja sekä myös värittää pintoja. Sähköpostisovelluk-

sen käyttäjä varmaankin arvaa, että ohjelmalla voi ainakin kirjoittaa ja lukea sekä lähettää ja vastaanottaa viestejä. Hotelliportaalin käyttäjä varmaankin arvaa, että sovelluksella voi selata hotelli- ja huonevaihtoehtoja sekä varata huoneita.

Voimme vertauskuvallisesti ajatella sovelluksen käytön oppimista sillan rakentamisena joen yli. Sovellusalueen tietämys on käyttäjälle kovaa maankamaraa jalkojen alla. Käyttäjän tietämys niistä asioista, jotka hän haluaa saada aikaiseksi, edustaa maankamaraa joen vastarannalla. Sovelluksen oppiminen on siis kuin sillan rakentamista sovellusalueen tietämyksen ja käyttäjän tavoitteiden välille. Silta rakennetaan sovelluksen käyttötaidoista. Käyttöliittymä täytyy suunnitella niin, että rakennusaineet ovat hyvin esillä silloin, kun niitä tarvitaan.

Johdonmukainen jaottelu auttaa navigoinnissa

Jos käyttäjällä on edes hieman aikaisempaa kokemusta tietokoneen käytöstä, hän osaa todennäköisesti lähteä etsimään tarvitsemiaan toimintoja mm. valikoista ja työkalupalkeista. Uteliaimmat käyttäjät pyrkivät aktiivisesti rakentamaan itselleen käsitystä siitä, mitä kaikkea sovelluksella voi tehdä. He pyrkivät myös muodostamaan itselleen käsitystä siitä, mistä he löytävät tarvitsemansa tiedot ja toiminnot.

Käyttöliittymäalan kirjallisuus kutsuu näitä asioiden löytämisen taitoja **navigointitaidoiksi**. Edellä PowerPoint-esimerkissä todettiin jo, että käyttäjän täytyy päästä helposti käsiksi dokumentin eri osiin. Tämän lisäksi hänen on myös löydettävä sovelluksesta tarvitsemansa tiedot ja toiminnot.

Käyttäjä muodostaa itselleen jonkinlaisen sisäisen kartan siitä, miten sovelluksen tiedot ja toiminnot on jaoteltu ikkunoihin, näkymiin ja dialogeihin. Valikot ja komentopainikkeet ovat käyttäjälle eräänlaisia tienviittoja, joiden avulla hän navigoi tarvitsemiensa tietojen ja toimintojen luo.

Käyttäjät eivät halua luottaa navigoinnissa pelkkään ulkomuistiin. He pyrkivät ymmärtämään sitä logiikkaa, jolla tiedot ja toiminnot on jaoteltu.

Mitä paremmin he ymmärtävät tämän logiikan, sitä vähemmän heidän tarvitsee rasittaa muistiaan.

Käyttäjälle ymmärrettävän ja loogisen jaottelun tekeminen on haaste, johon en osaa antaa aivan tyhjentäviä ohjeita. Kaksi ohjetta osaan kuitenkin antaa:

1. Muista käyttäjän näkökulma!
2. Hyödynnä käytettävyydestejä! Sopivasti laadituilla testeillä saa hyödyllistä tietoa siitä, miten käyttäjät tulkitsevat sovelluksen logiikkaa. (Testaamisesta enemmän luvussa 13)

Sovelluksen jaottelun käytettävyydestaamiseen ei tarvita valmista sovellusta tai edes prototyypppejä. Käyttäjien ajatusmalleista saa hyödyllistä informaatioita seuraavalla yksinkertaisella tavalla: Kirjoita toimintojen otsikot pienille korteille – yksi otsikko yhdelle kortille. Pyydä koehenkilöitä järjestämään lajittelemaan kortit pinoihin niin, että toisiinsa liittyvät otsikot tulevat samaan pinoon.

Seuraava käyttäjän näkökulmaa navigoinnissa havainnollistava esimerkki on lainattu kirjasta ”Atk-näytöt – pikaopas parempien näyttöjen suunnitteluun” (Tekijät: Benson, Hyryläinen, Saaren-Seppälä, Vuolanto; kustantaja: Kari Saaren-Seppälä Ky, 1990). Kirja esittelee tosielämästä poimittuja näyttöjä sekä niille tehtyjä parannuksia.

Ensin esimerkkejä alkuperäisistä valikoista:

1.	Tietokannan ylläpito
2.	Kyselyt
3.	Raportit
4.	Eräajot
5.	Liittymät

1. Asiakastietokanta
2. Tuotetietokanta
3. Tilastietokanta
4. Laskutustietokanta
5. Perittävät
6. Tilastotietokanta
7. kk-vaihteen tilastoajat
8. v-vaihteen tilastoajat
9. v-vaihteen nollausajat

Tässä on käyttäjän näkökulma pahan kerran hukassa! Sovellus on jaoteltu Atk-tehtävien ja tietokannan rakenteen näkökulmasta eikä käyttäjän tehtävien näkökulmasta.

Ja sitten paranneltu vaihtoehto:

1. Tilaukset
2. Laskut ja perittävät
3. Tuote- ja asiakaskyselyt
4. Myyntibudjetti- ja ennuste
5. Myynti asiakkaittain
6. Myynti tuotteittain
7. Myynti alueittain
8. Asiakastietojen ylläpito
9. Tuotetietojen ylläpito
10. Järjestelmän hoito

Tässä valikko keskustelee käyttäjän kanssa hänen tehtäviinsä liittyvillä käsitteillä. Tämän sovelluksen jaottelu on käyttäjälle paljon ymmärrettävämpi ja opittavampi.

Navigointi verkkosivustoilla

Esimerkiksi PowerPointin käyttäjä joutuu siis tekemään kahdenlaista navigointia:

1. Navigointia sovelluksessa
2. Navigointia käsiteltävässä dokumentissa.

Useimmissa välinesovelluksissa navigointi dokumentissa on melko yksinkertaista, koska sovelluksen rakenne on yksinkertainen. PowerPointin dokumentit koostuvat vain joukosta peräkkäisiä kalvoja. Wordin doku-

mentit koostuvat peräkkäisistä sivuista. Lisäksi niiden dokumentit ovat käyttäjän itsensä tekemiä, joten navigoinnissa tarvittava sisäinen kartta muodostuu samaa tahtia dokumentin laatimisen kanssa.

Webbisivuston käyttäjälle navigointi dokumentissa voi olla hyvin haasteellista. Dokumentti voi olla hyvin laaja ja rakenteeltaan monimutkainen. Hypertekstilinkit voivat olla käyttäjälle hyödyllisiä, mutta ne voivat myös hämmentää käyttäjää ja vaikeuttaa sisäisen kartan muodostamista. Esimerkiksi linkit läpi luettavaksi tarkoitettussa tekstissä saattavat aiheuttaa, että käyttäjä ei pääse tekstin loppuun, vaan eksyy linkkien sokkeloihin.

Webbisivuston suunnittelija voi auttaa käyttäjää sisäisen kartan muodostamisessa. Yleisesti käytetään jatkuvasti esillä olevia välilehtiä ja aihevalikkoja:

SATURDAY, OCTOBER 25, 2003 · Editions: [N. America](#) | [Europe](#) | [Asia](#) | [Edition Preference](#) BW =Subscribers Only

[Customer Service](#)
[Register](#)
[Subscribe to BW](#)

Recent Tech Features
[The Info Tech 100](#)
[Tech Special Reports](#)

[CNET Reviews](#)
[CNET News](#)
[Internet Security](#)
[Science & Health](#)

[Tech Columns](#)
[Tech Videos](#)

Technology

NEWS ANALYSIS

The Jury Hangs, Quattrone Waits

With a retrial looming, but not definite, the former CSFB tech banker's defense may reconsider having him take the stand again

Sponsored by **IBM**
Can you see it?

On demand business. Get there with **@business on demand™**

MARKET INFO		
FTSE 100	4344.00	+0.00
Nikkei	10335.70	+0.80
Hang Seng	11736.30	-0.80
DJIA	9582.46	-30.70

Create / Check Portfolio
Launch Popup Ticker

Stock Lookup

Enter name or ticker

Market Info sponsored by:

Käyttäjä on valinnut välilehden ”Technology”. Vasemmassa reunassa on välilehtikohtainen aihevalikko. Se on sama kaikille tähän välilehteen liittyville sivuille.

Tällaiseen matriisirakenteeseen voi järjestää varsin paljon informaatiota helposti navigoitavaan muotoon. Jatkuvasti esillä olevat aihealueet auttavat käyttäjää muodostamaan sisäisen kartan sivustosta.

Mutta missä määrin käyttäjä todella tarvitsee sisäistä karttaa? Olen esimerkiksi käyttänyt webbikirjakauppa Amazonin sivustoja kymmeniä kertoja, mutta minulla ei kuitenkaan ole mielikuvaa sivuston rakenteesta.

Kun tullen Amazonin sivuille, menen yleensä suoraan hakutoimintoihin, ja suoritan haun tekijän nimellä, kirjan nimellä tai aihealueeseen liittyvillä hakusanoilla.

Webin käyttö on usein tällä tavalla hakukonekeskeistä. Käyttäjät eivät välttämättä kaipaa aina sisäistä karttaa koko sivustosta. Suunnittelijan täytyy huomioida, että käyttäjät saattavat tulla julkisen hakukoneen kautta keskelle sivustoa. Käyttäjä on ensisijaisesti kiinnostunut etsimästään asiasta — hän ei ehkä ole kovin kiinnostunut sivuston rakenteesta ja kokonaissisällöstä. Sivun suunnittelijan kannattaa panna näkyvästi esille linkkejä ko. sivuun läheisesti liittyviin muihin otsikoihin. Sivuston rakenteen esille paneminen on vähemmän tärkeää.

Ainakin kahden lajin käyttäjät tarvitsevat käsitystä sivuston rakenteesta:

1. Jotkut käyttäjät haluavat seurata toistuvasti jotain aihealuetta. He tarvitsevat käsityksen siitä, että miten he löytävät sivustosta tämän aihealueen uudet artikkelit.
2. Jos sivusto tarjoaa käsikirjan tyyppistä informaatiota, käyttäjä tarvitsee kattavan sisäisen kartan. Esimerkiksi ohjelmointivälineen käsikirjan pitäisi auttaa käyttäjää hahmottamaan käsikirjan rakennetta ja sisältöä.

Monet verkkosivut käyttävät ns. leivänmuruvanaa (breadcrumb trail) navigionnin apuvälineenä:



Nimitys tulee sadusta ”Hannu ja Kerttu”, jossa päähenkilöt löytävät takaisin kotiin synkstä korvesta, koska he ovat jättäneet jokaiseen polunristeykseen leivänmuruja kertomaan, mistä suunnasta he ovat tulleet.

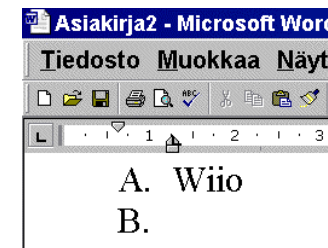
Tämä nimitys on kuitenkin harhaanjohtava! Kuten kuvasta yllä näkyy, leivänmuruvana ei kerro käyttäjän reittiä vaan se kertoo valitun sivun yläpuolella olevan aihepiirien hierarkkisen rakenteen. Esimerkiksi kuvan artikkeli kuuluu aihepiiriin ”Mideast”, joka kuuluu aihepiiriin ”World” jne.

Yllätykset vaarantavat ajatusmallin

Seuraavassa kuvassa olen kirjoittanut Microsoftin Wordilla nimeni lyhennettynä rivin alkuun:



Mitähän tapahtuu, kun painan Enter-näppäintä?



Eräs koomisten romaanien ja sarjakuvien vakiohahmoista on sellainen Avulias Aatu, joka tarkoittaa hyvää mutta sotkee enemmän kuin auttaa. Joskus Wordia käyttäessä minulle tulee tunne, että olen tekemisissä juuri tällaisen Avuliaan Aatun kanssa.

Tässä esimerkissä Word on omin lupineen päätellyt, että olen aloittamassa listaa, jossa on ensimmäisenä kohta A. Niinpä se on ystävällisesti halunnut auttaa minua aloittamalla seuraavan kappaleen listan seuraavalla

kohdalla eli B:llä. Lisäksi ohjelma on avuliaasti päätellyt, että haluan sisentää oletettuun listaan kuuluvat kappaleet.

Word tekee monissa tilanteissa vääriä oletuksia käyttäjän aikeista. Tämä on tietenkin käytettävyysongelma. Mutta tällainen pyytämätön apu voi olla haitallista silloinkin, kun Word arvaa oikein käyttäjän tarkoitukset!

Aloitteleva käyttäjä pyrkii muodostamaan itselleen ajatusmallia ohjelman ominaisuuksista ja sen käytöstä. Hän pyrkii ymmärtämään, että minkälaisia objekteja sovellus käsittelee, ja mitä niille voi tehdä.

Tekstinkäsittelijä saattaa esimerkiksi oppia, että teksti jakaantuu kappaleisiin. Hänen pitäisi sitten oppia, että kappaleita voi esimerkiksi sisentää ja niiden riviväliä voi muuttaa, sekä myös että kappaleiden välejä voi säätää. Nämä asiat hän oppii parhaiten, jos

- hänelle tarjotaan kappaleen muokkaustoiminnot havainnollisesti esille asetettuna, ja
- hän voi kokeilla näitä muokkaustoimintoja ja nähdä toimenpiteittensä *johdonmukaiset* tulokset.

Wordin Avuliaan Aatun touhut rikkovat tämän oppimiselle välttämättömän toimenpiteiden ja niiden seurausten johdonmukaisen yhteyden. Käyttäjä kohtaa yllätyksellisiä tapahtumia, joiden syyt eivät ole selkeästi näkyvillä.

Käyttäjän täytyy joka tapauksessa oppia, että minkälaisia asetuksia hän voi esimerkiksi tekstin kappaleille tehdä. Kun Word tekee kappaleille joltain omin päin, käyttäjän täytyy yrittää ymmärtää, mitä Word on tehnyt näille asetuksille. Ei riitä, että hän rakentaa ajatusmallin kappaleista ja niiden ominaisuuksista – hänen täytyy myös muodostaa ajatusmalli Avuliaan Aatun toiminnan logiikasta ja toimintatavoista:

- Minkälaisissa tilanteissa Aatu aktivoituu toimimaan?
- Mitä Aatu tekee tekstin objektien asetuksille?

Wordin Avulias Aatu vaikeuttaa siis ohjelman oppimista kahdella tavalla:

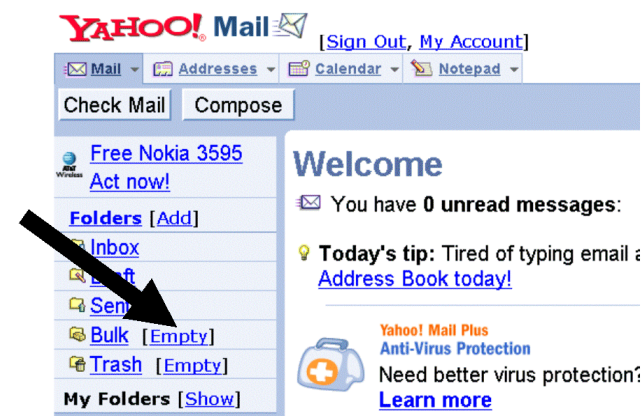
1. Se haittaa syyn ja seurauksen yhteyden näkyvyyttä.
2. Se *lisää* niiden asioiden määrää, joista käyttäjän on muodostettava itselleen ajatusmalli.

Sovellusta ei kannata yrittää tehdä käyttäjää fiksummaksi! On parempi tehdä käyttäjän oman fiksuuden soveltaminen mahdollisimman helpoksi.

Älä hämmennä käyttäjän ajatusmallia!

Käyttäjä voi kyllä ottaa nämä Wordin automaattiset muotoilutoiminnot pois päältä – jos osaa etsiä tällaista mahdollisuutta, ja jos löytää oikeat dialogit ja valinnat. Asiaa mutkistaa vielä se, että automaattisiin muotoiluihin vaikuttavia valintoja on useammassa paikassa.

Käyttäjää hämmentää, jos yhteen ja samaan asiaan liittyviä valintoja on useammassa paikassa. Toinen hyvä tapa hämmentää käyttäjää on antaa yhdellä kontrollilla useampia tehtäviä:



Tässä esimerkissä nuolen osoittama linkki ”Empty”

- tyhjentää kansion ”Bulk” sekä
- toimii hyperlinkkinä tähän kansioon.

Yhdellä kontrollilla on siis kaksi eri tehtävää. Tämä ei auta käyttäjää, joka yrittää muodostaa ajatusmallia sovelluksen toiminnasta.

Yksinkertaisiin tarpeisiin yksinkertainen käsitelmä

Tekstiviestien lähettäminen ei ole aivan vaivatonta. Alkuaikojen kännyköillä niiden näpertäminen oli ehdottomasti vaivalloista. Siitä huolimatta laaja käyttäjäkunta otti ne nopeasti omakseen. Miten tämä on mahdollista?

Tekstiviestien **ymmärrettävyysskynnys** on ilmeisesti melko alhainen, koska sovelluksen käsitelmä on yksinkertainen. Käyttäjälle riittää, että hän tuntee yhden objektityypin – tekstiviestin. On melko helppoa arvata, että näitä viestejä voi ainakin lähettää ja vastaanottaa sekä lukea ja kirjoittaa.

Kaikki kännykät pystyvät nykyisin tallentamaan tekstiviestejä. Tallennettujen viestien käsittely vaatii kuitenkin käyttäjältä uuden objektityypin – viestimistin – oppimista. Hänen täytyy oppia selaamaan tätä muistia sekä poistamaan sieltä viestejä.

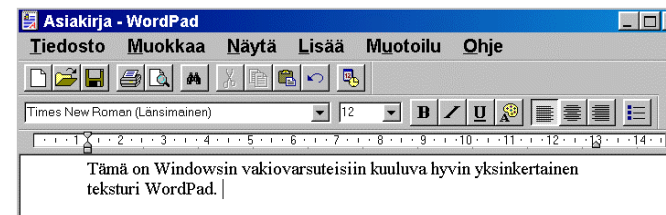
Joillain kännyköillä viestejä voi järjestellä eri kansioihin. Tämä on hyödyllinen ominaisuus, mutta se kasvattaa jälleen käyttäjälle tarpeellisten käsitelmien määrää.

Useimmilla kännyköillä voi kuitenkin lähettää ja vastaanottaa tekstareita, vaikka ei tietäisi viestien tallentamisesta ja kansioista mitään. Alkuun pääsemiseen riittää yksinkertaistettu käsitelmä, jossa ei ole mukana viestien tallentamiseen liittyviä käsitelmiä.

Kerrostetut käsitelmät

Yksinkertaisiin käyttäjän tarpeisiin pitäisi riittää yksinkertaisen käsitelmän oppiminen. Käyttöliittymät voi suunnitella niin, että ne tarjoavat aloittelevalle tai satunnaiselle käyttäjälle yksinkertaistetun käsitelmän. Alan kirjallisuudessa puhutaan **kerrostetuista käyttöliittymistä**, jotka askelittain tarjoavat käyttäjälle yhä monimutkaisempia mahdollisuuksia.

Windowsin vakiovarusteisiin kuuluu teksturi WordPad:



WordPad on varsin helppo oppia, koska sen käsitelmä on todella yksinkertainen. Siinä ei ole kerroksia eikä syvyyttä. Käyttäjä ei joudu opettelemaan mitään merkkejä, kappaleita ja marginaaleja kummempia käsitelmiä.

WordPadiin verrattuna Microsoftin Word tarjoaa paljon enemmän ominaisuuksia – ja myös monimutkaisemman käsitelmän. Tyyli (styles) ja tyylihakemistot (style sheets) ovat keskeisimpiä niistä käsitelmistä, joita WordPad ei tunne.

WordPadin teettää käyttäjällä paljon töitä, jos hän esimerkiksi haluaa, että pitkän asiakirjan kaikissa väliotsikoissa käytetään kolmentoista pisteen kokoista lihavoitua Arial-fonttia, ja otsikoiden edellä ja perässä on kuuden pisteen verran tyhjää tilaa. Käyttäjä joutuu tekemään nämä määrittelyt jokaiselle otsikolle erikseen. Wordin käyttäjä taas voi luoda väliotsikoille kappaletyylin, joka sisältää nämä otsikon muotoiluohjeet. Hän voi yhdistää kappaletyylin johonkin funktionäppäimeen. Tämän jälkeen hän voi merkitä kappaleen väliotsikoksi yksinkertaisesti painamalla tätä funktionäppäintä. Tämän seurauksena kappaleelle tehdään kaikki kappaletyylille määritellyt muotoilut.

Kappaletyylit siis säästävät vaivaa tekstin muotoilussa. Niistä on myös hyötyä, jos käyttäjä muuttaa mieltään siitä, miltä esimerkiksi välitulosikko näyttää. Hän tekee muutokset vain tyylin määrittelyyn – Word tekee sitten muutokset kaikkiin tällä tyylillä merkittyihin kappaleisiin.

Satunnaiset ja aloittelevat käyttäjät voivat aivan hyvin käyttää Wordia, vaikkeivät tiedä tyyleistä ja tyylihakemistoista mitään. Kappaleisiin voi nimittäin tehdä suoraan muutoksia, jotka tulevat tavallaan tyyliäärittelyten päälle. He voivat siis esimerkiksi suoraan muuttaa otsikon fontin li-

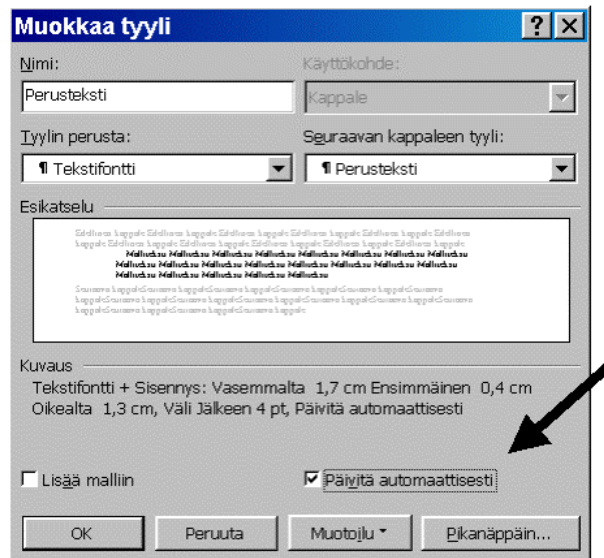
havoiduksi Arialiksi. Näin siis yksinkertaisia asioita tekevä käyttäjä pärjää yksinkertaisella käsitemallilla.

Satunnaisen Wordin käyttäjän ei tarvitse tietää, että jokaiseen kappaleeseen liittyy oletustyylimäärittely sisältävä oletustyyliobjekti. Tällaisten oletusarvojen käyttö on hyvä tapa tehdä sovelluksiin kerroksia.

Hyvän kerroksittaisen käyttöliittymän suunnittelu ei ole mikään helppo juttu. Jos tavoitteena on yksinkertainen käyttöliittymä yksinkertaisiin tarpeisiin, pitää kysyä, että *kenen* yksinkertaisiin tarpeisiin! Satunnaisten ja aloittelevienkin käyttäjien tarpeet ja prioriteetit voivat vaihdella kovasti.

On myös vaativaa suunnitella kerroksittainen käyttöliittymä niin, että monimutkaisemman käsitemallin vaikutukset eivät vuoda yksinkertaisemman mallin käyttäjän näkyviin.

Esimerkiksi Wordissa on tyylimäärittelydialogissa ruksiruutu ”Päivitä automaattisesti”.



Tämän valinnan ruksaaminen aiheuttaa asioita, jotka ainakin minä koin yllättävinä. Jos minä esimerkiksi sisennän yhtä kappaletta, joka on mer-

kitty tyylillä ”perusteksti”, Word muuttaa vastaavasti tämän tyylin sisennysmäärittymiä tyylihakemistossa. Tästä seuraa, että kaikki tyylillä ”perusteksti” merkityt kappaleet sisennetään.

Satunnainen käyttäjä saattaa esimerkiksi saada tekstipohjan, jossa tämä tyylin automaattinen muuttaminen on ruksattu päälle. Jos hän kirjoittaessaan muuttaa jonkin kappaleen ulkoasun asetuksia, nämä muutokset vaikuttavat yllättäen kaikkiin kappaleisiin! Jos käyttäjä ei tiedä mitään tyylihakemistoista, hänen on mahdotonta ymmärtää, miksi ohjelma käyttäytyy niin kuin käyttäytyy. Tässä kohdin siis tyylien vaikutukset vuotavat sellaiselle käyttäjälle, joka ei niitä tarvitse.

Lopuksi

Yksinkertaistetun käytön suunnittelussa on syytä myös muistaa Albert Einsteinin ohje oppilailleen: ”Tee asioista mahdollisimman yksinkertaisia, mutta älä yhtään sitä yksinkertaisempia”. Sovelluskehittäjä voi tulkita näitä sanoja niin, että käyttöliittymästä ei pidä tehdä käyttäjän tarpeita yksinkertaisempaa.

Yhteenveto

Käyttäjän käsitemalli on käyttöliittymän suunnittelijan laatima käsitemalli objekteista, jotka sovelluksen käyttäjä kohtaa käyttöliittymässä.

Käyttäjä taas rakentaa itselleen sovelluksesta **ajatusmallia** eli jonkinlaista tietorakennetta, joka kuvaa sovelluksen mahdollisuuksia ja käyttöä.

Sovellusalueen tietämys on se kiinteä ydin, jonka varaan käyttäjä alkaa rakentaa ajatusmallia sovelluksesta. Joskus käyttäjää voidaan auttaa lainaamalla **vertauskuviksi** eli metaforiksi käsitteitä jostain hänelle tutusta aihepiiristä.

Jos sovelluksen käyttöliittymän käsitemallissa on paljon käsitteitä, ajatusmallin muodostaminen on käyttäjälle vaativaa. Jos käyttäjän tarpeet ovat yksinkertaiset, hänelle saattaa riittää yksinkertaistettu käsitemalli.

Kerroksittainen käyttöliittymä tarjoaa yksinkertaisiin tarpeisiin yksinkertaisen käsitemallin.

