

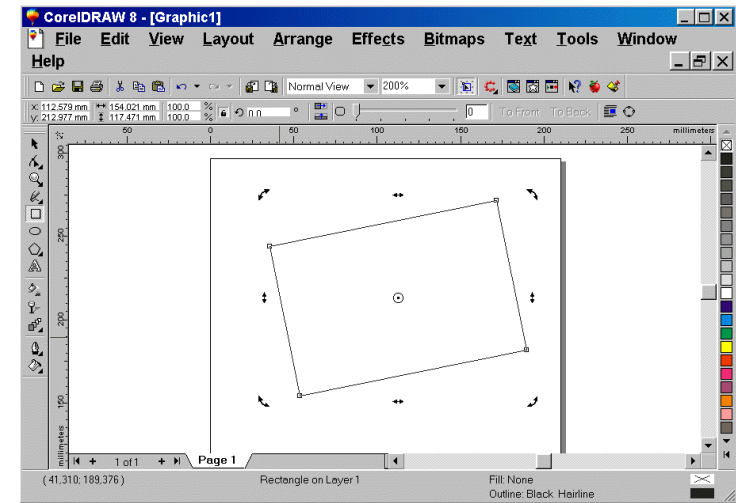
8

Käyttöliittymän ratkaisumallit

Aikaisemmissa luvuissa olemme etsineet niitä eväitä, jotka käyttöliittymän suunnittelija tarvitsee systeemisuunnittelulta. Tässä luvussa pääsemme vihdoinkin itse käyttöliittymän suunnittelun kysymyksiin!

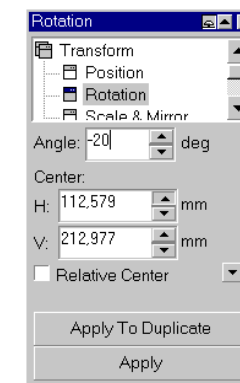
Etsimme seuraavissa luvuissa käyttäjän ja sovelluksen välisen viestinnän keskeisiä periaatteita. Tässä lyhyessä luvussa opimme muutamia peruskäsitteitä, joita tarvitsemme seuraavissa luvuissa.

Kaksi vuorovaikutusmallia



Tässä ollaan kiertämässä Corel Draw –piirtämisohjelmalla nelikulmion muotoista kuva-alkiota toiseen asentoon. Tämä tapahtuu tarttumalla hiirellä kiertokahvasta (O) ja vetämällä nelikulmio haluttuun asentoon.

Corel Draw tarjoaa myös toisen tavan tehdä sama asia. Pääva-likosta avataan muunnoksien (Transformations) dialogi:



Valitaan muunnokseksi kierto (Rotation), ja kirjoitetaan haluttu kierron määrä asteina.

Tässä on siis yhden ja saman asian tekemiseen kaksi täysin erilaista käyttämisen tapaa eli vuorovaikutusmallia. Edellistä hiirellä vetämiseen perustuvaa tapaa nimitetään **suorakäsittelyksi**. Jälkimmäistä tapaa nimitetään **keskustelukäyttöksi**.

Suorakäsittelyssä käyttäjällä on edessään olioita, joihin hän voi tarttua. Hän voi kohdistaa niihin erilaisia toimenpiteitä, joiden vaikutukset hän näkee välittömästi. Esimerkiksi tekstinkäsittelijä voi tarttua sivun marginaaliin ja leventää tai kaventaa sitä.

Suorakäsittelyssä yhdistetään usein näet mitä saat -periaate (What You See Is What You Get eli WYSIWYG). Mutta esimerkiksi Word tarjoaa WYSIWYG-tyyppisen asettelunäkymän lisäksi normaalinäkymän ja jäsennysnäkymän, jotka eivät esitä tekstin tulostusasia. Nämäkin näkymät perustuvat kuitenkin selkeästi suorakäsittelyyn. Ratkaisevaa on se, että käyttäjä käsittelee suoraan objekteja ja näkee välittömästi toimenpiteitensä tulokset.

Keskustelukäyttö tapahtuu yleensä valikoiden ja lomakkeiden avulla. Esimerkiksi monet taloushallinnon sovellukset ovat täysin keskustelukäyttöisiä.

Kuten Corel Draw'sta näimme, suorakäsittelyyn perustuvissa sovelluksissakin on keskusteluina toteutettuja osia. Toisaalta monissa keskustelukäyttöisissä sovelluksissa on joissain yksityiskohdissa hyödynnetty suorakäsittelyä. On esimerkiksi hyvin yleistä, että lomakkeella asioiden siirtäminen listasta toiseen voidaan tehdä joko komentopainikkeilla (keskusteluna) tai hiirellä vetämällä (suorakäsittelynä).

Kaksi sovellustyyppiä

Useimmat mikroissamme olevat ohjelmat ovat työvälineitä, joilla tuotamme erilaisia dokumentteja: tekstejä, piirroksia, esityksiä, laskelmia jne. Kutsumme välinesovelluksiksi tällaisia sovelluksia, joilla luodaan ja muokataan yhden käyttäjän hallinnassa olevia selviä kokonaisuuksia.

Pankkiautomaatti on esimerkki aivan toisenlaisesta sovelluksesta. Sen käyttäjä ei luo mitään kokonaisuutta, vaan hän käyttää erään hyvin laajan järjestelmän tuottamia palveluita. Kutsumme tällaisia sovelluksia palvelusovelluksiksi.

Sekä väline- että palvelusovellusten käyttäjät joutuvat pohtimaan, pitäisikö käyttöliittymän perustua suorakäsittelyyn vai keskustelukäyttöön. Voimme muodostaa näistä asioista seuraavanlaisen matriisin:

	Palvelusovellus	Välinesovellus
Suorakäsittely	Tuotannonohjaus	Tekstit Julkaisut Kuvat Laskelmat Suunnitelmat
Keskustelukäyttö	Pankkiautomaatti Kirjanpito Webbikauppa	Käyntikorttiautomaatti

Kuhunkin matriisiin ruutuun on sijoitettu esimerkkejä sinne sopivista sovelluksista.

Suorakäsittely pääsee yleensä parhaiten oikeuksiinsa välinesovelluksissa. Parhaimmillaan se tarjoaa havainnollisuutta, helppoa opittavuutta sekä tehokasta työskentelyä.

Tekstejäkin on kyllä aikoinaan tehty niin sanotuilla rivieditoreilla keskustelupohjaisesti. Minulla on pöytälaatikossa rintanappi, joka julistaa, että "Tosimiehet käyttävät EDLINiä". Hyvähän siitä on nyt vitsailla, mutta ei kukaan meistä vanhoista parroista varmaan tosissaan kaipaa tätä MS-DOSin alkuaikojen rivieditoria.

Joissain yksinkertaisissa ja satunnaisissa tehtävissä keskustelukäyttö saattaa puolustaa paikkaansa välinesovelluksissa. Esimerkiksi julkisilla paikoilla olevien käyntikorttiautomaattien käyttäjät antavat ensin lomakkeella tietonsa ja valitsevat sen jälkeen luettelosta itselleen sopivan kort-

timallin. Tässä syntyy selvä kokonaisuus - käyntikortti. Kyseessä on siis välinesovellus, vaikka sen käyttö onkin keskustelukäyttöä.

Joihinkin monimutkaisiin palvelusovelluksiin saattaa liittyä sellaisia toimintoja, joissa suorakäsittelyn havainnollisuus on paikallaan. Esimerkiksi tuotannonohjauksessa ja -suunnittelussa on ehdottomasti kyse palvelusovelluksesta. Jos tuotannon suunnittelija esimerkiksi siirtää töitä hii-
rellä vetäen työpisteiden työjonojen välillä, on kyse suorakäsittelystä.

Suorakäsittely voi tarjota monissa tilanteissa suurempaa havainnolli-
suutta kuin keskustelukäyttö. Ei kuitenkaan pidä haksahda ajattelemaan, että suorakäsittelyyn perustuva käyttöliittymä olisi automaattisesti kaikis-
sa tilanteissa parempi kuin keskustelukäyttöinen liittymä.

Esimerkiksi pankkiautomaatteihin olisi varmaankin mahdollista toteut-
taa suorakäsittelyä. Jos automaatti olisi varustettu kosketusnäytöllä, käyttäjälle voitaisiin tarjota kuvaruudulla setelipinkkoja, joista hän voisi sormella vetää seteleitä ulos annettavaksi. Mutta olisiko tällainen käyttö-
liittymä millään lailla helpompi tai tehokkaampi kuin nykyiset keskuste-
lukäyttöiset automaattit? Enpä usko, että olisi!

Yhteenveto

Käyttöliittymä voi perustua kahteen vaihtoehtoiseen vuorovaikutusmal-
liin.

Suorakäsittelyssä käyttäjä kohdistaa käyttöliittymässä esillä oleviin ob-
jekteihin toimenpiteitä, joiden vaikutukset hän näkee välittömästi. Tämä vuorovaikutusmalli tarjoaa parhaimmillaan suurta havainnollisuutta ja käytön tehokkuutta. Hyvien suorakäsittelyyn perustuvien käyttöliittymien suunnittelu ja ohjelmointi on varsin vaativaa. Suorakäsittelyn havainnol-
lisuutta hyödynnetään useimmiten välinesovelluksissa, mutta joissain palvelusovelluksissakin saattaa olla toimintoja, joissa tämä havainnolli-
suus pääsee oikeuksiinsa.

Keskustelukäyttö tapahtuu yleensä lomakkeiden ja valikkojen avulla. Tämä vuorovaikutusmalli on yleensä suorakäsittelyä helpompi toteuttaa,

mutta senkin tekeminen hyvin vaatii paneutumista ja osaamista. Palvelu-
sovelluksissa tämä on yleensä hallitseva vuorovaikutusmalli.