

Harri Laine

# **Johdatus sovellussuunnitteluun**

## **Osa 4 Käyttöliittymät**

Helsingin yliopisto

Tietojenkäsittelytieteen laitos

2000

# Sisällys

<b>6</b>	<b>KÄYTTÖLIITTYMÄT.....</b>	<b>1</b>
6.1	KÄYTTÖTAPAUSTEN PAKETOINTI.....	1
6.2	KÄYTETTÄVYYS .....	3
6.2.1	<i>Opittavuus</i> .....	3
6.2.2	<i>Tehokkuus</i> .....	8
6.2.3	<i>Muistettavuus</i> .....	11
6.2.4	<i>Virheettömyys</i> .....	12
6.2.5	<i>Tyytyväisyys</i> .....	16
6.2.6	<i>WWW-sibujen käytettävyys</i> .....	20

## 6 Käyttöliittymät

Tietojärjestelmän voi ajatella muodostuvan loogisesti kolmesta kerroksesta:

- käyttöliittymäkerroksesta
- sovelluslogiikkakerroksesta ja
- tiedon säilytyskerroksesta.

Käyttöliittymäkerros muodostaa järjestelmän ulkoisen rajapinnan; yhteydet käyttäjiin ja toisiin järjestelmiin. Ohjelmatoteutuksessa käyttöliittymäkerrokseet kuuluvat erilaiset käyttöliittymäoliot tai –komponentit kuten ikkunat, kentät, valikot, napit, www-sivut, näytöt ja raportit. Sovelluslogiikkakerros toteuttaa 'reaalimaailman simulointimallin'. Se tarjoaa sisällön järjestelmän palveluille. Tiedon säilytyskerroksen tehtävänä on huolehtia siitä, että järjestelmän tietosisältö säilyy ohjelmien suorituskertojen välillä.

Jos järjestelmä toteutetaan olioperustaisesti muodostuu kunkin kerroksen toteutus joukosta luokkia ja olioita. Sovelluslogiikkakerroksen luokkarakenteen ytimenä ovat määrittelyn yhteydessä löydettyjä reaalimaailman luokkia vastaavat luokat. Näiden lisäksi kerrokseen joudutaan yleensä lisäämään ohjelman teknisessä toteutuksessa tarvittavia apuluokkia esimerkiksi tietokokoelmien hallintaan.

Tiedon säilytystason keskeinen suunnittelukohde on järjestelmän tietokannan rakenne. Tämäkin on ytimeltään johdettavissa suoraviivaisesti määrittelyn yhteydessä aikaansaadusta luokkakaaviosta.

Käyttöliittymäkerroksen komponentit eivät ole aivan yhtä suorassa yhteydessä määrittelyn tuottamaan luokkakaavioon. Komponenttien mukaanottaminen riippuu siitä, miten tiedot esitetään ja miten käyttötapaukset teknisesti toteutetaan. Vaihtoehtoja on runsaasti.

### **6.1 Käyttötapausten paketointi**

Perustana käyttöliittymän suunnittelulle ovat järjestelmän käyttötapaukset. Suunnittelun päätavoitteena on tuottaa käyttöliittymä, jossa käyttötapaukset ovat hoidettavissa

mahdollisimman suoraan, jotta käyttäjä pystyisi saavuttamaan käyttötapaukseen liittyvät tavoitteensa ilman välivaiheita ja ylimääräisiä koukeroita. Käyttötapausten määrittelyyn liitettävät esimerkkitaipaukset muodostavat hyvän lähtökohdan käyttöliittymän suunnittelulle.

Yksi ohjelma tukee yleensä useita käyttötapauksia. Käyttötapausten paketointi samalla ohjelmalla hoidettaviksi kokonaisuuksiksi onkin lähtökohta, jonka pohjalta voidaan ryhtyä yksityiskohtaisemmin suunnittelemaan käyttöliittymän toteutusta. Tavoitteena on sellainen paketointi, jossa

- käyttäjän saa kaikki johonkin asiakokonaisuuteen liittyvät palvelunsa saman käyttöliittymän kautta (tämä ei tarkoita kuitenkaan sitä, ettei vaihtoehtoja voisi olla tarjolla)
- käyttöliittymän ei tarjoa palveluita, joita käyttäjä ei tarvitse tai joita hänellä ei ole oikeutta käyttää. Esimerkiksi opiskelijarekisteriin liittyviä hallintohenkilöstön ja opiskelijoiden käyttötapauksia ei ole syytä toteuttaa yhteisen käyttöliittymän kautta vaikka kummallakin ryhmällä olisikin yhteisiä käyttötapauksia, esimerkiksi opiskelijan osoitteen muutos. Samoin järjestelmän ylläpitäjälle tarkoitettuja käyttötapauksia ei ole yleensä syytä laittaa näkyviin normaalikäyttäjälle tarkoitettuun käyttöliittymään.

Sama tai lähes sama käyttötapaus voi olla tarjolla eri käyttöliittymien kautta. Tällöin käyttötapaus voidaan toteuttaa näissä teknisesti erilaisena. Esimerkiksi puhelimitse suoritettava tilin saldokysely on tekniseltä toteutukseltaan erilainen kuin pankki-automaatista tai web-pankkipalvelusta suoritettava saldokysely. Toisaalta on mahdollista toteuttaa usealle ohjelmalle yhteiset käyttötapaukset yhteisenä osajärjestelmänä.

Toteutuslaluista saattaa vaikuttaa siihen mitkä käyttötapaukset pakettiin otetaan, esimerkiksi puhelimen välityksellä ei ole mahdollista tai järkevää tarjota samoja palveluita kuin työasemaliittymän kautta. Toteutuslaluistaan sisältyvät laitteisto, käyttöjärjestelmä, ikkunointijärjestelmä ja mahdollinen käyttöliittymäalusta esimerkiksi WWW-selain. Alustan valinta rajoittaa mahdollisuuksia käyttöliittymän toteutukseen.

## 6.2 Käytettävyys

Järjestelmän käytettävyydellä (usability) tarkoitetaan järjestelmän sopivuutta sille tarkoitettuihin tehtäviin sille tarkoitetuilla käyttäjillä. Järjestelmä on käytettävä, jos se auttaa käyttäjää pääsemään tavoitteeseensa tehokkaasti, tuottavasti ja miellyttävästi.

Nielsenin mukaan<sup>1</sup> käytettävyyden tekijöitä ovat

- opittavuus
- tehokkuus
- muistettavuus
- virheettömyys ja
- tyydyttävyyys.

Seuraavassa tarkastellaan kutakin näistä tekijöistä sekä muutamia keinoja, joilla kyseiseen tekijään voidaan vaikuttaa. Monet keinoista eivät vaikuta vain yhteen vaan useampaan tekijöistä.

### 6.2.1 Opittavuus

Opittavuudella (learnability) tarkoitetaan järjestelmän käytön oppimisen helppoutta. Opittavuuden merkitys käytettävyystekijänä vaihtelee. WWW-pohjaisissa suurelle yleisölle tarkoitetuissa järjestelmissä se on hyvin tärkeä. Järjestelmän käyttöliittymän on tällöin oltava itsensäselittävä. Joissain tilanteissa puolestaan kaikki järjestelmän käyttäjät on mahdollista kouluttaa järjestelmän käyttöön eikä opittavuuteen tarvitse panostaa yhtä paljon.

Käyttäjät voidaan järjestelmän käytön määrän suhteen sijoittaa akselille, jonka toisessa päässä ovat aloittelijat ja toisessa kokeneet käyttäjät. Käytön ajallisen jakautumisen perusteella heidät voidaan sijoittaa akselille satunnaiskäyttäjä - säännöllinen käyttäjä. Opittavuus on tärkeä erityisesti aloittelijoille mutta myös satunnaiskäyttäjille.

---

<sup>1</sup> Nielsen J.: Usability Engineering, Academic Press, 1993

## Opittavuuteen vaikuttavia tekijöitä

### Kieli

Järjestelmän ymmärrettävyydessä keskeistä on käyttäjän oman kielen käyttö. Käyttöliittymän tulisi perustua käyttäjän tuntemiin sovellusalueen termeihin, ei atk-tekniisiin termeihin. Atk-tekniisin termein käyttäjälle voitaisiin esimerkiksi välittää suoraan tietokantaohjelmistolta saatava virheilmoitus:

*Viite-eheys vaatimusta rikottu taulun ilmoittautuminen sarakkeessa kurssinimi sen sijaan että ilmoitettaisiin*

*Järjestelmä ei tunne kurssia 'Johdatus sivellyssuunnitteluun'.*

Muita kuin sovellusalueelle ominaisia lyhenteitä tulisi välttää, sillä ne voidaan tulkita väärin. Otsakkeiden, ohjetekstien ja valintavaihtoehtojen tulisi olla kuvaavia. Samasta asiasta pitää käyttää aina samaa termiä. Yhdenmukaisuutta kaikissa muissakin muodoissa on syytä harjoittaa. Esimerkiksi seuraava valikko ei ole kielellisesti yhdenmukainen. Kaksi ensimmäistä vaihtoehtoa ovat verbin imperatiiveja, kaksi viimeistä ovat substantiiveja.

leikkaa
liimaa
kopiointi
lisäys

### Standardit ja käytännöt

Käyttöliittymien suhteen on olemassa pikemminkin käytäntöjä kuin varsinaisia standardeja sille, miten jokin asia pitäisi tehdä. Ikkunointiympäristöjen tuottajat ovat laatineet tyylioppaita, joissa suositellaan miten sovelluksia pitäisi kyseiseen ympäristöön rakentaa. Laajalle levinneet sovellukset toimivat esimerkkeinä erilaisten käyttöliittymäratkaisujen käytöstä. Jos sovellusalueelle on olemassa käyttöliittymästandardeja tai suosituksia, niitä tulisi noudattaa. Käyttäjät käyttävät yleensä useita ohjelmia ja oppivat niistä käytäntöjä. Jos toiminnon saa aikaan samoin kuin muissa

ohjelmissa, sen oppiminen on helppoa; käyttäjä osaa sen ennestään. Poikkeamat vallitsevasta käytännöstä aiheuttavat hämminkiä.

### Intuitivisuus ja houkuttelevuus

Käyttöliittymän tulisi ulkoasullaan ohjata ja houkutella käyttäjää tekemään asiat oikein. Esimerkiksi nappi, jos käyttäjä sen sellaiseksi tunnistaa, houkuttelee painamaan, liukusäädin säätämään, jne.

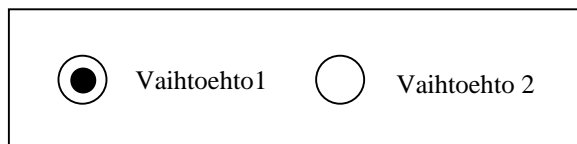
### Yksinkertaisuus ja selkeys

Käyttöliittymän tulisi olla mahdollisimman yksinkertainen. Toimenpidesarjojen tulisi olla lyhyitä ja vaiheita tavoitteen saavuttamiseen vähän. Ulkoasultaan käyttöliittymän tulisi olla siisti, jäsentynyt, ja väljä pikemmin kuin täyteen ahdettu. Komponentteja ei pidä sijoittaa satunnaisesti näytölle vaan järjestyneesti riveiksi ja sarakkeiksi. Yhteenkuuluvat asiat on syytä koota ryhmiksi. Ryhmän erottamiseen ympäristöstä voi käyttää tyhjää tilaa, värejä tai viivoja.

### Valinta kirjoittamisen asemasta

Koodien, nimien ja arvovaihtoehtojen muistaminen voi olla vaikeaa ja kirjoitettaessa syntyy kirjoitusvirheitä. Käyttäjän on helpompi tunnistaa hänelle esitetty vaihtoehto kuin muistaa miten se olisi kirjoitettava. Valinta kirjoittamisen asemasta tulisikin tarjota aina kun vaihtoehtoja on rajattu määrä.

Graafisessa käyttöliittymässä yhden vaihtoehdon valinta pienestä vaihtoehtojoukosta hoidetaan yleensä valintanappiryhmän (radio button group) avulla. Ryhmässä napin painaminen mitätöi aiemman valinnan ja aktivoi uuden.



Valintanapit

Valintaruutuja (checkbox) käytetään valittaessa nolla tai useampia vaihtoehtoja pienehköstä vaihtoehtojoukosta.

<input type="checkbox"/>	Vaihtoehto1	<input checked="" type="checkbox"/>	Vaihtoehto2
<input type="checkbox"/>	Vaihtoehto3	<input checked="" type="checkbox"/>	Vaihtoehto4

Valintanappeja ja ruutuja ei käytetä yleensä toimintojen käynnistämiseen.

Valikot (menu) mahdollistavat yhden tai useamman vaihtoehdon valinnan. Vaihtoehdot ilmaistaan yleensä tekstinä. Yhden vaihtoehdon valinta on luontevaa valikon käyttöä. Usean vaihtoehdon valinta valikosta ei ole välttämättä ilmeistä käyttäjille eivätkä käyttäjät osaa tehdä valintaa. Valikot toteutetaan usein siten että niistä näkyy vain nimi tai valittu vaihtoehto ja valikon saa esiin näpäyttämällä nimeä, siirtymällä kenttään tai avaamalla sen avausnapista. Tällainen valikko vie suljettuna vähemmän tilaa kuin nappiryhmät. Valikot voivat olla vieritettäviä ja sisältää runsaastikin vaihtoehtoja. Ison valikon käsittely on kuitenkin melko hankalaa. Esimerkiksi opintosuorituksen hyväksyjän syöttäminen hakemalla opettajan nimi 2000 opettajanimen valikosta on työlästä vaikka onkin helppo oppia.

Valikon vaihtoehtojen tulisi olla järjestyksessä jollain perusteella, esimerkiksi aakkosjärjestyksessä, yleisyysjärjestyksessä, tärkeysjärjestyksessä, yms. Valikkoa voi käyttää myös toiminnon käynnistämiseen.

Hierarkkisella valikolla tarkoitetaan valikkoa, jossa vaihtoehdon valinta avaa uuden valikon. Hierarkkisten valikoiden ongelmana on se, että käyttäjän täytyy tuntea hierarkia löytääkseen haluamansa vaihtoehdon. Vain käyttäjän sovellusalueella esiintyvien hierarkioiden voi olettaa olevan käyttäjälle tuttuja. Keinotekoiset tekniset hierarkiat vaativat aina opettelua ja niitä tulisi välttää.

Valinta voidaan tehdä myös säädinten ja kasvatettavien kenttien avulla. Myös kuvakkeita (icon) voi käyttää valintaan. Kuvakkeet ovat pieniä vaihtoehdolle annettavia kuvasymboleja. Kuvake saattaa viedä vähemmän tilaa kuin teksti.



Kuvasymbolien merkitys on aina opeteltava. Oppimista voi helpottaa liittämällä kuvakkeeseen vihjelappunen, joka ilmestyy esiin kun kursori on ollut kuvakkeen päällä jonkin aikaa.

## Avusteet

Ohjeet huomaa parhaiten, jos ne annetaan käyttöliittymässä itsessään syöttökentän vieressä tai itse syöttökentässä. Myös kenttään liitetyt automaattisesti ilmestyvät vihjelappuset ovat käyttökelpoisia.

Esimerkki:

Alkamisaika (pp.kk.vvvv): \_\_\_\_\_

Alkamisaika: pp.kk.vvvv

Alkamisaika: 30.11.2000

Alimmaisessa esimerkkitapauksessa oletusarvo antaa samalla ohjeen päiväyksen muotoilusta.

Käyttöohje, mikäli sellaista tarvitaan, pitäisi tarjota on-line avusteena esimerkiksi hypertekstinä. Käyttöohjeeseen turvautuminen on usein käyttäjälle kuitenkin viimeinen vaihtoehto, jos ongelma ei muuten ratkea. Käyttöohjetta ei lueta kuten kirjaa alusta loppuun, vaan käyttäjä yrittää ohjeen avulla löytää ratkaisun ongelmaansa. Ohjeen pitäisi tästä syystä tarjota etsintämahdollisuus. Vaikka käyttöliittymässä samasta asiasta pitäisi käyttää samaa nimeä, tämä ei päde etsintään käyttöohjeesta. Käyttäjät eivät välttämättä tiedä millä nimellä järjestelmän tekijä on päättänyt kutsua jotain toimenpidettä, joten asian etsiminen synonyymien avulla olisi toivottavaa.

## 6.2.2 Tehokkuus

Käytön tehokkuus usein tärkeä tekijä järjestelmissä, sillä tehokas käyttöliittymä saattaa aiheuttaa merkittäviä kustannussäästöjä. Seuraavassa tarkastellaan kolmea tehokkuuden osatekijää vasteaikaa, käyttäjän työmäärää sekä huomion herpaantumista.

### Vasteaika

Vasteajalla (response time) tarkoitetaan aikaa, joka kuluu käyttäjän toimenpiteestä siihen, että järjestelmä reagoi siihen jollain palautteella. Vasteajan ollessa alle 0.1 sekuntia, käyttäjällä on tuntuma välittömästä reagoinnista. Vasteajan kasvaessa tästä, käyttäjä huomaa viiveen. Jos vasteaika pysyy alle sekunnissa, viive ei vielä häiritse eikä edellytä ilmoitusta toiminnon kestosta. Vasteajan kasvaessa yli 10 sekuntiin käyttäjän huomio herpaantuu ja hän ryhtyy tekemään muita tehtäviä. Yli parin sekunnin vasteajat edellyttävät jo ilmoitusta odotettavasta pitkästä kestosta.

Yleensä ongelmana ovat pitkät vasteajat. Tehokkuus laskee koska käyttäjä vain odottelee vastausta tekemättä mitään. Joskus liian nopea vasteaikakin saattaa olla ongelma. Esimerkiksi jokin ilmoitus katoaa näkyvistä ennen kuin käyttäjä ehtii lukea sen.

Vasteaikaan voidaan vaikuttaa vähentämällä siirrettävän tiedon määrää. Siirretään vain sellaista tietoa, jota käyttäjä välttämättä tarvitsee tehtävässään. Esimerkiksi, jos käyttäjä tarvitsee tietoja yhdestä asiakkaasta, ei tämän saamiseksi ole tarpeen siirtää kaikkien asiakkaiden tietoja. Siirrettävä aineisto voidaan jakaa osiin ja käyttäjälle annetaan mahdollisuus hakea lisäosat tarpeensa mukaan. Kuvat ovat yleensä varsin suuria siirtokohteita (paljon enemmän kuin 1000 sanaa). Esimerkiksi www-sivujen kohdalla kannattaa harkita kuvien tarpeellisuutta.

## Työmäärä

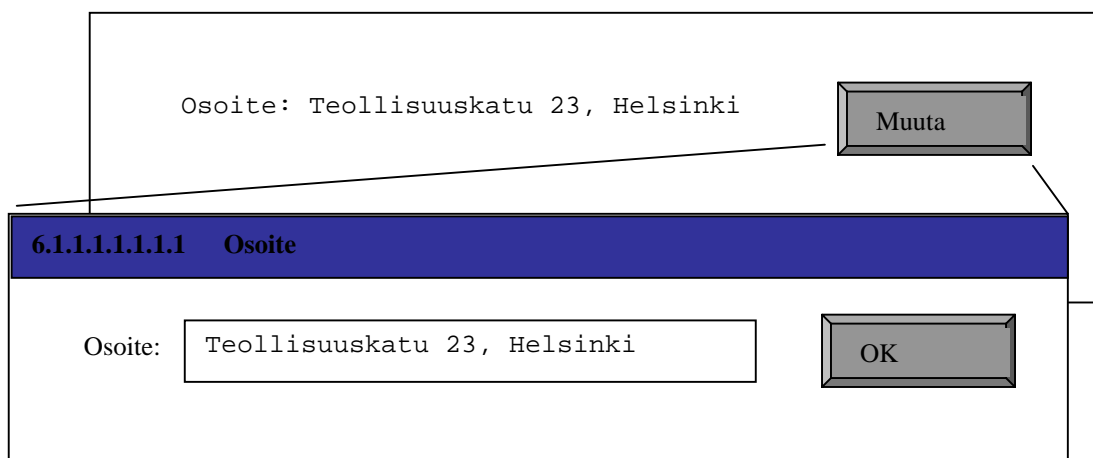
Käyttäjän työmäärää voidaan mitata esimerkiksi

- operaation suoritukseen kuluvalle ajalle
- näppäinpainallusten määrällä
- hiirioperaatioiden määrällä
- kirjoitettavan tiedon määrällä.

Näiden vähentäminen lisää tehokkuutta.

Turhien tehtävien karsiminen vähentää työmäärää. Esimerkkeinä turhista tehtävistä voidaan mainita

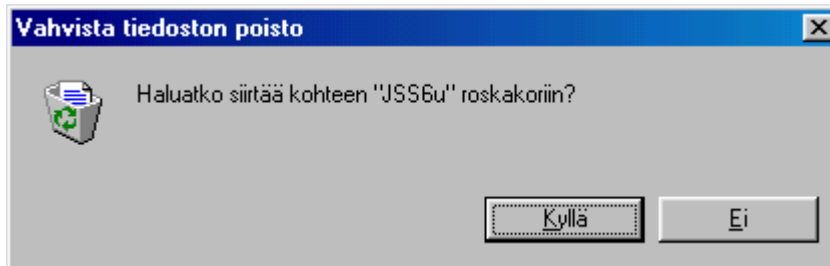
- Erillisen editointi-ikkunan avaaminen tiedon muuttamista varten kun editoinnin voisi tehdä suoraankin



- Tarpeeton kuittailu

Kuittauksen pyytämistä operaatiolle voi perustella sillä, että kuittauksella estetään käyttäjää tekemästä jotain fataalia virhettä. Kuittaus ei kuitenkaan välttämättä toimi edes tässä tilanteessa. Se muodostuu nopeasti automaattitoiminnoksi ja silloin kun siitä olisi hyötyä tuleekin operaation keskeytyksen asemasta valittua operaation jatkaminen. Kuittauksiin vastailu

vie myös käyttäjän huomion kuittausikkunaan, joka saattaa jopa peittää sen asian johon kuittaus liittyy. Ei fataaleissa tilanteissa kuittaukset ovat pelkästään hidaste, joka ärsyttää käyttäjää.



Tällä kuittauksella ei estetä mitään fataalia virhettä, sillä roskakori itsessään on tekniikka, jolla voidaan pelastaa vahingossa poistetut tiedostot. Kuittaus on siis turha.

- Hyppiminen ikkunoiden välillä

Nykyisetkin näytöt ovat usein niin pieniä, että useita ikkunoita käytettäessä ikkunat menevät osittain päällekkäin. Siirryttäessä ikkunasta toiseen käyttäjä joutuu uuteen ympäristöön ikäänkuin uuteen huoneeseen.<sup>2</sup> Onko tällainen siirtyminen tarpeellista ja onko siitä joitain hyötyä? Käyttäjä saattaa seota ikkunoissa ja tehdä huomaamattaan muutoksia ikkunaan johon ei ollut niitä tarkoittanut.

Yhden ikkunan sisällä toimittaessa ikkunan sisältö saattaa olla lipukkeilla jaettu useaan 'alisivuun'. Näiden sisältö on suunniteltava huolellisesti jotta vältettäisiin myös tarpeeton hyppiminen alisivujen välillä.

- Tiedon syöttäminen moneen kertaan

Tietokoneissa on muisti, jossa tietoa voi säilyttää. Täten tiedon syöttäminen kertaalleen riittää eikä samaa asiaa pidä kysyä uudelleen vaikkapa siirryttäessä uudelle näytölle.

---

<sup>2</sup> Cooper A: About Face, IDG Books Worldwide, 1995

- Tietokoneen tiedossa olevien asioiden kysyminen käyttäjältä  
Tietokoneessa on esimerkiksi kello ja kalenteri.

Hiiren liikuttamista, katseen harjailua ja näppäilyjen määrää voi vähentää hyvällä käyttöliittymäelementtien sijoittelulla. Oikoteiden (shortcut) käyttö nopeuttaa usein toistuvia operaatioita. Ikkunakoko olisi suunniteltava sellaiseksi, ettei ikkunaa tarvitse vierittää etenkään vaakasuunnassa.

### Huomion herpaantuminen

Huomion herpaantuminen aiheuttaa asiayhteyden vaihtumisen käyttäjän mielessä. Jatkettaessa toimintaa on asiayhteys rakennettava uudelleen ja tämä vaatii aikaa. Huomion herpaantumisen voi aiheuttaa:

- liian pitkä vasteaika
- ponnahdusikkunat ja –viestit
- näytöllä tapahtuvat muutokset kuten vilkkuvat kuvat ja tekstit, piippaukset, ja mainokset.

### 6.2.3 Muistettavuus

Muistettavuudella tarkoitetaan sitä, että käyttäjä muistaa oppimansa seuraavalla käyttökerralla. Muistettavuuteen voidaan vaikuttaa samoin keinoin kuin opittavuuteen:

- selkeät ja yksinkertaiset toimintaketjut
- vihjetekstien käyttö
- valinta kirjoittamisen asemasta
- yhdenmukaisuus

Joissakin järjestelmissä käyttäjä voi säätää käyttöliittymän omien tarpeidensa mukaiseksi. Järjestelmän pitäisi tällöin muistaa tehdyt säädöt seuraavaan käyttökertaan.

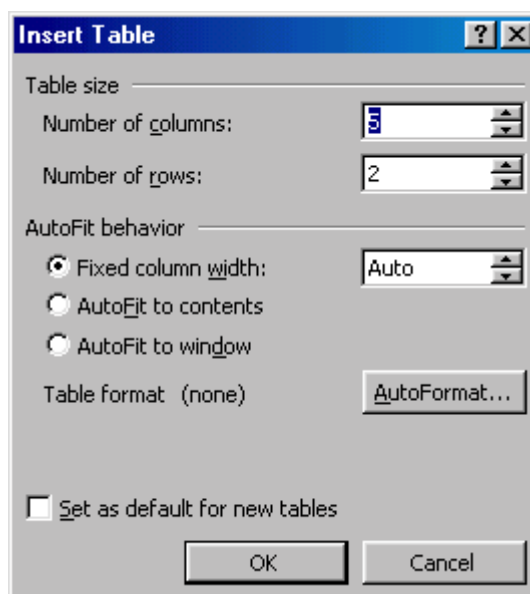
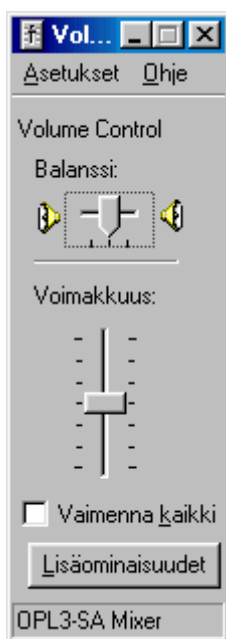
## 6.2.4 Virheettömyys

Ohjelman pitäisi tietenkin toimia virheettömästi ainakin, jos sille annettavat syötteet ovat oikein. Toiminta virheellisten syötteiden yhteydessä vaikuttaa käytettävyyteen. Käytettävyyden kannalta parasta olisi käyttöliittymä joka ei mahdollista virheellisiä syötteitä. Käyttäjät tekevät kirjoittaessaan kirjoitusvirheitä. He saattavat myös suorittaa operaatioita, joita eivät aikoneet. Käyttöliittymä tulisi rakentaa virheitä ehkäiseväksi.

Virheiden ehkäisy

Virheitä ehkäiseviä tekniikoita ovat esimerkiksi:

- Valitseminen kirjoittamisen asemasta  
Tällä tekniikalla voidaan eliminoida kirjoitusvirheet. Valittavaksi pitäisi tarjota vain kyseisessä valintatilanteessa kelvolliset vaihtoehdot.
- Arvovälirajoitteiset syöttökentät  
Esimerkiksi liukusäätimet ja kasvatettavat kentät.



- Ei kysytä asioita, jotka järjestelmä tietää
- Arvataan oikea vaihtoehto  
Kelpuutetaan isot ja pienet kirjaimet. Annetaan käyttäjän kirjoittaa useassa eri muodossa, järjestelmä muuttaa tiedon mieleisekseen.
- Hyväksytään vaihtoehtoisia syötteitä
- Väljennetään syntaksivaatimuksia  
Esimerkiksi hankinta-aika kenttään voisi kelvata päiväyksen asemasta myös pelkkä vuosi tai vuosi ja kuukausi.
- Automaattisesti täydentyvät kentät
- Automaattinen muotoilu syöttökentässä
- Täyttöohjeet kentän yhteydessä tai itse kentässä
- Numeerinen kenttä hyväksyy vain numeerista dataa ja reagoi välittömästi kun kenttään yritetään kirjoittaa tekstiä
- Käytetään jotain mediaa, jolla tieto voidaan välittää kirjoittamatta esim. tunnistuskorttia.

## Virheilmoitukset

Virheitä tehdään eikä kaikkia voi estää. Jos syötettävä tieto on välttämätön toiminnan jatkamisen kannalta on siitä ilmoitettava käyttäjälle. Seuraavassa tarkastellaan keinoja virheestä ilmoittamiseen.

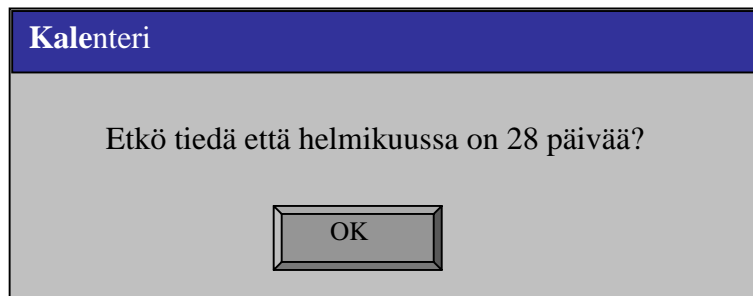
- Ei mitään ilmoitusta  
Operaation onnistumisesta voidaan antaa positiivista palautetta. Tällöin esimerkiksi kentän väri muuttuu kun sinne on kirjoitettu hyväksyttävä arvo. Kursori voi myös siirtyä seuraavaan kenttään tai näytöllä tapahtua haluttu muutos. Jos positiivista palautetta ei tule, käyttäjä huomaa tehneensä virheen.
- Kentän viereen ilmestyvä opaste  
Jos syötetty arvo on virheellinen ilmestyy kentän läheisyyteen opasteteksti eikä kentästä pääse pois.

- Erilliseen ilmoituskenttään ilmestyvä ilmoitus  
Ilmoituskenttään kirjoitettava ilmoitus on kentän viereen kirjoitettavaa huonompi siinä mielessä, että käyttäjä joutuu siirtämään katseensa ilmoituskenttään ja sitten takaisin syöttökenttään. Toisaalta erillinen ilmoituskenttä voi johtaa selkeämpään ulkoasuun.
- Esiinponnahtava ilmoitusikkuna  
Esiinponnahtava ilmoitusikkuna on standarditekniikka useissa sovelluksissa. Ikkuna vaatii käyttäjältä kuittauksen tai käyttäjän on muutoin suljettava ikkuna. Ikkuna saattaa ilmestyessään peittää virheellisen kohdan. Nähdäkseen virheellisen kohdan käyttäjä joutuu tällöin siirtämään ilmoitusta. Käyttäjä pääsee muuttamaan tietoja vasta suljettuaan ikkunan. Muutettaessa ilmoitus ei ole enää näkyvä eikä siten toimi avusteena.

Annettavan ilmoituksen tulisi olla oikein. Sen tulee olla opastava ja asiallinen. Ilmoitus ei saa olla käyttäjää loukkaava, syyttävä, ärsyttävä, 'hauska', eikä typerä.

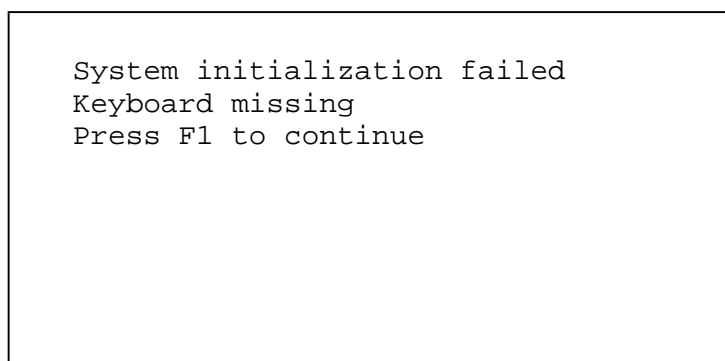
Esimerkkejä:

- Oraclen PL/SQL kääntäjä antaa virheilmoituksen  
*' Rivillä 5 sarakkeessa 30 on ....'.*  
Kääntäjä ei kuitenkaan ota rivejä laskiessaan lainkaan huomioon kommenttirivejä. Tällöin rivitieto ei ole käyttäjän kannalta oikein.
- Loukkaava tai hauska – tyhmä käyttäjä



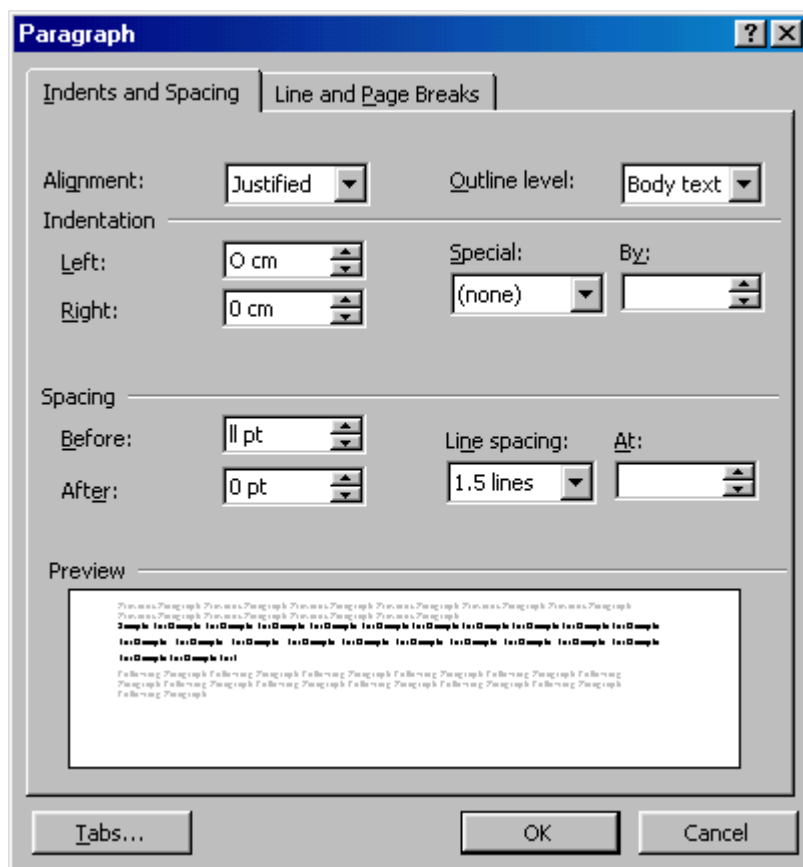


- Typerä



Virheestä on syytä ilmoittaa heti kun se on syntynyt, esimerkiksi kun ollaan poistumassa kentästä, eikä vasta ikkunaa suljettaessa tai lomaketta lähetettäessä. Myöhään annettavasta ilmoituksesta ei välttämättä selviä missä virhe on. Käyttäjä joutuu etsimään virhekohtaa ja palaamaan takaisin.

#### Ikkuna



ja sen virheilmoitus lopussa



Mikä kenttä on väärin?

Peruutusmahdollisuus eliminoi vakavat virheet. Monen askeleen peruutus on parempi kuin yhden askeleen.

Lisää turvallisuutta

Automaattinen tilan tallennus estää tehdyn työn mitätöitymisen. Ohjelman lopussa tehtävä automaattitalletus suojaa käyttäjää talletuksen unohtamiselta. Automaattitalletusta ei ole kuitenkaan syytä tehdä olemassa olevan tiedoston päälle vaan siitä on syytä tehdä uusi versio. Versiointi lisää turvallisuutta päälletallennukseen verrattuna.

Liian lähellä toisiaan olevat operaatiot ovat vaarallisia:

Esimerkki: UNIX-komennot

- rm – tiedostojen poisto
- em – emacs editorin käynnistys
- e ja r ovat vierekkäin näppäimistöissä.

### 6.2.5 Tyytyväisyys

Tyytyväisyydellä tarkoitetaan käyttäjän kokemusta käyttöliittymän miellyttävyydestä. Tyytyväisyyteen vaikuttavat käyttäjän arvostukset ja tottumukset. Seuraavassa käsitellään muutamia tyytyväisyyteen vaikuttavia tekijöitä. Myös aiemmin opittavuuden, tehokkuuden ja virheettömyyden yhteydessä esitetyt asiat vaikuttavat tyytyväisyyteen.

## Värit

Käyttöliittymässä voidaan käyttää värejä esimerkiksi

- asioiden ryhmittelyyn
- huomion kiinnittämiseen
- houkuttelevuuden lisäämiseen ja
- tyylikkyyden tavoitteluun.

Värejä on syytä käyttää hillitysti. Liiallinen värimäärä johtaa värikaakseen. Näytöstä tulee levoton ja sekava. Räikeät värit saattavat ärsyttää käyttäjää. Näytöt ovat erilaisia ja tietty värikoodi saattaa eri näytöillä ja näytönohjaimilla tuottaa eri värin. Kaikki värit eivät sovi yhteen, esimerkiksi

- vihreä pohja, pinkki teksti
- sininen pohja, punainen teksti.

Taustan ja tekstin kontrastieron tulisi olla mahdollisimman suuri, esimerkiksi musta teksti valkoisella tai vaalealla pohjalla. Taustan tulisi olla tasavärinen tai hyvin heikosti kuvioitu. Jos jollekin asialle on sovittu tietty väri ei koodausta ole syytä muuttaa. Esimerkiksi WWW-selaimet näyttävät linkit oletusarvoisesti sinisellä ja alleviivattuina. Joissain selaimissa käyttäjä voi itse määritellä linkkien esitystavan. Sinistä alleviivattua tekstiä ei tulisi WWW-sivuilla käyttää mihinkään muutun tarkoitukseen. Sivua ei saisi myöskään määritellä syrjäyttämään käyttäjän tekemiä selainmäärittelyksiä<sup>3</sup>.

## Fontit

Fontteja tulisi kuten värejäkin käyttää hillitysti, 4-5 fonttia korkeintaan. ISOILLA KIRJAIMILLA KIRJOITETUN TEKSTIN LUETTAVUUS ON HUONOMPI KUIN SEKATEKSTINÄ KIRJOITETUN TEKSTIN. Kirjasinkoon tulisi olla riittävä.

## Muistin rasitus

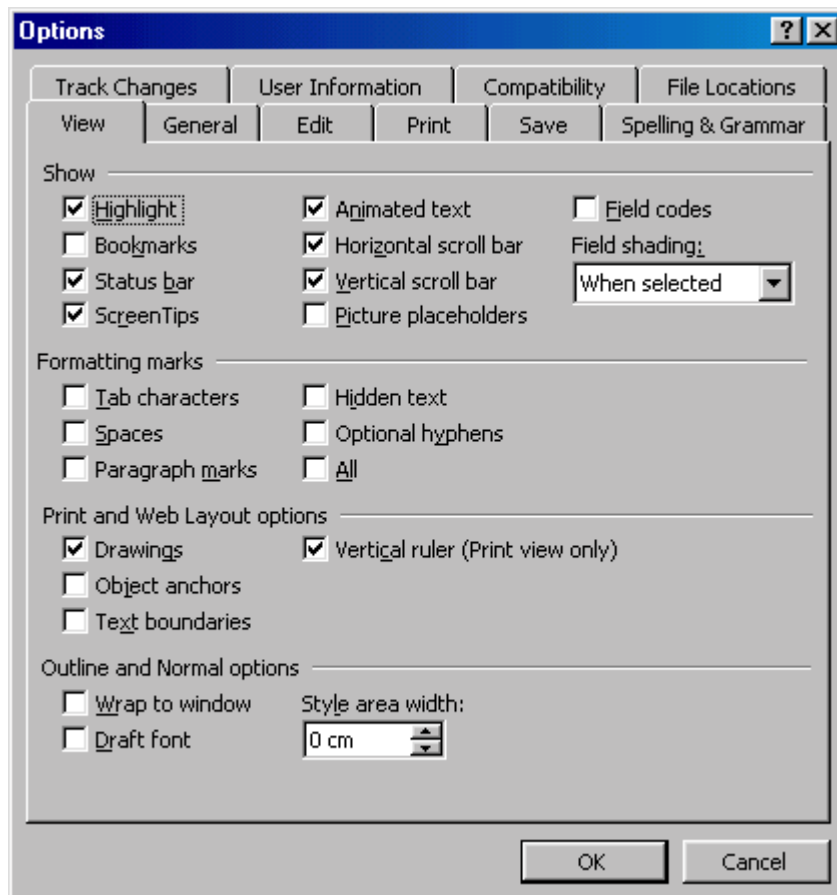
Ihmisillä on kapasiteetiltaan rajattu työmuisti. Sen koko on  $7\pm 2$  objektia. Tällainen objekti voi olla jokin yksittäinen luku tai jopa kokonainen kuva tai lause. Suuri elementtien määrä käyttöliittymässä saattaa johtaa työmuistin ylivuotoon. Käyttäjä saattaa hahmottaa jonkin diagrammin yhtenä objektina mutta jastaavan asian esittävän lukujoukon jokaisen luvun eri objekteina. Tästä syystä kaavioiden käyttäminen on selkeää ja nopeammin hahmotettavaa kuin numerotiedon.

## Paikkatieto

Usein käyttöliittymä muodostuu useasta 'sivusta' ja käyttäjä navigoi sivulta toiselle. Käyttäjän työskentely ohjelman kassa saattaa keskeytyä eri syistä, esimerkiksi puhelinsoiton johdosta. Kun käyttäjä jatkaa työskentelyään hänen on aluksi palautettava mieleensä missä vaiheessa työskentely oli. Käyttöliittymän tulisi auttaa käyttäjää nopeasti palauttamaan asiayhteys. Lipukkeiden (tabs) käyttö sivujen osoittamiseen on havainnollinen esittää paikka, jossa ollaan.

---

<sup>3</sup> Nielsen J: Designing Web Usability, New Riders Publication, 1999



Yllä olevassa kuvassa lipukkeet muodostavat yksitasoisen valikon. WWW-sivuilla käytetään joskus hierarkkisia lipukkeita, joissa lipukkeen valitseminen lisää valikkoon seuraavan hierarkiatason.

## Turvallisuus

Käyttäjä on tyytyväinen, jos hän pystyy suorittamaan tehtävänsä vaivattomasti ja turvallisesti. Tehdyn työn tulokset eivät yllättäen katoa eikä käyttäjän tarvitse jatkuvasti varoa toimintojensa vaikutuksia.

## 6.2.6 WWW-sibujen käytettävyys

Nielsen on artikkelissaan <sup>4</sup> määritellyt 10 asiaa joita WWW-sivuilla tulisi välttää: seuraavassa nämä on lueteltu Nielsenin myöhemmin määrittämää tärkeysjärjestyksen mukaisesti<sup>5</sup>

- Vanhentunut tietosisältö
- Virheet
- Hidas latautuminen
- Liikkuva teksti ja animaationsilmukat
- Epästandardit linkit
- Navigointituen puute - ei tiedetä missä ollaan
- Mutkikkaat URL:t
- Kehykset - suurin ongelma on, ettei alisivuun pysty viittaamaan
- Orvot sivut - ei linkkejä tai paluumahdollisuutta
- Vieritettävät linkkilistat - käyttäjä ei etene loppuun.

Myöhemmin Nielsen on vielä täydentänyt puutelistansa <sup>6</sup>

- Back-nappulan toiminnan estäminen tai hidastus
- Uusien ikkunoiden avaus
- Käyttöliittymäkomponenttien epästandardi käyttö
- Tekijätietojen puute
- Arkistojen puute
- Ohjaus toiseen osoitteeseen
- Otsikot jotka eivät liity sisältöön
- Huonot hakusanat
- Hitaat palvelimet
- Kaikki mainokselta näytävä.

---

<sup>4</sup> Nielsen J: Top Ten Mistakes in Web Design, <http://www.useit.com/alertbox/9605.html>, 1996

<sup>5</sup> Nielsen J: "Top Ten Mistakes" Revisited Three Years Later, <http://www.useit.com/alertbox/990502.html>, 1999

<sup>6</sup> Nielsen J: The Top Ten *New* Mistakes of Web Design, <http://www.useit.com/alertbox/990530.html>, 1999

Nielsen on myös listannut asioita, joita pitäisi löytyä kaikilta WWW-sivuilta<sup>7</sup> Näitä ovat:

- Nimen ja logon sijoittaminen jokaiselle sivulle ja logon toiminta linkkinä kotisivulle
- Etsintämahdollisuus, jos sivustossa on ylöi 100 sivua
- Suoraviivaiset ja yksinkertaiset otsakkeet sekä sivunimet
- Silmäilyä tukeva esitystapa – oleelliset ensin yhdellä silmäyksellä
- Hypertekstin hyväksikäyttö – pois lineaarisesta esityksestä
- Tuotteen kuvat – jos kyseessä on tuote-esittely
- Kuvien rajaaminen vain oleelliseen ja tärkeään
- Linkkivihjeet, jotka kertovat mihin linkki johtaa
- Lukemisen mahdollistaminen myös käyttörajoitteisille
- Yhdenmukaisuus muiden kanssa

---

<sup>7</sup> Nielsen j: Ten Good Deeds in Web Design, <http://www.useit.com/alertbox/991003.html>, 1999