

Felhasználói felületek

Definíció

Történelem

Tervezési alapelvek

Megvalósítás (UNIX, Windows XP, Mac OS)

Tanulságok – néhány „klasszikus” hiba

Olvasnivalók

Definíció

- A felhasználói felület az ember és a számítógép kapcsolattartását biztosító réteg

- Lehetőséget ad az adatok be- és kivitелére
- Irányítási és vezérlési funkciókat valósít meg
- Hatékony munkavégzést tesz lehetővé
- Esztétikai élményt nyújt



Történelem

1) A kapcsolók és a mérnöki pult

- Kizárólag funkcionális felület
- Kapcsolók és lámpák biztosítják a vezérlést és a visszajelzést



Történelem

2) A parancssoros felület

- Az adatbevitel és -megjelenítés szöveges
- Az interakciók köre erősen korlátozott
 - Szövegek begépelése és olvasása
- Egyszerre egy feladatot adhatunk a gépnek
 - Fejlettebb rendszereknél:
 - A feladatok összekapcsolhatók (kötegetl üzem mód)
 - Az indított programok „leválhatnak” a parancsértelmezőről
 - Párhuzamosan több parancssoros felületet (terminált) használhatunk

Történelem

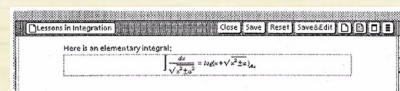
3) Grafikus felületek

- Az adatbevitel szövegesen és/vagy mutatóeszközzel történhet
- Az adatok megjelenítése grafikus
- Az interakciók köre sokkal bővebb
 - Kattintások, húzás, ejtés, vezérlők és kezelőszervek állapotmódosítása
- Egyszerre több feladatot adhatunk a gépnek
 - Többfeladatos ablakozó rendszereknél a képernyőn több ablakban dolgozhatunk

Történelem

3) Grafikus felületek

- Asztal-metafora
 - Nem új elképzelés
 - Első ötlet: Vannevar Bush, 1945 (!)
 - Megvalósítás: Xerox (Palo Alto Research Center), 1973



- A grafikus felület munkaasztal mintáz
- Elemei: ablakok, ikonok, menük, egyszerű és összetett vezérlők (kezelőszervek)

Tervezési alapelvek

a) A felhasználó megismerése

- Célszerű meghallgatni a felhasználókat
 - o A tervező sokszor nem, vagy rosszul ismeri az „átlagos felhasználót”
 - o A szoftverek használata nem könnyű
 - o A szoftverfelületek használatának képessége nem azonos az intelligenciával(!)
- Fontos a felhasználói tesztelés
 - o A felület tervezője sok hibát nem vesz észre
 - o A módszer általában a megfigyelés

Tervezési alapelvek

b) Következetesség

- Belső következetesség
 - o Felhasználói műveletek kezelése
 - o Pl. a gyorsbillentyűk mindenhol azonosak-e
 - o Program elemeinek kezelése
 - o Látható és nem látható elemek
 - o Belső funkciók elérhetősége
- Külső következetesség
 - o Megfelelés a felhasználó várakozásainak
 - o Megfelelés az operációs rendszernek
 - o Megfelelés a cégszintű szabványoknak
 - o Megfelelés az alkalmazástípus szabványainak

Fontosság



Tervezési alapelvek

c) Vezérlési képesség érzetének megadása

- Állapot jelzése és megtartása
 - o Következtesen reagáló állapotjelzők
 - o Pl. kiválasztott objektum, állapotsor...
 - o Az egymástól szinte megkülönböztethetetlen „üzemmódok” használata hátrányos
 - o Kontextus érzékeltetése („most hol is vagyok?”)
 - o Éppen nem elérhető funkciók, nem aktív elemek jelzése
 - o Kerülendő a hirtelen kontextusváltás
 - o Beállítások megjegyzése
- Véges, de szabadon kezelhető felület
 - o Ne kelljen keresni az alapvető információkat
- Alapértelmezések és „gyorsított eljárások”

Tervezési alapelvek

d) Metaforák alkalmazása

- A felhasználó számára ismert világ elemeinek tükrözése
 - o A metafora alkalmazása legyen átfogó
 - o Egyszerre több metafora is alkalmazható
 - o Pl. zeneszerkesztők: magnó & kotta metafora
 - o A metaforák a forrás korlátait is öröklik
 - o A metaforák sokszor csak egy kultúrán belül érthetők
 - o Pl. amerikai levelesláda:



Tervezési alapelvek

e) Funkciók rendelkezésre bocsátása

- Az „absztrakt” és az „érzékelhető” felületet kedvelők aránya 1:3
 - Absztrakt: parancssor, script, plugin stb.
 - Érzékelhető: eszköztár, párbeszédablak stb.
- Funkciók rétegzett elérhetősége
 - o Eszközsor
 - o Menüpont
 - o Almenüpont
 - o Párbeszédablak
 - o Másodlagos párbeszédablak
 - o „Speciális” párbeszédablak
 - o Vezérelt funkciók (pl. script-ek)



Absztrakció

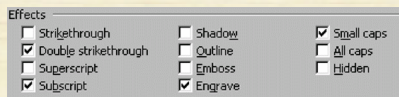
Tervezési alapelvek

f) Hatékonyság

- Felhasználói kontra programozói
 - o Pl. mikrohullámú sütő
- Várakozási idő minimalizálása
 - o Érzékelt kontra tényleges várakozás
 - o Többszálú feldolgozás
 - o Többfokozatú visszajelzés
- Igazi hatékonyság csak a teljes rendszerre vonatkozóan érhető el
 - o Hatékony részfolyamatok ≠ hatékony rendszer
 - o Fitts-szabály

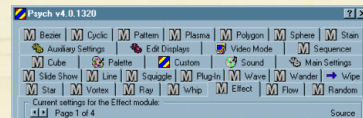
Néhány „klasszikus” hiba

- Elemek nem megfelelő célú használata
 - Microsoft Word, betűformázási beállítások
 - Egymást kizáró választási lehetőségek pipákkal
 - Jobb megoldás: rádiógombok / kombinált lista



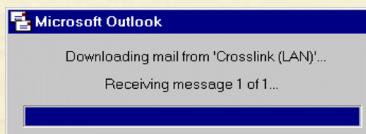
Néhány „klasszikus” hiba

- Emészthetetlen adatmennyiség – túl sok azonos vagy hasonló kezelőszerv
 - Más típusra vagy áttevezésre lenne szükség



Néhány „klasszikus” hiba

- Folyamatjelző mindössze 1–2 fokozattal
 - Microsoft Outlook
 - Összesen két állapotú folyamatjelző
 - Információtartalma nincs, félrevezető



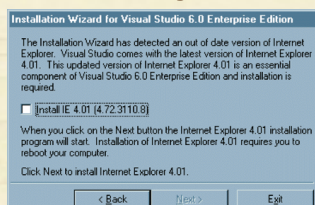
Néhány „klasszikus” hiba

- Olvashatatlan feliratok




Néhány „klasszikus” hiba

- A „varázsló” felület túlzásba vitt használata
 - Microsoft Visual Studio 6.0
 - A felhasználónak nincs választási lehetősége értelmetlen a pipa használata – sőt, az egész párbeszédablak felesleges



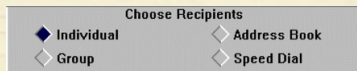
Néhány „klasszikus” hiba

- Rossz/eltorzult metafora
 - Apple MacOS
 - Kuka  a felhasználókat azonnal a törlésre asszociálnak
 - A kukába dobás azonban lemezkiadási funkciót is megvalósít



Néhány „klasszikus” hiba

- Felesleges egyénieskedés
 - IBM Aptiva Communications Center
 - Felesleges, a felhasználót könnyen zavarba hozó megoldás



Olvasnivalók

- <http://www.cooper.com/>
 - Alan Cooper (Cooper Interaction Design) honlapja
- <http://www.asktog.com/>
 - Bruce Tognazzini (Nielsen Norman Group) honlapja
- <http://www.iarchitect.com/shame.htm>
 - Az itt bemutatott hibákkal kapcsolatos anyagok forrása