

Dr. Hódos Tibor

## Ember-gép kapcsolat

Az 50/1999. (XI. 3.) rendelet mellékletének 3. „ember-gép kapcsolat” című fejezete, amelyen fontos része a rendeletnek, annyira kevéssé vált érthetővé a laikus felhasználók számára, és egyúttal olyan vihart kavart a munkahelyeken alkalmazott „rendszergazdák” között, hogy érdemes alaposabban bemutatni, miről is szól a mellékletnek ez a fontos része.

Ennek a fejezetnek a részeként szerepel a képernyőn megjelenő, illetve ezek kinyomtatott formáján szereplő szövegek magyar helyesírási szabályok szerinti használata. Ezt értették félre – valószínűleg – a rendszergazdák. A közkeletűen használt Windows rendszerek és Office csomagok már magyar nyelven tüntetnek fel majdnem minden program adta lehetőséget, sűgőik is magyar nyelvűek. A magyar nyelv használatát senki sem várja el a Windows behívása során megjelenő (és a felhasználó számára közömbös tartalmú) 2-3 képernyőoldalon, mert ez a világon mindenütt szabvány számítógép-angol zsargon. Sokkal nagyobb problémát okoz a nem Windows alapon futó programok magyar adaptációja. Itt a követelmény egyszerűen megfogalmazható: mindannak, ami a magyar Windows-ban és a magyar Office-ban magyarul van, annak kell azokban a programokban is magyarul lennie. Ez megoldható, igaz csak komoly, aprólékos munkával. 2001-től már ennek a követelménynek is esedékes az ellenőrzése.

Van még egy igen fontos követelmény ebben a fejezetben: a dolgozó tudomása nélkül nem szabad semmilyen mennyiségi vagy minőségi ellenőrzést alkalmazni. A számítógép ad ilyen lehetőséget, jogos a titkos ellenőrzés tiltása.

### Felhasználóbarát szoftver

A számítógép nem akármilyen műszaki eszköz. A vele való foglalkozás egészen új követelményeket támaszt, amelyek inkább szellemiek, mint fizikaiak. Ezek pl. kognitív folyamatok, mint az észlelés, gondol-

kodás, emlékezet, tanulás. A szoftve-  
rekkel való foglalkozás nehézsége  
elsősorban azon múlik, hogy az egyes  
programok szolgáltatási lehetőségei  
és teljesítmény-potenciáljai igen na-  
gyok, de az ember pszichés, kogni-  
tív lehetőségeit kevéssé veszik figye-  
lembe a programok kialakításakor.  
Ennek következtében igen magas le-  
het a hibázás gyakorisága és alacsony  
szintű a programban rejlő lehetősé-  
gek kihasználása. Hogy egy program  
mennyire felhasználóbarát, olyan  
kritériumok alapján dől el, mint pl. a  
feladatnak való megfelelés, önme-  
gjelentő képesség, könnyű kezelhető-  
ség, az elvárásnak való megfelelés,  
hibatűrés, áttekinthetőség, viszony-  
lag egyszerű megtanulhatóság.

Mindenki, aki egyszer már végig  
olvasta egy egyszerű háztartási gép  
használat utasítását, pontosan tudja,  
milyen nehéz is ezen eszközök,  
videorekorder, mikrohullámú kony-  
hagép használata. A háztartáson kí-  
vül is sok mindennapi készülék ismer-  
etes, amelyekkel az emberek nehe-  
zen boldogulnak. Jó példa lehet erre  
a közlekedési jegyeket árusító automa-  
ta. Nem az intelligencia és nem a ta-  
paszlatat kérdése, hogy mindig újra  
azt tapasztaljuk, milyen sok ember  
nem a kívánt eredményt éri el ezekkel  
az eszközökkel.

### Funkció és alkalmazhatóság

Amellett, hogy egy terméket  
amennyire csak lehet funkcióképes-  
ségi kell tenni, arról is szó van mindig,  
hogy ennek a funkciónak az alkalmaz-  
zását meg kell könnyíteni az ember  
számára. Ennek a példája lehet a csavarhúzó. A csavarhúzó feladata: va-  
lahol egy csavart be- vagy kicsavar-  
ni. A csavarhúzó egy fém-pálca,  
amelynek a vége keskenyre van ala-  
kítva, hogy beférjen a csavar vágatá-  
ba. Ez a csavarhúzó az a része,  
amely a funkciójához szükséges. Ha  
csak a funkciót kellene a terméknek  
teljesíteni, akkor a csavarhúzó már  
kész is lenne. Ezzel a vékony fém-  
rúddal az emberi kéz nehezen boldo-  
gulna. Ha több csavart kellene egy  
ilyen eszközzel be- vagy kicsavarni,  
az ember keze bedagadna és kevés  
eredményt hozna. Tehát szükség van  
a termékek alkalmazhatóságát illető-

en is tervezésre. A csavarhúzóknak kell  
egy markolati rész, amely jól illik az  
ember kezébe. Így már optimálisan  
tudja a funkciót teljesíteni.

Ezt az egyszerű példát alkalmas  
lehet a mindennapi élet számos más  
eszközére is. Szerszámok, gépek és  
használati tárgyak az ember lehetősé-  
geihez, adottságaihoz alkalmazko-  
dó formában készülnek. Az ollók, a  
székek, a cipők, a vasalók, a fényké-  
pezőgépek, a személyautók stb. meg-  
felelő formájúak, méretűek, szélessé-  
gűek és súlyúak, valamint könnyen  
felhasználhatók.

### A szoftverek felhasználó-barátsága

Mi köze van a fentieknek a szoft-  
verek (a programok) felhasználó-bará-  
tságához? Először is, mi az, hogy  
szoftver? A szoftverrel akkor kerü-  
lünk csak kapcsolatba, ha bekapcsol-  
juk a számítógépet. A szoftver funk-  
cióit a képernyőről ismerhetjük meg.  
A „szoft” alatt „könnyűt” értünk,  
azaz a számítógépet (a hardvert) a  
programok hozzák működésbe.  
Amit a szoftverből a képernyőn lá-  
tunk, az folytonosan változik annak  
függvényében, hogy mit írunk be a  
gépbe. A számítógépnek sok funkci-  
ót adhatunk, ehhez több programot  
alkalmazhatunk: könyvelő, szöveg-  
szerkesztő, adat- és címfeldolgozó,  
grafikai tervező, internetező stb.  
programokat.

A hardverek kézzel megfogható-  
ak, a szoftverek nem. Ezért nehéz el-  
képzelni, mi is az a szoftver, és mit  
tud. Ehhez hozzátartozik az is, hogy  
a programok nagyon hasonlítanak  
egy labirintushoz; nekünk tehát meg  
kell tanulnunk, milyen célhoz, me-  
lyik utat válasszuk.

### A labirintus útjain

Ha egy levelet akarunk írni, akkor  
a szövegszerkesztő programban egy  
új dokumentumot kell előhívni, majd  
a levelet be kell írni a gépbe, a leve-  
let valamilyen néven el kell menteni  
a gépben, majd ki kell nyomtatni.  
Minden egyes lépés e feladatban sok  
lépésből áll, ezeket ismernünk és al-  
kalmaznunk kell. Amíg ezt megta-  
nuljuk, biztosan elkövetünk hibákat.

Dr. Hódos Tibor  
OKK-OMFI  
1097 Budapest, Nagyvárad tér 2.

De minden hibából megtanuljuk azt, hogy hogyan kell jól csinálni. Kutatások feltárták, hogy a kezdeti időben a számítógépes tevékenység 20%-a hibakereséssel és hibaelhárítással telik el.

Egy idő után már minden nehézség nélkül megy a levélírási feladat. Pontosan tudjuk, hogyan kell csinálni. Ha egy ilyen feladat-teljesítéssel megbarátkoztunk és bízunk a sikeres alkalmazásában, mihelyt a megszokott útról letérünk, új élményben lesz részünk. Gyakran rádöbbenünk arra, hogy egy számítógép program lényegesen több utat tartalmaz a labirintusban, mint amennyit valaha is alkalmazni fogunk. Ez azzal függ össze, hogy megszokott, kipróbált és ismert utat egy feladat megoldására nem szívesen cserélünk fel egy mással.

A szoftver kialakításától függ, hogy mennyire egyszerűen, gyorsan és biztosan lehet egy programmal megbarátkozni, és mennyire egyszerűen, gyorsan és biztosan tudjuk ezt a programot feladataink során felhasználni.

A számítógépes technológia alapvetően különbözik minden előző munkaeszköztől.

### A „régi” és az „új” technológiák különbsége

A számítógép minden hardver és szoftver formájában lényegesen más munkaeszköz vagy szervezőeszköz, mint az összes korábbi technológiák. Ennek következményei vannak magára a munkaeszközre, a termékre és a felhasználóval szembeni követelményekre nézve.

#### Mi ez?

A modern információ- és kommunikációs technológiák a felhasználó és az elérendő munkaeredmény között helyezkednek el. A potenciális termékek (iratok, vázlatok, táblázatok, autók) távvezérléssel készülhetnek. Ezzel a munka már kevésbé jelent fizikai kapcsolatot a dolgozó személy és a megmunkálendő tárgy között. Ez azt jelenti, hogy ez az új munka az „absztrakt folyamatvezérlésből” áll. Az emberek abszolút többsége el se tudja képzelni, hogyan települ rá a program a merev lemezre, ott elhelyezkedve hogyan működik.

A legtöbb eddigi gép egycélú be-

rendezés volt. Funkciója egyértelműen meghatározott volt, és már a neve is megfelelő felvilágosítást adott a rendeltetéséről. A markológéppel markolnak, a mosógéppel mosnak, az esztergapaddal esztergálnak, a járművekkel utaznak. Amit ezek a gépek csinálnak, az közvetlenül megfigyelhető, a gép működése azonosan ismételtető. A megfelelő kar elmozdítására a daru jobbra fordul, gázadáskor a kocsi felgyorsul. Amit ezek a gépek csinálnak, az három dimenziós, látható dolog. A gépműködés közvetlen eredményei a gép helyéhez kötöttek, és csak a konkrét helyzetben hatékonyak. Ezek a gépek kultúrtörténetileg az ember fizikai tevékenységének meghosszabbításai. Az igazgatási és szervezési munka túlsúlya egyre több kognitív követelményt támasztott, és új munkaeszközöket hozott létre.

További különbség az eddigi eszközökhöz képest, hogy a számítógép képességei nem az ember fizikai képességeit helyettesítik vagy hosszabbítják meg (emelés, húzás, mozgatás...), hanem, először a történelemben, a szellemi tevékenység egy része alól mentesítik az embert. A számítógépek elemeznek, regisztrálnak, osztályoznak, kombinálnak, számolnak, információkat tárolnak. A számítógép egy helyben marad, műveletei csak két dimenziós képből láthatóak, még akkor is, ha telephelyén kívül fejt ki hatását.

Az információfeldolgozási rendszerekben végzett munka eredményei nem anyagiak. A programok felhasználói így új helyzetbe kerülnek: a számítógépnek nincsenek olyan egyértelmű vezérlőeszközei, mint a daru emelőkarja. A számítógép vezérlőeszközei mindig más funkciót töltenek be, egyik sem látható közvetlenül a térben, a működési modellek a felhasználó fejében alakulnak ki.

### Pszichológiai szempontok

A szoftver-ergonómiát az ember és munkaeszköz közötti illesztőfelület értelmében időnként kognitív-ergonómiának is nevezik, szemben a monitorral, a billentyűzettel, az egérrel stb. foglalkozó hardver-ergonómiával, amely elsősorban ezek fizikai jellemzőit vizsgálja, míg a szoftver-ergonómia ezek alkalmazásához szükséges szellemi funkciókat vizsgálja. A szoftver-ergonómia központi kate-

góriái (észlelés, figyelem, emlékezet, gondolkodás, tanulás ...) a pszichológiából származnak.

### Statikus és dinamikus információ-feldolgozás

Egy program megtervezésekor két egymással szoros kölcsönös kapcsolatban álló elemzési kategória van: egyrészt a szoftver statikus szempontja (minden egyes mozdulatlan képernyőkép megtervezése), másrészt a szoftver dinamikus aspektusa (a felhasználó vezetése a program útvesztőiben, közvetlen párbeszéd folyamatok lefutása).

A statikus szempont egy film egyes kockáihoz, a dinamikus aspektus pedig egy rövid film lejátszásához hasonlítható. A film egyik kockáján pl. egy személy egy széken ül. Ha az előző és az éppen következő filmkockákat nem láttuk, nem tudhatjuk, hogy már régen ott ül és éppen most fog felkelni, vagy éppen most ült le. Hogy miről szól a kérdéses kisfilm és mi az üzenete, azt csak az egész film megtekintése után tudjuk meg. A statikus információ a képernyő egy, mozdulatlan képén található valamennyi aktuális jelet magában foglalja. Az ergonómiai értékelés pszichológiai kritériumok (észlelhetőség, egyértelműen értelmezhető jelzésviszonyok, struktúráltság, jelzések mennyisége és minősége (nagyság, szín, forma stb.) szerint történhet.

A szoftver dinamikus oldalát képviseli a felhasználó beavatkozása nyomán előálló minden változás a képernyő felületén, két képernyőoldal közötti időben.

A dinamika két egymást közvetítő statikus információ-egység közötti viszonyt fogható fel. A szoftver dinamikus jellemzőinek megfelelőségét olyan kritériumok szerint lehet értékelni, amelyek előnyösen befolyásolják a felhasználónál a kognitív dramaturgiát (emlékezeti, tanulási-, gondolkodási műveleteket). Ezzel érhető el a program céltudatos és a felhasználó által ellenőrzött vezérlése, irányítása. Szoftver-ergonómiai követelmények e tekintetben: a programok átláthatósága, áttekinthetősége és az elvárásoknak való megfelelése.

Amikor a számítógép programjával dolgozunk a szoftverfejlesztő rendezői funkcióit gyakoroljuk. A szoftverfejlesztő meghatározta, milyen utak közül választhatunk, és

mit választunk. Valahányszor a mi rendezői döntésünk eltér a programozó által választottól, hibás eredményhez jutunk, vagy zsákutcába kerülünk.

A dinamikus változások tehát folyamat-jellegűek, ami a felhasználót arra kényszeríti, hogy a dramaturgiát elképzelje vagy megtanulja, mert enélkül nem képes a programot vezérelni. A szoftver vezérlése a felhasználótól egy sor kognitív műveletet és cselekvést igényel. Hogy mennyire sikerül a célt eredményesen elérni, az attól függ, hogy mennyire képes a felhasználó a megfelelő, vagy éppen hibás kognitív műveleteket, illetve hipotéziseket kiválasztani, alkalmazni az elérni kívánt célíg vezető úton. Ebben fontos szerepet játszanak általában a szoftver-ergonómiai ismeretek és különösen ezek pszichológiai alapelvei.

A számítógép és a számítógépes programok azért válnak nehezen kezessé báránnyunkká, mert ellentétben a többi géppel, amelyeken a látható vezérlőszervek (emelőkar, kapcsoló, nyomógomb, forgótárcsás vagy egyenes vonalú szabályozó) állandóak, a számítógépen viszont többségük állandóan változik. (Egy autónál nem képzelhető el, hogy fékezni egyszer a bal, máskor a jobb pedállal lehet, vagy gázt lehet az indexváltókkal adni.) A lehetséges funkciók nagy száma a felhasználótól magasszintű emlékezeti és tanulási teljesítményt igényel, ezzel együtt növeli a hibázás kockázatát is.

### A felhasználó kognitív műveletei

Minden felhasználó impliciten olyan belső modellekből indul ki, amilyenekből a többi ember, amelyeket azok is kialakítanak, megtanulnak, sőt amiket használva tárgyalnak másokkal is. Minden olyan tárgynak, amelyet majd más emberek fognak kiszolgálni, a tervtanulmányában ilyen belső modellel is foglalkozni kell. Ha valaki egy használati utasítást ír, jól meg kell gondolnia, hogyan strukturalja ezt a fejezetet ahhoz, hogy az olvasók a tartalmat úgy értsék meg, ahogy azt a szerző elérni kívánta. A didaktika című tantárgy foglalkozik ilyen kérdésekkel. Valószínűleg számos technikai berendezés használati utasítását nem didaktikai megfontolások alapján állították össze, mert különben nem panaszkodnának annyian számtalan használati eszköz, berendezés használati utasítására.

A számítógép program rendszerezett elemzéséhez a felhasználónak olyan belső modellre van szüksége, amelynek révén lépésről-lépésre haladva elérheti a kitűzött célját. A felhasználó egyetlen műveleti egységnek kivitelezéséhez számos pszichikus folyamatra van szüksége.

### Az észlelés

A felhasználó egy képernyő-oldalról információkat vesz fel. A G. Miller-féle „mágikus” 5±2 törvény szerinti meghatározott mennyiségű információt vesz fel az ember „egyből” (azaz: azonnal), a többit meghatározott hierarchikus kritériumok szerint folyamatos egymásutániségben. Elsőként a legfeltűnőbbet, a mozgásban lévő, a színt, a szubjektíve fontosat észleli.

### 2. Értelmezés

Amit észlelünk, azt értelmezni kell, valamilyen jelentőséget tulajdonítunk az észlelt információnak. Az értelmezéshez a felhasználónak aktualizálnia kell a rövididejű vagy a tartós emlékezetében tárolt ismeretek egy részét.

### 3. Célképzetek és cselekvési tervek

A jelzések értelmezései alapján a felhasználóban kialakul egy célképzet. Célja elérése érdekében az ember kialakít egy kognitív modellt, milyen cselekedetek vezetnek az elérni kívánt célhoz, illetve kialakít egy gondolati tárgyalási szituációt („Ha ezt és azt csinálom, bekövetkezik amit akarok!”). Az ember esetleg több belső modellt hasonlít össze, és végül dönt az egyik mellett. Az előzetes tapasztalatoknak nagy szerepük van a cselekvési tervek létrejöttében. Ha tudjuk, hogy meghatározott cselekvés célravezető volt, és most ugyanaz a célunk (megint levelet akarunk írni), akkor azt választjuk. Ha most kell megtanulnunk a cselekvéssort, akkor azzal a feltételezéssel látunk hozzá a megoldáshoz, hogy valószínűleg sikerülni fog. Itt is szerepe van az előzetes ismereteknek: hasonló esetekben mi volt az eredményünk? Ha semmilyen sejtésünk sincs, akkor „próba-szerencse” alapján indulunk sétára az útvesztőben. Ez belső feszültséget okoz, félelem léphet fel, hogy valamilyen áttekint-

hetetlen vagy irreverzibilis szituációba kerülünk.

### 4. Az eredmény ellenőrzése, ismételtes

A mozgásos (motoros) feladat teljesítése után az ember (észlelés és értelmezés segítségével) megvizsgálja, hogy az eredmény megfelel-e a cselekvési tervben kitűzött célnak (eredmény-ellenőrzés).

Ha nem a kitűzött célt érte el az ember, akkor megint elkezdődik az észlelt képernyőkép észlelése, értelmezése vagy rögtön egy új cselekvési terv kialakítása kognitív modell formájában. Most ezt kezdi végrehajtani. Ha sikerült, akkor új feladatot kezdhet vagy befejezi a munkát a számítógéppel. Ha most sem sikerült, akkor újból kezdődik a kognitív modell kialakítása.

A felhasználók folytonosan változtatják a szenzomotoros szintet (látni, érezni, kézzel végzett cselekvések) és a kognitív szintet (észlelés, emlékezet, értelmezés, modellképzés, tanulás).

Az ember egyes műveletei eredményeként a képernyőn megjelenő, leolvasható hatások visszajelentésként (Feedback) értelmezhetők. A visszajelzés a felhasználó megfontolásai eredményességének a mértéke, azaz megerősíti vagy elveti a helyességét a felhasználó mindazon kognitív modelljeinek, amit eddig alkalmazott.

A tervezett cél és a képernyőn bekövetkezett hatásának összehasonlítása, a kettő közötti gondolati tevékenység ellenőrzése. Eközben minden egyes részlépés, amely hibát idézhet elő, a program ergonómiai elemzésének, illetve megalkotásának tárgya. Minél inkább az ember pszichés adottságainak megfelelően alakítanak ki egy programot, annál könnyebben képes az ember azzal megbarátkozni.

### Közös kommunikációs kódok

Minden szoftverfejlesztő abból indul ki, hogy az általa készített program éppen annyira világos lesz a felhasználóknak, mint saját magának. Ez azt jelentené, hogy a szoftver készítőjének sikerül a felhasználókban ugyanazon kognitív megfontolásokat előhívni, mint amelyek neki magának vannak. Minden szoftverfejlesztő-team számára ismeretes, hogy

nincs egyhangú vélemény a probléma-megoldáshoz vezető legjobb útról a program minden lépésénél. Ezért kell a felhasználóknak is kintódnia, ha azt keresik, melyik az ehhez vagy ahhoz a problémához a leginkább szituációs-specifikus, „igazi” út.

A kommunikációval foglalkozó tudományokban egységes a vélemény, mely szerint szükségszerű egységes kódrendszerben megállapodni a program-fejlesztőknek és a felhasználóknak. Nem engedhető meg, hogy az egyik fél csak „kínaiul” beszéljen, a másik viszont csak „spanyolul” tudjon.

### Minőségellenőrzés, irányelvek és módszerek

Az ergonómiai szoftver tervezés professzionalizálódásának szüksége van az ember információ-feldolgozásának alapismereteire és egy ennek megfelelő társadalom-tudományi módszer-választékra ahhoz, hogy ergonómiai szempontok alapján elemezzen, értékeljen, programterveket, kész programokat.

### Az átláthatóság, mint kritérium

Egy ergonómiai szempontoknak megfelelő szoftver alapkritériuma az átláthatóság. Ez a kritérium mindezekelőtt az egyes képernyőképekre vonatkozik, de teljes mértékig számon kérhető a program lefutását illetően is. Az átláthatóság megkönnyíti a programon belüli tájékozódást és az információ-egységeinek a feldolgozását. Az átláthatóság mennyiségi jellemzője az információ-egységek mennyisége, ezek megoszlása, kódolása, egyformasága-különbsége pedig a minőségi jellemzők.

### Kognitív szintek

Az információ-közlés strukturáltságának és formájának hatása van az ember észlelési (figyelmi, felismerési, azonosítási, megkülönböztetési teljesítményére, valamint a rövididejű emlékezetben tárolható információ mennyiségére, a tanulásra, tehát a tartós emlékezetre is.

Az információ-felvétel első állomása a receptor és a szenzoros bemenet. A piros-zöld vak személyek ezeket a színeket nem képesek izoláltan észlelni. A figyelem egy energetikailag igen korlátozott és egyidejűleg

párhuzamos igénybevételeket alig megengedő pszichés funkció. Nem lehet egyszerre rádiót hallgatni, TV-t nézni és beszélgetni. Ez azt jelenti, hogy a figyelem egymásutániságban veszi fel a különböző információkat, prioritást adva mindenkori a szubjektíve legfontosabbnak vagy legérdekesebbnek. Az információ-egységek, „csomagok” egymásután és egymástól elválasztva kerülnek felvételre, majd megtörténik azonosításuk. Ez a folyamat annál sikeresebb, minél elkülönítettebben kerülnek felvételre az információ-egységek.

A rövididejű emlékezetben tényleg csak rövid idő áll rendelkezésre az információk tárolására. Minél több információt kell felvenni, annál jobban megterhelik a rövididejű (operatív) emlékezetet. A tartós emlékezet a tanulás eredménye, tehát egy állandó információ-tároló. Egyfajta archívumnak tekinthető, amely az információ-egységeket tárolja. Az, hogy a tartós emlékezet mennyi ideig őrzi meg, azaz mennyire válik tanulttá a felvett információ-egység, az függ az információ jellegétől és a tanulási folyamat sajátosságaitól. A kérdés ugyanis felvetődik, hogy ha a tartós emlékezetben tárolt információra szükség van, milyen gyorsan és teljesen idézhető fel. (Őn biztos tanult Róma alapításáról, de mikor is volt ez?). Ez attól függ, milyen a tárolási forma, hasonlóan az archívumokhoz.

### A program létrehozásának szempontjai

Ha az átláthatóság ergonómiai követelményeit a program kialakításában figyelembe vesszük, úgy ebből többek között a következő követelményeket kell érvényre juttatni:

- A korlátozottan rendelkezésre álló figyelmet gazdaságosan kell felhasználni. Ez azt jelenti, hogy minden szükséges információnak jelen kell lennie a képernyőképen, de egyetlen felesleges információt sem szabad megjeleníteni. Minden felesleges jelzés szellemi terhelés.

- A feldolgozandó információ komplexitását csökkenteni kell, ez terheltesíti a rövididejű emlékezetet.

- A jelzések azonosítása és megkülönböztetése annál könnyebb, minél tartalmasabbak a felhasználó számára, azaz minél nagyobb a jelentőségük, és minél jobban strukturáltak. Érthetetlen hibajelzéseket meg kell előzni.

- Az emlékezeti tartalmak elraktározása és előhívása a tartós emlékezetből annál eredményesebb, minél jobban strukturáltak azok és minél gyakrabban „erősítik” meg azokat, azaz hívják elő őket. A grafikus felhasználói felületek elterjedése, pl. az ikon-szimbólumok formájában (nyomtatás, másolás, kivágás, beszúrás) a könnyítést jelentenek a tartós emlékezet számára, mert szöveges parancs előhívása helyett elegendő egy ikonra klikkelni.

A klasszikus pszichológia alaklélektani iskolájából ismert, hogy egy „egészséges” könnyen felismerhető a szituáció egészéből. „Egészséges” alatt egy összetartozó figurát értünk, ahol is bizonyos kritériumok ezt az összetartozást erősítik, pl. téri közelség, csoportosítás, azonos formák. Néhány vonal és görbe egy papírlapon egy vidék észlelését idézi elő, jóllehet az egyes elemei egyáltalán nem jelentenek vidéket. Egyes betűk egymásutánisága, mint pl. s,z,ü,n,e,t „egészet” eredményez, egy fontos szót: vége az órának.

Általában minél strukturáltabb a képernyőtartalom, annál kisebb szellemi megterhelést okoz az információ-felvétel, az azonosítás és megkülönböztetés, valamint az értelmezés. Fordítva is igaz kevéssé strukturált vagy strukturálatlanul ömlesztett jelzésekkel ugyanezen szellemi műveletek elvégzése időben elhúzódó, megterhelő feladat.

### Szoftverergonómiai követelmények

*A szoftver feleljen meg a feladatnak*

Ez mindazokat a kategóriákat jelenti, amelyek kiinduló alapként mindenki számára érthetőek és amelyeket a megoldandó feladat a szoftver funkcionálásától megkövetel. Ha ez így van, akkor a szoftver segíti és nem hátráltatja a feladat teljesítését.

*A program legyen könnyen megismerhető*

Az egyes képernyőképek és ezek egybefüggő futása a feladatmegoldás során adjon lehetőséget világos elképzelés kialakítására a programban nekünk rendelkezésünkre álló lehetőségekről. Ezek belső rendjéről és a párbeszéd lehetőségeiről, valamint a váratlan helyzetekben (pl. hibajelzések, rendszerüzenetek) teendő lépés-

seinkről minél többször és minél teljesebben tájékoztat a program, annál könnyebben tudunk vele együttműködni.

Idetartoznak olyan dolgok, hogy legyen állandó a funkcióbillentyűk és az egérműveletek módja, a képernyőképek, maszkok, menük és a hozzájuk tartozó listák, az ablakrendszer, a feladatnak éppen megfelelő segítségnyújtás (súgó, varázsló) álljon folyamatosan rendelkezésünkre, anyanyelvünkön kapjunk információt arról, hogy, milyen szakkönyv melyik részében kereshetünk kérdésünkre további válaszokat. A képernyőn alkalmazott színek, szimbólumok segítsenek a műveleti folyamatokban történő haladáshoz. Védjen meg a program az információ-túlterheléstől, informáljon az időről (időkorlátozról, hány óra van?).

### *A program legyen könnyen irányítható*

A felhasználó a szoftver plaszticitását és működésének ellenőrizhetőségét felhasználva legyen képes a programot irányítani. Más megfogalmazásban: ne legyen a felhasználó odaláncolva a program megállíthatatlan, továbbhaladásához, hanem módja legyen a haladás sebességét, bizonyos határok között a részfeladatok egymásutánosságát, (a program adta választási lehetőségeken belül) megváltoztatni, a feladatot időlegesen leállítani. A program adjon viszonylag nagy szerepet a felhasználó választásainak. Röviden: időnként legyen felhasználó irányította rendszer a rendszer irányított felhasználó helyett.

Idetartozik: kapjon a felhasználó lehetőséget arra, hogy néhány funkcióbillentyűt, műveleti mezőt szabadon határozhasson meg, legyen módja változtatni a párbeszéd módját, változtathasson a képernyőképen (eltolás, nagyítás, kicsinyítés, torzítás), az ablakokat egymás mögött vagy egymás mellett kiterítve jelenítse meg, készíthessen egyéni makrókat, változtathassa a műveletek egérrel, illetve billentyűvel való végzését.

### *A program feleljen meg az elvárásoknak*

A szoftver-használat során a felhasználóban kialakul egy elképzelés arról, milyen műveletek, milyen eredményekhez vezetnek. A prog-

ram segítse elő az ilyen akció reakció típusú elképzelések kialakítását, tapasztalati megerősítést, és ezzel optimalizálja az ok okozat típusú hipotézisek kialakulását. Rövidebb-hosszabb gyakorlás után a felhasználó jusson el a programmal kapcsolatban ahhoz a szintre, hogy kognitív terveit a legmegfelelőbbnek tűnő módon legyen képes megvalósítani a géppel.

Azonos információ-típusok kódolása maradjon állandó, azonos műveleti algoritmus azonos párbeszéd-történeten, a különböző típusú jelzések (rendszerjelzések, súgó- és varázsló-jelzések) legyenek állandó formájúak.

### *A program legyen hibaálló*

A hibás felhasználói műveletet a program ne hajtsa mereven keresztül. Ahol csak lehet, legyen lehetőség a hibás művelet visszavonására. Adatrögzítésnél minden beírt adatot jelezzen vissza a gép, ezzel is csökkentve a hibák arányát a végeredményben. A program legyen felkészítve a leggyakoribb hibázások automatikus korrekciójára, amihez felhasználótól újabb, most már helyes műveletet kér. A korrekciók ne vezessenek rendszerhibához, vagy a rendszer összeomlásához.

Tartalomfüggő súgói, varázslói segítség kell a hiba helyéről, okáról, kijavítási módjáról, az UNDO funkció felhasználásának megtanítása, biztonsági másolatok automatikus készítése, ismételt rákérdezések súlyos következményekkel járó beavatkozások előtt.

### *A program legyen átlátható*

A szoftverkészítés egyik alapvető szempontjáról van itt szó. Az egyes képernyőképeknek és ezek folyamatának figyelembe kell vennie a felhasználó ember információ-felvételi és -feldolgozási sajátosságait.

Példák: a képernyőnek maximálisan 50%-át szabad igénybe venni a programnak, a jelzéseket az alaklélektani és az észlelés-lélektani kritériumoknak megfelelően kell strukturálni, a menűfák inkább szélesek,

mint mélyek legyenek, egy menüben 7-8 pontnál több ne legyen, inkább analóg, mint digitális kijelzéseket alkalmazzon a gép, világosan legyen megkülönböztetve, hogy milyen funkciók állnak rendelkezésre és melyek nem, korlátozott számú kódolás legyen (szín, hely, szimbólum, hanghatás), „értelem-gazdag” információ-közlés absztrakt fogalmak helyett, az aktív munkaterületet a képernyőn ne fedjék el más jelzések.

### *A program legyen megtanulható*

Minden felhasználó arra alapoz, amit eddig megtanult a programból. A mi közép-európai olvasási technikánk balról jobbra és fentről lefelé halad. Egyes újabb programok az „analóg-tanulás” elvén alapulnak: az ismertet visszük át az ismeretlenre minden új tanulási lépésnél. Ha egy szituáció nem világos, a tanulás módszere a „próba-szerencse” típusú tanulás lesz. Amint az éppen a számítógépnél ismert, az emberek általában idegenkednek az ilyen tanulástól, mert félnek valami katasztrofához vezető hibázástól. Ezért egy belső korlát megakadályozza a legtöbb felhasználót a „próba-szerencse” típusú tanulástól. A programnak meg kell könnyítenie a kockázat és hibás tanulás megelőzését a programmal való gyakorlati együttműködés során.

A komplexitás csökkentése a legjobb tanulást könnyítő sajátossága egy programnak. A statikus képernyőképek és a dinamikus folyamat állandósága, a szokásos, régebbi ismeretekre való építés szükséges (pl. piros szín veszély, nyíl irány). A program átláthatósága nélkül nem könnyen sajátítható el. A program feleljen meg a rövididejű operatív emlékezet terjedelmének, a tartós emlékezet strukturáltságának, mert így könnyebb a megtanulása.

Végül szeretnénk megjegyezni, ha valaki helyi szoftvert rendel valamilyen célból, úgy az azzal a jövőben majd dolgozók érdekében ellenőrizze, megfelele-e az adott szoftver az ISO 9241 nemzetközi szabvány követelményeinek.

### **Helyreigazítás**

A Hippocrates 2001. szeptember-októberi számában megjelent Prof. Dr. Túri Sándor, Dr. Haszon Ibolya, Dr. Papp Ferenc: Újszülöttkori hypertonia c. cikkében közölt 5. táblázat (330. oldal) címe helyesen:

„Akut hypertenzióban és hypertenzív krízisben alkalmazott iv. gyógyszerek”