

16 FELHASZNÁLÓI FELÜLETEK TERVEZÉSE

TÉMAKÖRÖK

A fejezet célja a felhasználói felületek néhány, a szoftvertervezők számára fontos tervezési szempontjának bemutatása. Mire a fejezet végére érünk:

- meg fogjuk érteni a felhasználói felületek tervezésének alapelveit;
- tisztában leszünk egy szoftverrendszerrel való interakció öt különböző fajtájával, és hogy ezeket mikor a legalkalmasabb használni;
- meg fogjuk érteni, hogy mikor használjunk grafikus, és mikor szöveges megjelenítési módot;
- meg fogjuk ismerni, hogy a felhasználói felület tervezése milyen lényeges tevékenységekből áll;
- meg fogjuk érteni a használhatósági jellemzőket és az interfészek értékelésének különböző megközelítéseit.

TARTALOM

- 16.1. Tervezési kérdések
- 16.2. A felhasználói felületek tervezésének folyamata
- 16.3. A felhasználók elemzése
- 16.4. Felhasználói felület prototípusának elkészítése
- 16.5. A felületek értékelése

A számítógépes rendszerek tervezése a hardvertervezéstől a felhasználói felületek tervezéséig tevékenységek széles spektrumát foglalja magában. Míg a hardvertervezésre gyakran alkalmaznak szakembereket, felhasználói felületet tervező szakembereket csak a nagyobb szervezetek alkalmaznak. Éppen ezért a felhasználói felület tervezéséért is gyakran a szoftvertervezőknek kell vállalniuk a felelősséget ugyanúgy, mint a felületet megvalósító szoftver tervezéséért.

Még ha a szoftvertervezők és programozók jártasak is a felületek implementációjához használt technológiákban – például a Java Swing osztályaiban (Elliott és mások, 2002) vagy az XHTML-ben (Musciano és Kennedy, 2002) –, az általuk fejlesztett felhasználói felület gyakran nem túl vonzó, és alkalmatlan a célközönség megnyerésére. Éppen ezért itt a felhasználói felületek tervezésének folyamatára koncentrálok, az ezeket a lehetőségeket megvalósító szoftverek helyett. Terjedel-

mi korlátok miatt csupán a grafikus felhasználói felületek tárgyalására szorítkozok, és nem írok olyan interfészekről, amelyek speciális (akár nagyon egyszerű) megjelenítést igényelnek, mint például a mobiltelefonok, DVD-lejátszók, televíziók, fénymásoló és faxgépek. Természetesen itt csak a téma bemutatását tudom elvégezni, bővebb információkat a felhasználói felületek tervezéséről Dix és mások (Dix és mások, 2004), Weiss (Weiss, 2002) és Shneiderman (Shneiderman, 1998) könyveiben olvashatunk.

A gondos felhasználói felület tervezés fontos része a teljes szoftvertervezési folyamatnak. Ahhoz, hogy a szoftverrendszer teljesen kihasználhassa lehetőségeit, lényeges, hogy a kialakítandó felhasználói felület illeszkedjen a reménybeli felhasználó szakértelméhez, tapasztalatához és elvárásaihoz. A jó felhasználói felület terve kritikus a rendszer megbízhatósága szempontjából. Sok úgynevezett „felhasználói hibát” az okoz, hogy a felhasználói felületek nem veszik figyelembe a valódi felhasználók képességeit és munkakörnyezetét. Egy rosszul tervezett felhasználói felület azt eredményezheti, hogy a felhasználó esetleg nem fér hozzá a rendszer bizonyos jellemzőihez, hibázik, és úgy érzi, hogy a rendszer segítségnyújtás helyett inkább gátolja őt annak a célnak az elérésében, amiért a rendszert használja.

A felhasználói felületek tervezési döntései során figyelembe kell venniük a szoftvert használó emberek fizikai és mentális képességeit. Terjedelmi okokból itt humán kérdéseket nem tárgyalok részletesen, de az alábbi tényezőket fontosságuk miatt célszerű figyelembe venni

1. Az emberek korlátozott rövid távú memóriával rendelkeznek – kb. hét információelemre tudunk visszaemlékezni (Miller, 1957). Ezért, ha egyidejűleg túl sok információval terheljük meg a felhasználót, elképzelhető, hogy nem tudja azokat befogadni.
2. Mindannyian hibázunk, különösen ha túl sok információt kell kezelnünk vagy nagy nyomás alatt vagyunk. Ha a rendszer elromlik, és figyelmeztető, valamint hibaüzeneteket ad, akkor a felhasználók általában feszültebbé válnak, ami megnöveli a hibázás esélyét.
3. Fizikai képességeink is igencsak eltérhetnek. Némelyek jobban látnak és hallanak másoknál, míg mások színvakok, megint másoknak pedig jobb a kéz ügyességük. Nem szabad saját képességeinkre tervezni, és feltételezni, hogy majd minden felhasználó képes lesz megbirkózni a felülettel.
4. Különböző interakciós módokat részesítünk előnyben. Némelyek képekkel, mások szövegekkel szeretnek dolgozni. Bizonyos embereknek a közvetlen manipuláció a természetes, mások pedig az interakció olyan módját részesítik előnyben, ahol parancsokat adhatnak ki a rendszernek.

A 16.1. ábrán látható tervezési elvek alapját az emberi képességek jelentik. Ezek olyan általános elvek, amelyek minden felhasználói felület tervére alkalmazhatók, és ezekből speciális szervezetek vagy rendszertípusok számára részletesebb tervezési irányelvek származtathatók. A felhasználói felületek terve-

Elv	Leírás
Felhasználói jártasság	A felületnek olyan kifejezéseket és fogalmakat kell használnia, amelyek megfelelnek a rendszert legtöbbet használó emberek tapasztalatainak.
Konzisztencia	A felületnek konzisztensnek kell lennie, azaz lehetőség szerint a hasonló műveleteket hasonló módon kell realizálnia.
Minimális meglepetés	A rendszer soha ne okozzon meglepetést a felhasználóknak.
Visszaállíthatóság	A felületnek rendelkeznie kell olyan mechanizmusokkal, amelyek lehetővé teszik a felhasználók számára a hiba után történő visszaállítást.
Felhasználói útmutatás	A felületnek hiba bekövetkezése esetén értelmes visszacsatolást kell biztosítani és környezetérzékeny felhasználói súgóval is rendelkeznie kell.
Felhasználói sokféleség	A felületnek megfelelő interakciós lehetőségekkel kell rendelkeznie a rendszer különféle felhasználói számára.

16.1. ábra. A felhasználói felületek tervezésének elvei

zésének alapelveit Dix és mások (Dix és mások, 2004) részletezik. Shneiderman (Shneiderman, 1998) egy hosszabb felsorolást adott meg a speciálisabb felhasználói felületek tervezési irányelveiről.

A *felhasználói jártasság* elve azt mondja ki, hogy nem szabad egy felületet csak azért ráerőltetni a felhasználókra, mert az kényelmesen implementálható. A felületnek olyan kifejezéseket kell használnia, amelyeket a felhasználó ismer, és a rendszer által kezelt objektumokat közvetlenül a felhasználó környezetével kell kapcsolatba hozni. Például egy légiforgalom-irányító rendszer által kezelt objektumok a repülőgépek, repülési útvonalak, jelzőfények stb. lehetnek. Az ezekkel kapcsolatos műveletek a repülőgép sebességének növelése, illetve csökkentése, a repülési irány beállítása és a repülési magasság megváltoztatása lehetnek. A felület állományokban és adatszerkezetekben megadott implementációjának a végfelhasználó elől rejtve kell maradnia.

A *felhasználói felület konzisztenciájának* elve azt jelenti, hogy a rendszer parancsainak és menüinek ugyanazzal a formátummal kell rendelkezniük, a paramétereket minden parancshoz ugyanúgy kell átadni, és hasonlóknak kell lennie a parancsok formájának. A konzisztens felületek lerövidítik a felhasználók betanulási idejét. Az egy parancs vagy alkalmazás esetén megtanult ismeret a rendszer más részeiben is alkalmazható.

A felületek alkalmazások közötti konzisztenciája is nagyon lényeges. Amennyire csak lehet, a különböző alkalmazások hasonló értelemmel bíró parancsait ugyanolyan módon kell kifejezni. Hibák általában akkor következnek be, amikor ugyanazon billentyűparancs (például a „Control-b”) különböző rendszerekben különböző dolgokat jelent. Például az általam használt szövegszerkesztőben a „Control-b” a szöveg megvastagítására szolgál, míg az ábrák létrehozására használt grafikai programban a „Control-b” a kiválasztott objektum más objektum

mögé történő mozgatását jelenti. Ha együtt használjuk őket, hibákat követünk el, és a billentyűkombinációval gyakran az ábrák szövegét próbáljuk megvastagítani. Ez a szöveg eltűnését okozhatja, és ez nagyon zavaró lehet. Az ablakozórendszerek szabványai általában definiálják a parancsok gyorsbillentyűit, amelyeket az ilyen hibák elkerülése céljából követni kell.

Ez egy alacsony szintű konzisztencia. Egy felület tervezőinek minden esetben meg kell próbálniuk elérni ezeket egy felhasználói felület esetén. Néha egy magasabb szintű konzisztencia is kívánatos. Például helyénvaló lehet a rendszer entitásainak minden típusán ugyanazon műveletek támogatása (nyomtatás, másolás stb.). Grudin (Grudin, 1989) azonban rámutat, hogy a teljes konzisztencia sem nem lehetséges, sem nem kívánatos. Annak lehet értelme, hogy a törlést az entitások asztalról kukába történő vonzolásával implementáljuk, de nagyon kényelmetlen volna egy szövegszerkesztőben ily módon törölni egy szót.

Sajnos a felhasználói jártasság és a felhasználói felület konzisztenciájának elvei néha összeütközésbe kerülnek. Ideális esetben a közös jellegzetességekkel bíró alkalmazásoknak mindig ugyanazt a parancsot kellene használniuk az adott jellemző eléréséhez, azonban ez olyan esetekben összeütközésbe kerülhet a felhasználói gyakorlattal, amikor egy rendszert bizonyos felhasználói csoportok (például grafikus tervezők) számára terveznek meg. Ezek a felhasználók ugyanis már kialakult interakció-fajtákkal, terminológiával és működési konvenciókkal dolgoznak, amely összeütközésbe kerülhet a szövegszerkesztőkhöz hasonló általánosabb alkalmazások számára kialakult interakciós „szabványokkal”.

A *minimális meglepetés* elvére azért van szükség, mert a felhasználók nagyon ingerültté válnak, ha a rendszer nem várt módon viselkedik. A rendszert használva a felhasználók felépítik magukban a rendszer működésének gondolati modelljét. Ha egy tevékenység az egyik környezetben egy bizonyos változást okoz, akkor ésszerű módon elvárható, hogy ugyanaz a tevékenység egy másik környezetben ezzel összehasonlítható változást eredményezzen. Ha valami teljesen különböző dolog történik, a felhasználó meglepetté és zavarttá válik. A felületek tervezőinek éppen ezért biztosítaniuk kell azt, hogy az összehasonlítható tevékenységeknek összehasonlítható hatásai is legyenek.

A felhasználói felületekben bekövetkező meglepetések gyakran annak köszönhetőek, hogy a felületeket különféle módokra szokás osztani, azaz a munkavégzésnek több módja lehet (például nézegető mód, szerkesztő mód); a parancs hatása pedig attól függ, hogy éppen milyen módban vagyunk. Fontos tehát, hogy a felület tervezésekor gondoljunk arra, hogy a felhasználó számára vizuálisan ki derüljön, hogy éppen milyen módban használja a rendszert.

A *visszaállíthatóság* elve azért fontos, mert a felhasználók a rendszer használata közben elkerülhetetlenül hibáznak. A felület terve minimalizálhatja ugyan ezeket a hibákat (például menük használatával a gépelési hibák elkerülhetők), de azokat soha nem lehet teljesen kiküszöbölni. A felhasználói felületeknek olyan lehetőségeket kell biztosítaniuk, amelyek lehetővé teszik a felhasználók számára hibáik kijavítását. Ezen lehetőségeknek három fajtája lehetséges:

1. *Az ártalmatlan tevékenységek megerősítése.* Ha a felhasználók potenciálisan ártalmatlan tevékenységet hajtanak végre, az információ megsemmisítése előtt megerősítést kell tőlük kérni, hogy tényleg ezt akarják-e tenni.
2. *Visszavonási lehetőség biztosítása.* A visszavonás a rendszert visszaállítja a tevékenység bekövetkezése előtti állapotba. A többszintű visszavonás igen hasznos, mert a felhasználók nem mindig veszik rögtön észre, hogy hiba történt.
3. *Ellenőrző pontok készítése.* Célszerű a rendszer állapotát bizonyos időközönként elmenteni, így lehetővé téve, hogy azt a legutóbbi ellenőrző ponttól újra tudjuk indítani. Így, ha valamilyen hiba következett be, a felhasználók visszamehetnek az előző állapotra, és újra kezdenek a tevékenységet. Sok rendszer ezzel a módszerrel küzd a rendszerhibák ellen, de paradox módon nem teszik lehetővé a felhasználóknak, hogy ugyanezzel a technikával saját hibájukból helyreállhassanak.

Ehhez kapcsolódik a *felhasználó támogatásának* elve. A felületeknek beépített, felhasználót segítő vagy súgórendszerrel kell rendelkezniük. Ezeknek be kell épülniük a rendszerbe, és különböző szintű segítséget és tanácsot kell adniuk. A szintek a rendszerrel való kapcsolatba kerülésről szóló alapinformációktól egészen a rendszer lehetőségeinek teljes leírásáig terjedhetnek. A súgót strukturálni kell, és nem szabad a felhasználókat túlságosan elárasztani információkkal, amikor segítséget kérnek.

A *felhasználói sokféleség* elve azt ismeri fel, hogy az interaktív rendszereknek különböző típusú felhasználói lehetnek. Néhány közülük csak alkalmi felhasználó, akik csak egyszer-egyszer lépnek kapcsolatba a rendszerrel, míg a többiek olyan rendszeres felhasználók, akik naponta több órán keresztül használják a rendszert. Az eseti felhasználóknak olyan felületre van szükségük, amely útmutatót ad, míg a rendszeres felhasználóknak olyan gyorsbillentyűkre van szükségük, amelyek segítségével a lehető leggyorsabb interakciót tudják elvégezni. Ezenfelül a felhasználók különféle fogyatékoságokban szenvedhetnek, és lehetőség szerint a felületnek meg kell ezzel birkóznia. Ezért szükség lehet nagyított szöveg megjelenítésére, a hangok szöveggel történő helyettesítésére, nagyon nagy gombok létrehozására stb. Ebben a Universal Design (UD) eszméje köszön vissza (Preiser és Ostoff, 2001), annak a tervezési elképzelésnek, amelynek célja, hogy elkerüljük a meggondolatlan tervezési döntések következményeként kizárt felhasználókat.

A felhasználói sokféleség felismerésének elve összeütközésbe kerülhet a felület tervezésének egyéb elveivel, mert bizonyos felhasználók a gyors interakciót előnyben részesíthetik, mondjuk, a felhasználói felület konzisztenciájával szemben. Hasonlóan a különböző felhasználók egymástól akár nagyon eltérő szintű felhasználói útmutatót igényelhetnek, így lehetetlenné válhat a minden felhasználónak megfelelő támogatás kialakítása. A felület tervezőjének elkerülhetetlenül kompromisszumok segítségével kell összeegyeztetni a felhasználói igényeket.

16.1. TERVEZÉSI KÉRDÉSEK

Mielőtt továbbhaladnánk a felhasználói felület tervezési folyamatának a tárgyalásával, felvetek néhány általános tervezési kérdést, amelyeket a felhasználói felületek tervezőinek figyelembe kell venniük. Lényegében a tervező két fontos kérdéssel találja magát szembe:

1. Hogyan lép a felhasználó kapcsolatba a számítógépes rendszerrel?
2. Hogyan kell a számítógépes rendszer információit a felhasználó számára tálni?

Egy koherens felhasználói felületnek integrálnia kell a felhasználói interakciót és az információmegjelenítést. Ez azért nem egyszerű, mert a tervezőnek meg kell találnia a kompromisszumot a legmegfelelőbb kapcsolatfelvételi mód, az alkalmazás megjelenése, a rendszer felhasználóinak háttere és tapasztalata, valamint a rendelkezésre álló eszközök között.

16.1.1. Felhasználói interakciók

A felhasználói interakciók a számítógépes rendszernek adott parancsokat és az ezekhez kapcsolódó adatokat jelentik. A korai számítógépeken ennek egyetlen módja egy parancssoros felület volt, amelyen keresztül a géppel egy speciális célú nyelv segítségével lehetett kommunikálni. Ezt azonban csak szakértők tudták használni, és számos más, könnyebben használható módszer alakult ki. Shneiderman (Shneiderman, 1998) az interakció különböző formáit öt nagy csoportba osztotta:

1. *Közvetlen manipuláció*, ahol a felhasználó közvetlenül a képernyő objektumaira hat. Ez általában egy mutatóeszköz (egér, pálca, vagy érintőképernyők esetén az ujj) segítségével történik, ez jelöli ki a manipulálandó objektumot, és a tevékenységet, ami azt adja meg, hogy mit kell az adott objektummal tenni. Például egy állomány törléséhez a felhasználó rákattinthat az állomány ikonjára, és a kukába vonszolhatja azt.
2. *Menü kiválasztása*, ahol a felhasználó lehetőségek egy listájából (a menüből) választ ki egy parancsot. Gyakran előfordul, hogy ugyanekkor a képernyő egy másik objektuma is ki van választva és a parancs ezen az objektumon operál. Egy állomány törlése ebben a megközelítésben először az állomány kiválasztásával, majd a törlés parancs kiválasztásával zajlik.
3. *Űrlapkitöltés*, ahol a felhasználó egy űrlap mezőit tölti ki. Bizonyos mezőkhöz menük tartozhatnak, az űrlapnak pedig lehetnek tevékenység „gombjai”, melyeket lenyomva, bizonyos tevékenység veszi kezdetét. Egy állományt nagyon mesterkéltnak volna űrlapalapú felület segítségével törölni. Erre az állomány névének kitöltésével és a törlés gomb megnyomásával nyílna lehetőség.

4. *Parancsnyelv*, ahol a felhasználó speciális parancsokkal és ezekhez tartozó paraméterekkel utasítja a rendszert, hogy az mit tegyen. Egy állomány törléséhez a felhasználó kiadja a törlés parancsot az állomány nevével mint paraméterrel.
5. *Természetes nyelv*, ahol a felhasználó egy természetes nyelvi parancsot ad ki. Ez általában csak egy felület a parancsnyelv felé, a természetes nyelvi utasítás elemzésre kerül, majd rendszerparancsokká fordítódik át. Egy állomány törléséhez ezért a felhasználónak a következőt kell gépelnie: „töröld az xxx nevű állományt”.

A fenti interakció-fajták mindegyikének vannak előnyei és hátrányai is, valamint ezek az alkalmazások és a felhasználók különböző típusaira a legmegfelelőbbek (Shneiderman, 1998). A 16.2. ábra ezen interakció-fajták fő előnyeit és hátrányait foglalja össze, és bemutatja, hogy azok mely alkalmazástípusok esetén használhatók.

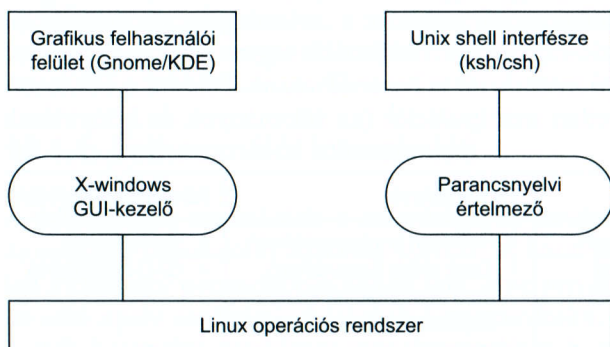
Természetesen az interakció-fajták keverhetők is, és ugyanabban az alkalmazásban több különböző fajtájú interakciót is használhatunk. Például a Microsoft Windows támogatja a közvetlen manipulációt (az állományok és könyvtárak

Interakció fajtája	Fő előnyök	Fő hátrányok	Alkalmazási példák
Közvetlen manipuláció	- Gyors és ösztönös interakció - Könnyen tanulható	- Nehezen implementálható - Csak akkor használható, ha a feladatoknak és az objektumoknak van képi metaforájuk	- Videojátékok - CAD-rendszerek
Menü kiválasztása	- Kivédi a felhasználói hibákat - Kevés gépelést igényel	- A tapasztalt felhasználók számára lassú - Bonyolulttá válhat, ha sok a választási lehetőség	A legtöbb általános célú rendszer
Űrlapkitöltés	Egyszerű adatbevitel	- Sok helyet foglal a képernyőn - Problémákat okoz, ha a felhasználó lehetőségei nem felelnek meg az űrlap mezőinek	- Raktárkészlet-felügyelet - Személyi kölcsön feldolgozása
Parancsnyelv	Erőteljes és rugalmas	- Nehezen tanulható - Gyenge hibakezelés	- Operációs rendszerek - Parancsvezérlő rendszerek
Természetes nyelv	- Az eseti felhasználók számára is elérhető - Könnyen kiterjeszthető	- Több gépelést igényel - A természetes nyelvet megértő rendszerek megbízhatatlanok	Információ-vissza-kereső rendszerek

16.2. ábra. Az interakció-fajták előnyei és hátrányai

ikon formájában történő ábrázolásával), a menüalapú parancskiválasztást, és bizonyos parancsok esetén (mint amilyenek a konfigurációs parancsok) a felhasználónak egy speciális célú űrlapot kell kitöltenie.

Elviekben az interakció fajtája és az általa a felhasználói felületen keresztül manipulált egyedek különválaszthatók. Ezen alapult a felhasználói felületek kezelésének Seeheim-modellje (Pfaff és ten Hagen, 1985). Ebben a modellben az információ megjelenítése, a párbeszédkezelés és az alkalmazás elkülönül egymástól. Valójában ez a modell sokkal inkább ideális, mint praktikus, de biztosan lehetőségünk van arra, hogy a felhasználók különböző csoportjai (mondjuk eseti és tapasztalt felhasználók) eltérő felhasználói felülettel rendelkezzenek, amelyek ugyanazon alrendszerrel tartják a kapcsolatot. Ezt mutatja be a 16.3. ábra, amelyen egy operációs rendszer (például Linux) parancsnyelvi és grafikus interfésze látható.

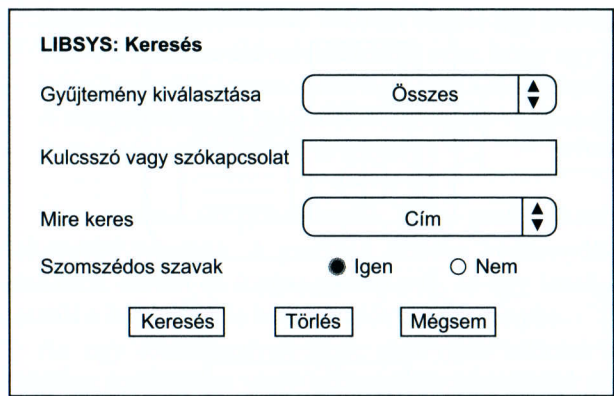


16.3. ábra. Többféle felhasználói felület

A webalapú felhasználói felületek a HTML-en és XHTML-en (a weblapok oldalainak leírására szolgáló nyelvek), valamint olyan, a Javához hasonló nyelveken alapulnak, amelyek egy lap komponenseihez programokat tudnak társítani. Mivel ezeket a webes felületeket gyakran az eseti felhasználóknak tervezik, ezek jobbra űrlalapú felületek. Van lehetőség közvetlen manipulációt végző webes felületek megadására is, de ez bonyolult programozási feladat. Mivel tapasztalt webfelhasználóból sok van, és nagyon sokféle kultúrából jönnek, ezért nehéz megalkotni egy általánosan elfogadható felhasználói felület metaforát a közvetlen interakcióra vonatkozóan.

A webalapú felhasználói interakciók tervezését a LIBSYS-rendszeren keresztül mutatom be. Ebben az esetben a felhasználók más könyvtárak dokumentumaihoz is hozzáférnek. Két alapvető műveletre lesz szükségünk:

1. *Dokumentumkeresés*, amikor a felhasználók a keresési lehetőségek segítségével próbálják megtalálni a számukra szükséges dokumentumokat.
2. *Dokumentum lekérése*, amikor a felhasználó egy adott dokumentumnak a saját gépére történő eljuttatását kéri annak érdekében, hogy ki tudja nyomtatni.



LIBSYS: Keresés

Gyűjtemény kiválasztása

Kulcsszó vagy szókapcsolat

Mire keres

Szomszédos szavak ☒ Igen ☐ Nem

16.4. ábra. A LIBSYS-rendszer űrlapalapú felülete

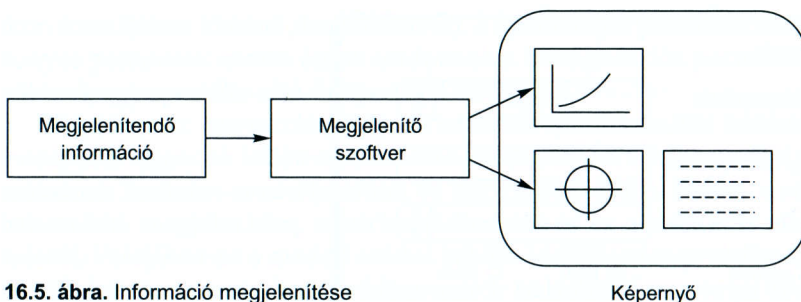
A LIBSYS felhasználói felületét egy webböngésző segítségével valósítjuk meg, így hát, mivel adott, hogy a felhasználóknak bizonyos információkat a rendszer rendelkezésére kell bocsátaniuk (ilyen a keresett dokumentum azonosítója, a felhasználó neve és azonosítási adatai), ezért leginkább egy űrlapalapú interfész használatának van értelme. A 16.4. ábrán a rendszer kereső komponensének egy lehetséges felhasználói felületterve látható.

Az űrlapalapú felületek esetén a felhasználó megadja az összes szükséges adatot, majd a tevékenységet egy gomb lenyomásával indítja el. Az űrlap mezői lehetnek menük, szövegmezők vagy rádiógombok is. Az ábrán látható példában a felhasználó az elérhető gyűjtemények menüjéből kiválasztja az általa keresni kívánt gyűjteményt (az alapértelmezés az „Összes”, amely az összes gyűjtemény keresését jelenti), begépezi a kereső kifejezést a szövegmezőbe, majd kiválasztja a könyvtári rekord egy mezőjét (itt a „Cím” az alapértelmezés), majd egy rádiógomb segítségével kiválasztja, hogy a keresési kifejezéseknek a rekordban szomszédos szavaknak kell-e lenniük.

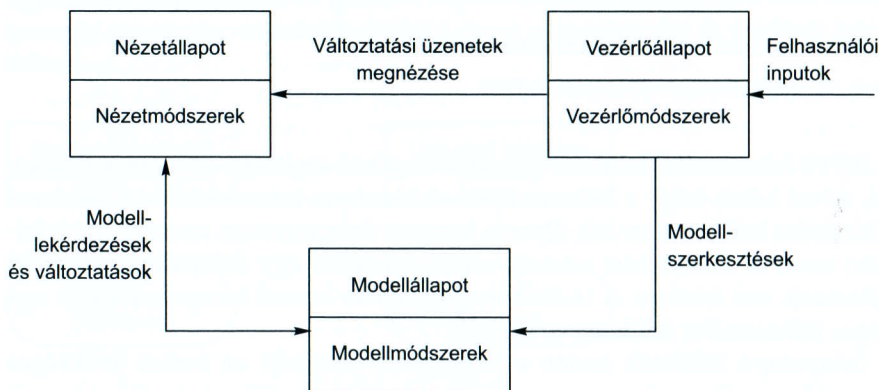
16.1.2. Információmegjelenítés

Minden interaktív rendszernek valamilyen módon biztosítani kell az információ felhasználó számára történő megjelenítését. Az információmegjelenítés lehet egyszerűen a bejövő információ közvetlen reprezentációja (például szöveg egy szövegszerkesztőben), vagy történhet grafikusán is. A jó rendszertervezési gyakorlat szerint az információ megjelenítését végző szoftvert magától az információtól külön kell választani. A megjelenítés adatoktól történő elválasztásával a felhasználó képernyőjén látható megjelenítés anélkül megváltoztatható, hogy a megfelelő számítási rendszert le kellene cserélni. Ez a 16.5. ábrán látható.

Az MVC szemlélet (16.6. ábra), amely először a Smalltalkban (Goldberg és Robson, 1983) jelent meg, hatékonyan támogatja az adatok többféle megjelenítését. A felhasználók az egyes megjelenítésekkel egy annak megfelelő stílus segítség-



16.5. ábra. Információ megjelenítése



16.6. ábra. A felhasználói interakciók MVC-modellje

gével léphetnek interakcióba. A megjelenítendő adat egy modellobjektumba van bezárva. Minden modellobjektumhoz számos nézetobjektum kapcsolódhat, ahol az egyes nézetek a modell különböző megjelenési formái.

Minden nézethez tartozik egy, a felhasználói bemenetet és az eszközökkel való kapcsolattartást kezelő vezérlőobjektum. Éppen ezért egy számadatot reprezentáló modell rendelkezhet olyan nézettel, amely az adatot hisztogramként ábrázolja, és egy olyanmal, amely az adatot táblázatként jeleníti meg. A modell a táblázatban lévő értékek megváltoztatásával, illetve a hisztogram sávjainak meghosszabbításával, illetve lerövidítésével szerkeszthető. Ezt részletesebben a 18. fejezetben tárgyalom, ahol bemutatom az Observer mintának az MVC-keretrendszer implementálásában betöltött szerepét.

Az információ legjobb megjelenítési módjának megtalálásához ismerni kell az információ felhasználói háttérét és azt, hogy milyen módon használják a rendszert. Az információmegjelenítés mikéntjének eldöntésekor számos tényezőt figyelembe kell vennünk:

1. A felhasználót a pontos információk vagy a különböző adatértékek között fennálló relációk érdeklik?
2. Milyen gyorsan változnak az értékek? Kell-e azonnal jelezni egy érték megváltozását a felhasználó felé?

3. Kell-e a felhasználónak valamit tennie egy információ megváltozása esetén?
4. Van-e a felhasználónak szüksége arra, hogy egy közvetlen manipulációs felületen keresztül kapcsolatba kerüljön a megjelenített információval?
5. A megjelenítendő információ szöveges vagy numerikus? Lényegesek-e az információelemek relatív értékei?

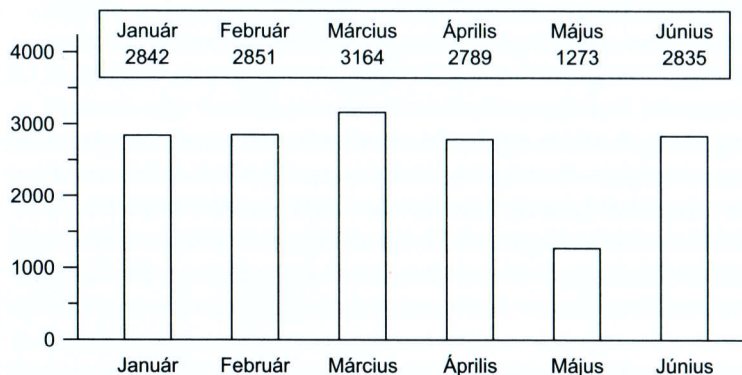
Nem szabad azt gondolnunk, hogy grafika használatával a megjelenítést érdekesebbé tehetjük. A grafikák értékes képernyőhelyet foglalnak (hordozható eszközök esetén ez fontos szempont), és egy lassú, betárcsázós kapcsolaton keresztül a letöltésük is hosszú időt vehet igénybe.

Az egy munkamenet ideje alatt nem változó információ az alkalmazástól függően grafikusán vagy szövegesen jeleníthető meg. A szöveges megjelenítés kevesebb területet foglal el a képernyőből, de nem lehet egy futó pillantást rávetve elolvasni. A nem változó információt megjelenítési stílusában is meg kell különböztetni a dinamikus információtól. Például a statikus információk egyéni betűtípussal jeleníthetők meg, vagy bizonyos színnel kihangsúlyozhatók, esetleg mindig ugyanaz a „statikus információ” ikon tartozhat hozzájuk.

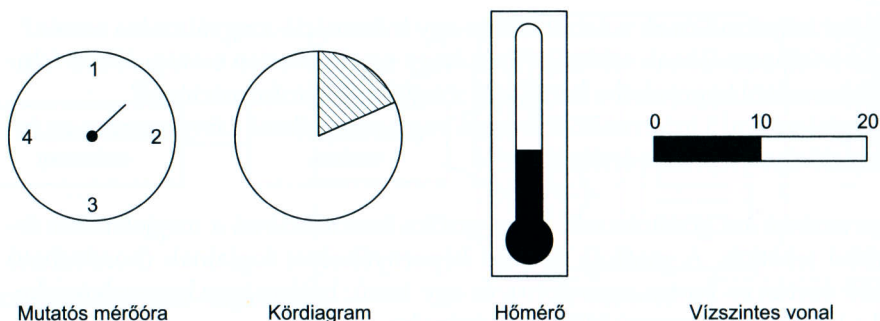
Ha pontos numerikus információra van szükség, vagy az információ viszonylag lassan változik, akkor szöveges megjelenítést érdemes választani. Ha az adat gyorsan változó, vagy az adatok közötti kapcsolatok lényegesek, akkor általában grafikus megjelenítést célszerű alkalmazni.

Példaként tekintsük egy cég eladási adatait havonként rögzítő és összegző rendszerét! A 16.7. ábrán látható ugyanannak az információnak a szöveges és grafikus megjelenítése is. Az eladási adatokat tanulmányozó menedzsereket általában a tendenciái és az esetleges visszasságok érdeklik, nem pedig a pontos értékek. Az információ hisztogramként történő grafikus megjelenítésével szembeötlővé válnak a márciusi és májusi, rendhagyó számok. A 16.7. ábrán az is látható, hogy ugyanannak az információnak a szöveges megjelenítése kevesebb helyet foglal, mint grafikus ábrázolásmódja.

Vezérlőtermekben vagy kapcsolótáblákon – idetartozik például az autó műszerfala is – a megjelenítendő információ egy másik rendszer állapotát írja le



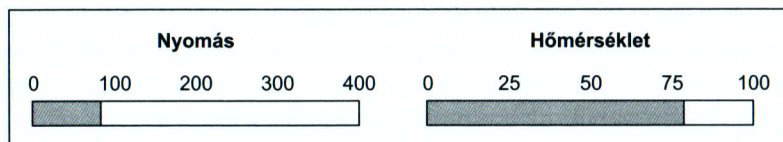
16.7. ábra. Alternatív információmegjelenítési módok



16.8. ábra. Dinamikusan változó numerikus információk megjelenítési módszerei

(például egy repülőgép magassága), és állandó változásban van. Az állandóan változó digitális megjelenítés zavaró, mert a pontos információt nehéz gyorsan elolvasni és feldolgozni. Az ilyen dinamikusan változó numerikus információkat általában legjobb grafikus, analóg ábrázolásmóddal megjeleníteni. A grafikus kijelzőt szükség esetén ki lehet egészíteni egy pontos, digitális kijelzővel. A dinamikus numerikus információk megjelenítésére a 16.8. ábrán láthatunk példákat.

A folytonos analóg kijelzők relatív értékek kifejezésére képesek. A 16.9. ábrán látható hőmérséklet- és nyomásértékek megközelítőleg azonosak, a grafikus kijelzőn azonban az látható, hogy a hőmérséklet közel van a maximumértékhez, míg a nyomás maximális értékének 25 százalékát sem éri el. Ha kizárólag csak a digitális értéket jelenítenénk meg, akkor a leolvasónak kellene ismernie a lehetséges maximumértékeket. A leolvasott érték relatív állapotát gondolatban kell kiszámítani. A gondolkodáshoz szükséges többletidő feszült helyzetekben, amikor probléma lép fel, és az operátor kijelzője rendellenes értéket mutat, emberi hibákhoz vezethet.



16.9. ábra. Relatív értékeket mutató grafikus információmegjelenítés

Nagy mennyiségű információ megjelenítésekor az összetartozó adatelemeket összekapcsoló absztrakt ábrázolás lehetséges. Ez olyan kapcsolatokra mutatható rá, amelyek a nyers adatokból nem nyilvánvalóak. A felhasználói felületek tervezőinek tisztában kell lenniük a megjelenítési lehetőségekkel, főként akkor, ha a rendszer felhasználói felületének fizikai entitásokat kell ábrázolnia. Példák adatmegjelenítésre:

1. Több forrásból összegyűjtött meteorológiai információk egy légnyomásértékeket, frontokat stb. ábrázoló meteorológiai térképen jeleníthetők meg.

2. Egy telefonhálózat állapota grafikus módon, a telefonközpont összekapcsolt csomópontjaival jeleníthető meg.
3. Egy vegyi üzem állapotát a tartályokban és csövekben uralkodó nyomás és hőmérséklet segítségével lehet megadni.
4. Egy molekulamodell három dimenzióban egy virtuálisvalóság-rendszer segítségével jeleníthető meg és kezelhető.
5. Weblapok egy csoportja hiperbolikus faként jeleníthető meg (Lamping és mások, 1995).

Shneiderman (Shneiderman, 1998) jó áttekintést nyújt a különféle megjelenítési szemléletekről és azonosítja a megjelenítés különböző csoportjait. Ezek ma-gukban foglalják az adatok 2 és 3 dimenziós, illetve faként és gráfokként történő megjelenítését. Ezek legtöbbje nagy mennyiségű, számítógépen kezelt információk megjelenítésével foglalkozik, azonban a felhasználói felületekben a megjelenítést leggyakrabban valamilyen fizikai szerkezetnek (például egy új gyógyszer molekuláris szerkezetének, egy telekommunikációs hálózat kapcsolatainak stb.) az ábrázolására használják. A termékek megjelenítésében különösen hatékonyak a speciális virtuálisvalóság-felszereléseket használó háromdimenziós eszközök. Ezen megjelenítések közvetlen manipulációja nagyon hatékony módja az adatokkal való együttműködésnek.

Az információ megjelenítési stílusán túlmenően gondosan kell eljárunk a színek megválasztása esetén is. A színek segíthetnek a felhasználóknak a bonyolultság megértésében és kezelésében, így javítják a felhasználói felületeket. Könnyen lehet azonban a színeket rosszul is használni, és így vizuálisan taszító és hibalehetőségeket magában rejtő felhasználói felületeket létrehozni. Shneiderman a felhasználói felületek hatékony színhasználatára vonatkozólag 14 fő irányelvet fogalmaz meg. Ezek közül a legfontosabbak:

1. *Korlátozzuk az alkalmazott színek számát és konzervatív módon használjuk azokat.* Nem szabad egy ablakban négy vagy öt, egy rendszer teljes felületében pedig hét színnél többet használni. Ha túl sok, avagy túl világos színeket használunk, a képernyő zavaró lehet. Néhány felhasználó a túlzottan sok szín használatát nyugtalanítónak és vizuálisan fárasztónak találhatja. Ha pedig a színeket következetlenül alkalmazzuk, még össze is zavarhatjuk a felhasználót.
2. *A rendszer állapotváltozását jelöljük színváltoztatással.* Egy kijelző színváltoztatásának egy fontos esemény bekövetkezését kell jelentenie. Így egy üzemanyagmérce esetén színváltoztatás jelezheti, ha fogytán van az üzemanyag. A színekkel történő kiemelés különösen fontos olyan bonyolult kijelzők esetén, ahol különböző entitások százait kell megjeleníteni.
3. *A felhasználók által végrehajtandó feladatokat lássuk el színkódokkal.* Ha a felhasználóknak a rendhagyó eseteket kell felismerniük, emeljük ki ezeket az eseteket; ha a hasonlóságokat is észre kell venniük, akkor azokat eltérő színnel emeljük ki!
4. *A színkódolást alkalmazzuk gondosan és következetesen.* Ha a rendszer egyik része, mondjuk, pirossal jeleníti meg a hibaüzeneteket, akkor a többi résznek is

hasonlóan kell tennie. A piros színt nem szabad semmi másra használni! Ha mégis, akkor lehet, hogy a felhasználó a piros megjelenését hibaüzenetként fogja értelmezni. Tisztában kell lennünk azzal, hogy a felhasználók egyes csoportjai feltételezéssel élhetnek bizonyos színek jelentését illetően.

5. *Óvatosnak kell lennünk a színek összepárosításával.* A szem élettani jellemzői miatt az emberek nem tudnak egyszerre a pirosra és a kékre fókuszálni. Ha valamit kék háttérrel pirossal jelenítünk meg, annak valószínű következménye a szem megerőltetése. Más színkombinációk szintén lehetnek zavarók és nehezen olvashatók.

Általánosságban azt mondhatjuk, hogy bár használhatjuk a színeket kiemelésre, de nem szabad az egyes színekhez jelentést rendelni. A férfiak körülbelül 10 százaléka színtévesztő, ezért félreértelmezhetné ezt a jelentést. Az emberek színérzékelései között eltérések vannak, és a különböző szakmákban különféle konvenciók vonatkoznak bizonyos színek jelentésére. A különböző háttérrel rendelkező felhasználók ugyanazt a színt öntudatlanul is eltérő módon értelmezhetik. Például egy sofőrnek a piros általában *veszélyt*, míg egy vegyésznek *forróságot* jelent.

Az alkalmazás adatainak megjelenítése mellett a rendszerek a felhasználókat üzenetek segítségével informálják az esetleges hibákról és a rendszer állapotról. A rendszer hibaüzenetei a felhasználók szoftverrendszerről alkotott első benyomását jelenthetik. A tapasztalatlan felhasználók munkához látnak, rögtön az elején elkövetnek egy hibát, és azonnal fel kell fogniuk az így kapott hibaüzenetet. Ez még a gyakorlott szoftvertervezőknek sem feltétlenül egyszerű dolog, tapasztalatlan, illetőleg eseti felhasználók esetén pedig gyakran lehetetlen. A rendszerüzenetek tervezésekor figyelembe veendő tényezők a 16.10. ábrán láthatók.

A felhasználók háttérére és tapasztalatára a hibaüzenetek tervezésénél rá kell érezni. Legyen mondjuk a rendszer felhasználója egy kórház intenzív osztályának egyik kórtermében dolgozó nővér. A betegfigyelést egy számítógépes rendszer végzi. Egy beteg aktuális állapotának (pulzus, testhőmérséklet stb.) megtekintéséhez a felhasználó egy menüből kiválasztja a „megjelenítés” menüpontot, és a 16.11. ábrán látható módon begépel a beteg nevét.

Tegyük fel, hogy ebben az esetben a beteg neve nem Kiss, hanem Kis, így a nővér által begépelte nevet nem lehet felismerni. A rendszer egy hibaüzenetet készít. A hibaüzeneteknek mindig udvariasnak, tömörnek, következetesnek és építő jellegűnek kell lenniük. Nem szabad gyalázkodónak lenniük, és nem szabad hozzájuk sípoló vagy egyéb hangoknak tartozniuk, amelyek zavarhatják a felhasználót. Lehetőség szerint az üzenetnek javaslatot kell tennie, hogy hogyan lehetne a hibát elhárítani. A hibaüzenetnek kapcsolódnia kell egy környezetérzékeny online súgórendszerhez.

A 16.12. ábrán jó és rossz hibaüzenetekre egyaránt láthatunk példát. Az ábra felső részében látható üzenet rosszul tervezett. Negatív szemléletű (hiba elkövetésével vádolja a felhasználót), nem a felhasználó szakképzettségéhez és ta-

Tényező	Leírás
Környezet	Ahol csak lehet, a rendszer által generált üzeneteknek az aktuális felhasználó környezetét kell tükrözniük. Ameddig csak lehet, a rendszernek tisztában kell lennie azzal, hogy mit csinál a felhasználó, és csak az aktuális tevékenység szempontjából lényeges üzeneteket kell előállítania.
Tapasztalat	Ahogy a felhasználók egyre jártasabbak lesznek egy rendszerben, a hosszú, „értelmes” üzenetek egyre inkább bosszantani fogják őket. A kezdők azonban nehezen képesek megérteni a probléma rövid megfogalmazását. A felhasználói útmutató rendszernek mindkét üzenetfajtát biztosítani kell, és lehetővé kell tennie a felhasználó számára, hogy az üzenetek tömörségét szabályozza.
Szakképzettség szintje	Az üzeneteket a felhasználók szakképzettségéhez ugyanúgy hozzá kell igazítani, mint tapasztalataikhoz. A különböző csoportba tartozó felhasználók üzenetei különböző módon adhatók meg attól függően, hogy az olvasó milyen terminológiában jártas.
Stílus	Az üzeneteknek inkább pozitívnak kell lenniük, mintsem negatívnak. Inkább aktív, mintsem passzív nyelvezetet kell használnia. Soha nem szabad sem sértőnek, sem pedig viccesnek lenniük.
Kultúra	Amennyiben lehetséges, az üzenetek tervezőjének jártasnak kell lennie annak az országnak a kultúrájában, amelyben a rendszer eladásra kerül. Határozott kulturális különbségek vannak Európa, Ázsia és Amerika között. Az egyik kultúrkörben megfelelő üzenet egy másikban akár elfogadhatatlan is lehet.

16.10. ábra. Az üzenet szövegezésének tervezési tényezői

Kérem adja meg a beteg nevét,
és utána kattintson az OK gombra

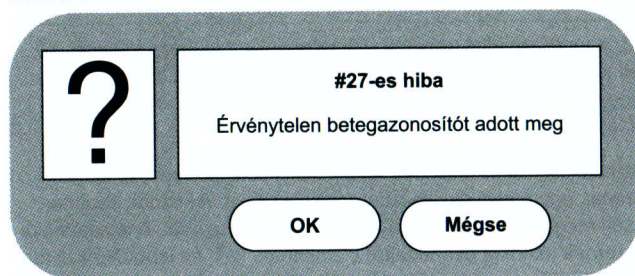
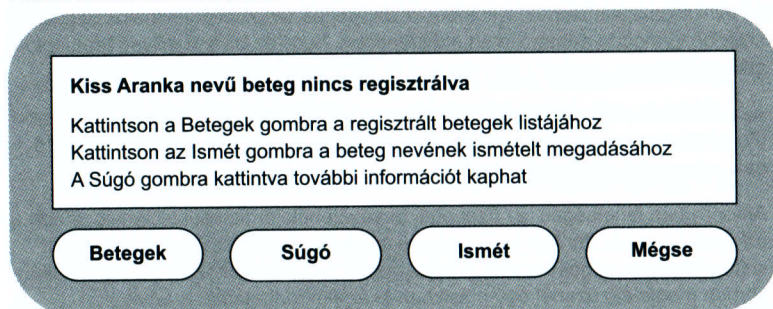
Beteg neve

Kiss Aranka

OK Mégse

16.11. ábra. Beteg nevének bekérése a nővértől

pasztalatához van igazítva, és nem veszi figyelembe a környezetre vonatkozó információt. Nem tesz arra vonatkozólag javaslatot, hogy miképpen lehetne a hibát orvosolni. Felhasználóorientált nyelv helyett rendszerspecifikus kifejezéseket (betegazonosító) használ. Az alsó üzenet jobb. Pozitív szemléletű, arra utal, hogy a probléma nem a felhasználó, hanem inkább a rendszer gondja. A problémát a nővérek nyelvén fogalmazza meg, és a hiba kijavítására könnyen, egyetlen gombnyomással elvégezhető módszert javasol. Szükség esetén a súgó is elérhető.

Rendszerorientált hibaüzenet**Felhasználóorientált hibaüzenet**

16.12. ábra. Rendszer- és felhasználóorientált hibaüzenetek

16.2. A FELHASZNÁLÓI FELÜLETEK TERVEZÉSÉNEK FOLYAMATA

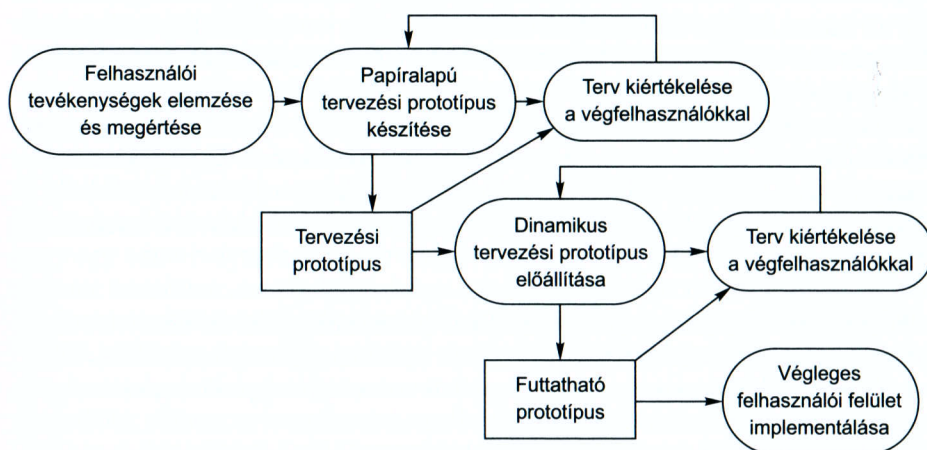
A felhasználói felületek tervezése olyan iteratív folyamat, ahol a felhasználók a tervezőkkel együttműködve, prototípusok segítségével döntenek a rendszer felhasználói felületének jellemzőiről, felépítéséről és kinézetéről. A felület prototípusa különállóan is elkészíthető, párhuzamosan más szoftvertervezési tevékenységekkel. Különösen iteratív fejlesztés esetén az is előfordulhat, hogy a felhasználói felület tervének fejlődése a szoftver fejlesztésével együtt inkrementális módon zajlik. Bármelyik megoldást alkalmazzuk is, a programozás megkezdése előtt papíralapú terveket kell készítenünk, amelyeket ideális esetben le is tesztlünk.

A felhasználói felület tervezésének teljes folyamata a 16.13. ábrán látható. A folyamat három alapvető tevékenységből áll:

1. *Felhasználók elemzése.* A felhasználók elemzése során meg kell vizsgálnunk, hogy ők milyen feladatokat végeznek, milyen munkakörnyezetben dolgoz-

nak, milyen egyéb rendszereket használnak, munkájuk során hogyan lépnek kapcsolatba más emberekkel stb. Ha nagyon eltérő környezetben működő felhasználókra számítunk, akkor ezt a vizsgálatot ki kell terjeszteni a különféle csoportokra, például potenciális felhasználók meghallgatásával.

2. *Prototípus-készítés.* A felhasználói felület tervezése és fejlesztése iteratív folyamat. Habár a felhasználók elmondhatják, hogy milyennek szeretnék látni a felületet, milyen lehetőségeket várnak el tőle, amíg nem látnak valami kézzelfogható, addig nehezen tudják részletezni a dolgokat. Éppen ezért prototípusokat kell készítenünk, amelyeket megmutatunk a felhasználóknak, akik ezek után irányíthatják a felület fejlődését.
3. *A felületek értékelése.* Annak ellenére, hogy a prototípus-készítés során már nyilvánvalóan tárgyaltunk a felhasználókkal, egy formálisabb értékelést is el kell végeznünk, melynek során a felhasználók interfészhasználat közben szerzett aktuális tapasztalatait gyűjtjük össze.



16.13. ábra. A felhasználói felület tervezésének folyamata

Ebben az alfejezetben a hangsúlyt a felhasználók elemzésére és a felületek értékelésére helyeztem, a prototípus-készítési technikákat csak röviden említem meg. A prototipizálás általános kérdéseiről és prototípus-készítési technikákról részletesebben a 17. fejezetben lesz szó.

A felhasználói felület tervezésének a szoftverkészítés folyamatán belüli időzí-
tése egy bizonyos fokig más tevékenységektől is függ. Ahogyan arról a 7. fejezet-
ben már tettem említést, a prototípus-készítés része lehet a követelményelemzési
folyamatnak is, így ebben az esetben célszerű a felhasználói felület tervezésének
folyamatát is ebben a szakaszban beindítani. A 17. fejezetben tárgyalt iteratív fo-
lyamatokban a felület tervezését a szoftverfejlesztéssel integráljuk, így magához
a szoftverhez hasonlóan a felhasználói felület is átstrukturálható és újratervezhe-
tő a fejlesztés során.

16.3. A FELHASZNÁLÓK ELEMZÉSE

A felhasználói felület tervezésének egyik kritikus pontja azon felhasználói tevékenységek elemzése, amelyeket a számítógépes rendszernek támogatnia kell. Anélkül, hogy megértenénk, a felhasználó mit akar csinálni a rendszerrel, nincs reális esélyünk hatékony felhasználói felület létrehozására. Ahhoz, hogy ezt megértsük, alkalmazhatjuk a feladatelemzés, az etnográfiai tanulmánykészítés, a felhasználói interjúk és megfigyelések technikáit vagy ezek tetszőleges keverékét.

A felhasználók elemzésébe bevont tervezők számára az igazi kihívás az, hogy megtalálják a módját annak, hogy úgy írják le a felhasználói elemzéseket, hogy utána az alapján a feladatok lényegét mind a többi tervező, mind pedig maguk a felhasználók felé kommunikálni tudják. Az UML szekvenciadiagramjai képesek a felhasználói interakciók leírására, és ideálisak a szoftvertervezőkkel való kapcsolattartásra. Azonban lehet, hogy a felhasználók ezeket a diagramokat túlságosan technikainak gondolják, és ezért meg sem próbálják megérteni azokat. Mivel nagyon fontos, hogy a felhasználók részt vegyenek a tervezési folyamatban, ezért a felhasználói tevékenységek leírására általában természetes nyelvi forgatókönyveket kell készíteni.

A 16.14. ábrán egy ilyen forgatókönyv látható, amely a LIBSYS-rendszer specifikációjának és tervezési folyamatának idején készülhetett. Egy olyan esetet ír le, amelyben a LIBSYS nem létezik, egy hallgatónak egy másik könyvtárból kell információt lekérnie. Ebből a forgatókönyvből a tervező az alábbi követelményeket tudja kiolvasni:

1. A felhasználók nem ismerik az alkalmas keresőkifejezéseket. Ennek kiválasztásához segítségre szorulhatnak.
2. Lehetőséget kell biztosítani a felhasználók számára, hogy válasszanak a gyűjtemények közül.
3. A felhasználóknak képesnek kell lenniük keresések végrehajtására és a tárgyhoz tartozó anyag másolatainak lekérésére.

Jane hittant hallgat, és egy esszét kell írnia arról, hogy miképpen befolyásolta az indiai építészetet a vallásgyakorlás. Ehhez szüksége volna a nevezetes épületek részletgazdag képeire, de a helyi könyvtárban ilyet nem talált. Elmondta az igényeit a könyvtárosnak, aki javaslatot tett arra, hogy milyen keresőkifejezést lenne érdemes használni, illetve javasolt egy újdolhi és egy londoni könyvtárat, ahol ez az anyag feltehetően meg is található. Jane belépett a könyvtár gyűjteményébe, és elvégezte a keresést. Talált némi forrásanyagot, és kérelmezte, hogy a képek másolatát, építészeti részletekkel együtt, postázzák el címére.

16.14. ábra. Egy könyvtár-együttműködési forgatókönyv

Nem szabad a felhasználók elemzésétől azt várni, hogy általa nagyon részletes felhasználói felület követelményekhez jutunk. Ez valójában arra szolgál, hogy megértsük az igényeket és a rendszer felhasználóinak érdekeit. Azáltal, hogy megismerjük, hogyan dolgoznak ők, milyen szempontjaik, milyen korláto-

zásaik vannak, a tervezés során ezt figyelembe tudjuk venni. Mindez pedig azt eredményezi, hogy a kezdeti terveink (amelyeket persze a prototípus-készítés során majd tovább finomítunk) nagyobb eséllyel lesznek számukra elfogadhatók, ezáltal meg lesznek győzve arról, hogy részt vegyenek a terv finomításának folyamatában is.

16.3.1. Elemzési technikák

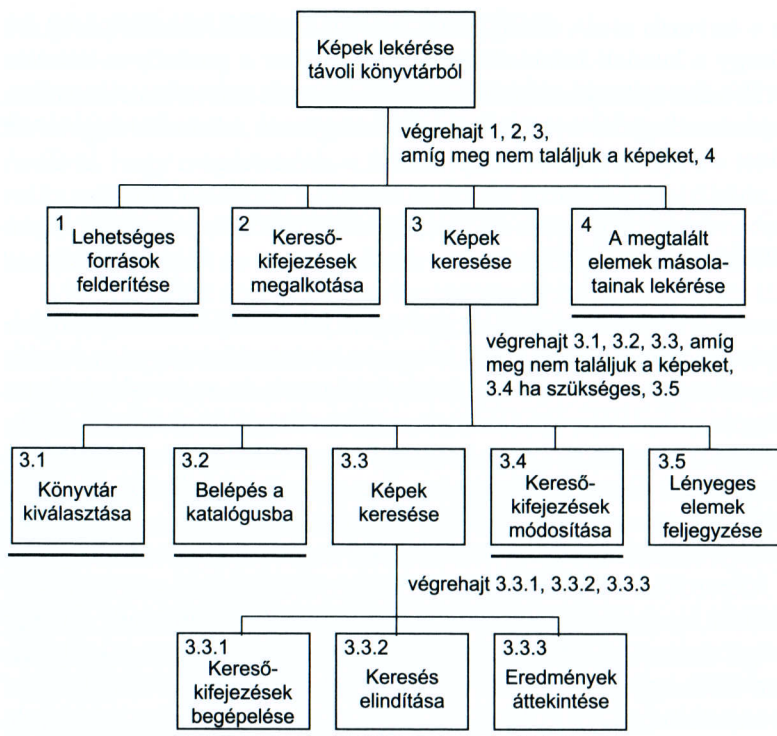
Ahogy az az előző alfejezetben már tárgyaltam, a felhasználók elemzésére három alapvető lehetőségünk van: a feladatelemzés, az interjúztatás vagy kérdőívek kitöltetése, valamint az etnográfia. A feladatelemzés és az interjúztatás az egyénre és az egyén munkájára összpontosít, míg az etnográfia szélesebb szemszögből vizsgálja azt, hogy az emberek hogyan kommunikálnak és hatnak egymásra, hogyan rendezik el munkakörülményeiket, és miként működnek együtt feladatok megoldásában.

A feladatelemzésnek (Diaper, 1989) különböző formái vannak, de legszélesebb körben a hierarchikus feladatelemzést (Hierarchical Task Analysis, HTA) használják. A HTA-t eredetileg a felhasználói kézikönyvek készítésének elősegítésére hozták létre, de használható annak meghatározására is, hogy mit kell a felhasználóknak tenniük egy bizonyos cél elérése érdekében. A HTA-ban a magas szintű feladatot részfeladatokra bontjuk, és terveket készítünk, hogy megadjuk, hogy egy adott helyzetben mi történhet. A felhasználó céljából kiindulva egy hierarchiát készítünk, amely leírja, hogy mit kell tennünk a cél érdekében. A 16.15. ábrán ez a szemlélet látható a 16.14. ábra könyvtári forgatókönyvét felhasználva. A HTA jelölésrendszerében a doboz alatti vonal azt jelöli, hogy az adott feladatot nem bontjuk részfeladatokra.

A HTA előnye a természetes nyelvi forgatókönyvekkel szemben, hogy rákényszerít bennünket arra, hogy minden feladatot átgondoljunk, és eldöntsük, hogy felbontható-e, avagy sem. A természetes nyelvi forgatókönyvek használata során könnyen kimaradhatnak fontos feladatok, emellett a forgatókönyvek hosszúvá és unalmassá válhatnak, ha sok részletet szeretnénk beleírni.

Ezen megközelítés hibája az, hogy leginkább csak a szekvenciális folyamatokként leírható feladatokra alkalmazható. A jelölésmód kényelmetlenné válik, ha konkurens tevékenységeket vagy sok részfeladatot tartalmazó feladatokat szeretnénk modellezni. Mindemellett a HTA nem jegyzi fel, hogy a feladatokat miért pont úgy kell végrehajtani, és nincs lehetőségünk a felhasználói folyamatokra vonatkozó megszorítások előírására sem. A HTA segítségével a felhasználó tevékenységeinek egy részleges áttekintését kapjuk, de a felhasználói felületek követelményeinek teljes megértéséhez további információkra van szükségünk.

A HTA számára általában a felhasználók megfigyelésével és a velük készített interjúk segítségével gyűjtünk információkat. Az interjúztatás során gyűjtünk némi további információt, amit a feladatelemzés elvégzése mellett szintén feljegyzünk. Amikor interjúztatással szeretnénk megtudni, hogy a felhasználók valójában mit is csinálnak, akkor ezeket az interjúkat úgy kell megtervezni, hogy



16.15. ábra. Hierarchikus feladatelemzés

a felhasználók minden, általuk (nem pedig általunk!) szükségesnek tartott információt közölhessenek. Ez azt jelenti, hogy nem szabad mereven ragaszkodnunk az előre elkészített kérdésekhez, hanem a kérdéseknek nyitottnak kell lenniük, és arra kell serkenteniük a felhasználókat, hogy elmondják, mit és miért tesznek.

Persze nem az interjúk készítése az egyetlen módja a feladatelemzés számára történő adatgyűjtésnek – ez egy általános információgyűjtési módszer. Az egyéni interjúkat kiegészíthetjük csoportos interjúkkal és fókuszcsoportokkal. Utóbbiak előnye, hogy a felhasználók egymást serkentik arra, hogy információt adjanak, és ez teret ad a rendszerek használata során gyűjtött különböző tapasztalatok kiértékelésének is.

A feladatelemzés középpontjában az áll, hogy az egyén hogyan dolgozik, de természetesen ez a munka valójában kooperatív. Az emberek együtt dolgoznak egy cél elérésének érdekében, és a felhasználók nehezen tudnak arról beszélni, hogy ez az együttműködés valójában hogyan is zajlik. Éppen ezért a felhasználók elemzésének fontos kiegészítő technikájaként szükség van annak közvetlen megfigyelésére, hogy a felhasználók hogyan dolgoznak és használnak számítógépes rendszereket.

A közvetlen megfigyelés egyik széles körben használt eszköze az etnográfia (Suchman, 1983; Hughes és mások, 1997; Crabtree, 2003). Az etnográfiát a 7. feje-

zetben már tárgyaltam mint a követelményelemzés egyik technikáját. Az etnográfusok közelről figyelik, hogy az emberek hogyan dolgoznak, hogyan lépnek kapcsolatba egymással, hogyan hatnak egymásra, és hogy a munkahely jellemzői hogyan segítik munkájukat. Ennek előnye, hogy így intuitív tevékenységeket és informális együttműködések is meg tudnak figyelni, amelyek a munkáról szóló további beszélgetéseket indíthat be.

Annak bemutatására, hogy az etnográfia hogyan képes befolyásolni a felhasználói felületek tervezését, a 16.16. ábra egy olyan, légi irányítókról szóló etnográfiai tanulmány jelentésének részletét tartalmazza, amelynek elkészítésében magam is részt vettem (Bentley és mások, 1992). Egy jobban automatizált légi irányító rendszer felületének tervezését végeztük, és ezekből a megfigyelésekből két fontos dolgot tanultunk:

1. A légi irányítóknak képesnek kell lenniük arra, hogy a szektor minden járatát lássák (ezért terítették szét a szalagokat az asztalon). Ezért nem szabad olyan görgethető felületet biztosítani számukra, amelyben a képernyő tetejéről vagy aljáról eltűnnek a járatok.
2. A felületnek valamilyen módon közölnie kell a légi irányítókkal, hogy hány járat van a szomszédos szektorokban, mert így tudnak terheléelosztást végezni.

A szomszédos szektorok ellenőrzését a légi irányítók automatikusan végzik, így nagyon valószínű, hogy a légi irányítás folyamatának átbeszélése során sem említették volna, így ezt a fontos követelményt csak közvetlen megfigyelés segítségével deríthettük fel.

Ezen felhasználói elemzési technikák egyike sem nyújt önmagában teljes képet arról, hogy a felhasználók mit csinálnak. Ezek olyan egymást kiegészítő megközelítések, amelyeket együtt kell használnunk azért, hogy megértsük, hogy a felhasználók mit tesznek, és képet kapjuk arról, hogy milyen felület lenne a legmegfelelőbb.

A légi irányítás során számos „vezérlőszekrényt” használnak. A légtér szomszédos szektorainak vezérlőszekrényei fizikailag egymás mellett helyezkednek el. A szektorban lévő járatokat fa tartókeretbe helyezett papírszalagok jelképezik, a szektorban elfoglalt pozíciójukat tükröző sorrendben. Ha a tartókeretben nincs elég hely (azaz a légtérben nagyon sokan vannak), a légi irányítók a szalagokat a tartókeret mellett lévő asztalra helyezik. Miközben megfigyeltük a légi irányítókat, észrevettük, hogy rendszeresen rápillantanak a szomszédos szektor szalagjaira. Megkérdeztük, hogy ezt miért teszik, és azt válaszolták, hogy ha a szomszédos légi irányítóknak vannak szalagjai az asztalon, az azt jelenti, hogy az ő szektorukba sok járat fog belépni. Ezért a jelenleg ott tartózkodó járatok sebességét megpróbálják növelni annak érdekében, hogy így „helyet csináljanak” a szektorban a belépő repülőgépek számára.

16.16. ábra. Jelentés a légi irányítók tevékenységének megfigyeléseiről

16.4. FELHASZNÁLÓI FELÜLET PROTOTÍPUSÁNAK ELKÉSZÍTÉSE

A felhasználói felületek dinamikus természete miatt a felületekkel szemben támasztott követelmények kifejezésére a szöveges leírások és a diagramok nem elég jók. A szoftverrendszerek grafikus felhasználói felületének tervezésének és fejlesztésének egyetlen praktikus módja a végfelhasználó bevonásával végzett prototípus-készítés. A felhasználók tervezési és fejlesztési folyamatba történő bevonása az interaktív rendszerek tervezési filozófiájának, a *felhasználó-központú tervezésnek* (Norman és Draper, 1986) egyik lényeges pontja.

A prototípus-készítés célja lehetővé tenni a felhasználóknak, hogy közvetlen tapasztalatokat szerezzenek a felülettel kapcsolatban. Legtöbbünknek nehézséget okoz absztrakt módon gondolkodni egy felhasználói felületről, és pontosan elmagyarázni, hogy mit szeretnénk. Azonban ha látunk egy példát, akkor könnyen meg tudjuk mondani, hogy mely jellemzőket szeretjük, és melyeket nem.

Ideális esetben a prototípus-készítés során a folyamatot két lépésben hajtjuk végre:

1. A folyamat elején papíralapú prototípusokat – a képernyőtervek makettjeit – készítünk, és ezeket áttekintjük a végfelhasználókkal.
2. Majd finomítjuk a tervet, és egyre kifinomultabb automatizált prototípusokat fejlesztünk, amelyeket tesztelésre és a tevékenységek szimulálása érdekében odaadunk a felhasználóknak.

A papíralapú prototípus-készítés igen olcsó és meglepően hatékony módja a prototípus-fejlesztésnek (Snyder, 2003). Nem kell futtatható programot készítenünk, és a terveket sem kell profi módon megrajzolni. Csupán a rendszer azon képernyőit kell papírra lerajzolni, amelyeken kapcsolatba lép a rendszerrel. Emellett forgatókönyveket is létrehozunk, amelyek azt tartalmazzák, hogy a rendszer miképpen használható. A forgatókönyv bővülésével mindig fel tudjuk vázolni a megjelenítendő információkat, és választási lehetőségeket is kínálhatunk a felhasználóknak.

A forgatókönyvek alapján egy szimuláció során át tudjuk tekinteni a felhasználókkal együtt, hogy a rendszert hogyan lehet használni. Ez hatékony módja annak, hogy megkapjuk a felhasználók első reakcióit a felület tervére, a számukra szükséges információkra, valamint a rendszerrel való kapcsolattartásukra vonatkozóan.

Emellett egy másik technikát is alkalmazhatunk a felület tervének bemutatására: a forgatókönyveket. Egy *forgatókönyv* egy interakció-sorozatot bemutató jelenetsor. Ez kevésbé kézenfekvő, de ha nem egyéneknek, hanem csoportoknak kell bemutatni a felületet, akkor alkalmasabb lehet.

A papíralapú prototípus kezdeti tapasztalatai után a felület tervének szoftveres prototípusát kell implementálnunk. A probléma persze ott van, hogy szükség van némi rendszer-funkcionalitásra, amit a felhasználók használhatnának. Ha ezt a prototípus-készítést a rendszerfejlesztés korai szakaszában végezzük, akkor

ez esetleg nem áll rendelkezésre. Ezen probléma kiküszöbölésére az „Óz, a nagy varázsló” prototípus-készítési technikát használhatjuk (ha nem látta a filmet, magyarázatért látogassa meg a weboldalt). Ekkor a felhasználók egy számítógépes rendszernek látszó dologgal lépnek kapcsolatba, de valójában bemeneteik egy rejtett személyhez jutnak el, aki szimulálja a rendszer válaszait. Ez vagy közvetlenül, vagy a kívánt válaszok valamely más rendszer használatával történő kiszámításával történhet. Ebben az esetben a bemutatott felhasználói felületen kívül nincs szükség lefuttatható szoftverre.

Ezt követően további prototipizálást végezhetünk az evolúciós, illetve az eldobható megközelítés alkalmazásával. Ezeket a 17. fejezetben mutatom be, ahol a prototípus-készítéshez és a gyors alkalmazásfejlesztéshez használható technikákat is tárgyalom. A felhasználói felületek prototípusának elkészítésére az alábbi három lehetőségünk van:

1. *Szkriptvezérelt megközelítés.* Ha egyszerűen csak gondolatokat szeretnénk bemutatni a felhasználóknak, egy olyan szkriptvezérelt megközelítést alkalmazhatunk, mint amilyen a Macromedia Directorban is van. Ekkor létrehozunk a vizuális elemeket (gombokat, menüket stb.) tartalmazó képernyőket, és ezekhez szkripteket társítunk. A felhasználói interakció során a szkript végrehajtásra kerül, és megjelenik a következő képernyő, amely a korábbi tevékenységek eredményeit mutatja be. Ekkor nem hajtunk végre alkalmazáslogikát.
2. *Vizuális programozási nyelvek.* Az olyan vizuális nyelvek, mint amilyen például a Visual Basic, erőteljes fejlesztési környezetet, számos újrafelhasználható objektumot és egy felhasználói felületet fejlesztő rendszert tartalmaznak, amelynek segítségével gyorsan létrehozhatjuk a felületet, és annak objektumaihoz komponenseket és szkripteket társíthatunk. A vizuális fejlesztőrendszerek leírása a 17. fejezetben olvasható.
3. *Internetalapú prototípus-készítés.* Ezek a megoldások egy webböngészőn és egy nyelven (például Java) alapulnak. A megjelenítendő információhoz funkcionalitást Java-kódrészletekkel rendelhetünk. Ezek a kódrészletek (appletek) az oldal böngészőbe töltésekor automatikusan végrehajtnak. Bár ez a módszer lehetővé teszi a felhasználói felület prototípusának gyors elkészítését, de a böngésző és a Java biztonsági modellje miatt bizonyos korlátozásokat is tartalmaz.

A prototípus-készítés nyilvánvalóan szoros kapcsolatban áll a felületek értékelésével. A korai prototípusok esetén a formális értékelés valószínűleg nem költséghatékony, így ebben a szakaszban csak egy „alakító értékelést” kell végoznünk, amikor azt vizsgáljuk, hogy hogyan javítható a felület. Ahogyan a prototípus egyre teljesebbé válik, alkalmazhatjuk a következő alfejezetben leírt szisztematikus értékelési technikákat is.

16.5. A FELÜLETEK ÉRTÉKELÉSE

A felületek értékelése az a folyamat, amelyben megállapítjuk a felületek használhatóságát, és ellenőrizzük, hogy ezek megfelelnek-e a felhasználói követelményeknek. Éppen ezért ennek a szoftverrendszerek normál verifikációs és validációs folyamata részének kell lennie. Neilsen (Neilsen, 1993) a használhatóság tervezéséről szóló könyvében egy nagyszerű fejezetet szán ennek a témának.

Ideális esetben az értékelés a 16.17. ábrán látható használhatósági jellemzőkön alapuló használhatósági specifikációból levezethető. Ezen használhatósági jellemzők számára metrikákat határozhatunk meg. Például egy tanulhatósági mérték azt mondhatja, hogy az adott munkában járatos operátor egy háromórás oktatást követően a rendszer funkcionalitásának 80 százalékát lesz képes kihasználni. A használhatóságot azonban legtöbbször kvalitatív módon adjuk meg, nem pedig metrikák segítségével (ha egyáltalán megadjuk). Éppen ezért a felületek értékelésében saját véleményeinket és tapasztalatainkat kell használnunk.

Jellemző	Leírás
Tanulhatóság	Mennyi időbe telik egy új felhasználónak, hogy hasznót termeljen a rendszer segítségével?
Műveleti sebesség	A rendszer válaszideje mennyire felel meg a felhasználók munkagyakorlatának?
Robusztusság	Mennyire toleráns a rendszer a felhasználói hibákkal szemben?
Visszaállíthatóság	Mennyire jó a rendszer a felhasználói hibák visszaállításában?
Adaptálhatóság	Mennyire van a rendszer egy egyszerű munkamodellhez kötve?

16.17. ábra. Használhatósági jellemzők

Egy felhasználói felület tervének szisztematikus értékelése a kognitív tudományokkal foglalkozó tudósok és grafikusok bevonásával zajló költséges folyamat lehet. Ez a folyamat magában foglalhatja statisztikai értelemben jelentős számú, megfigyelőfelszerelésekkel ellátott, speciálisan kialakított laboratóriumokban átlagfelhasználókkal elvégzett kísérlet megtervezését és végrehajtását. Ez a fajta felületértékelés a kis szervezetek korlátozott erőforrásaival fejlesztett rendszereinek esetében gazdaságilag kivitelezhetetlen.

A felhasználói felületek értékelésére számos egyszerűbb, kevésbé drága, és a tervezés bizonyos hiányosságainak azonosítására képes technika is létezik:

1. kérdőívek, amelyek arról gyűjtenek információt, hogy a felhasználók mit gondolnak a felületről;
2. a rendszer felhasználóinak munka közben történő megfigyelése és „hangosan gondolkodás” arról, hogyan próbálják a rendszert bizonyos feladatok végrehajtására használni;
3. a jellegzetes rendszerhasználat video „pillanatfelvételei”;

4. olyan kódrészlet szoftverbe történő beépítése, amely a legtöbbször használt lehetőségekről és a leggyakoribb hibákról gyűjt információt.

A felhasználók kérdőív segítségével történő felmérése a felület értékelésének egy viszonylag olcsó módja. A kérdéseknek inkább pontosaknak, semmint általánosaknak kell lenniük. Olyan kérdéseknek semmi haszna, mint például: „Mi a véleménye a felület használhatóságáról?”, mivel a válaszok valószínűleg oly mértékben el fognak térni, hogy abból semmilyen következtetést nem lehet levonni. Ehelyett inkább speciális kérdéseket érdemes feltenni, például az „Értékelje a hibaüzenetek érthetőségét egy 1-től 5-ig terjedő skálán! Az 1-es osztályzat nagyon érthető, az 5-ös pedig érthetetlen üzeneteket jelent”, sokkal jobb. Egyrészt könnyebb megválaszolni, másrészt pedig nagyobb a valószínűsége, hogy a felület fejlesztése szempontjából hasznos információt kapunk.

Kérdőívküldéskor a felhasználóknak saját tapasztalataikra és hátterükre kell támaszkodniuk. Ez lehetővé teszi, hogy a tervező kiszűrje, ha egy bizonyos háttérrel rendelkező felhasználóknak valamilyen problémájuk van a felülettel. Ha a felület rendelkezik papíralapú modellel, és azt értékeljük, akkor a kérdőívek már azelőtt használhatók, hogy a futtatható rendszer elérhető lenne.

A megfigyelésen alapuló értékelés egyszerűen a felhasználók figyelését jelenti, miközben azok használják a rendszert. Meg kell vizsgálni az általuk használt lehetőségeket és az általuk ejtett hibákat, és ezeket ki lehet egészíteni „hangosan gondolkodás” szakaszokkal, ahol a felhasználók arról beszélnek, hogy mit akarnak megvalósítani, hogyan értik a rendszert, és hogyan próbálják meg azt saját céljaik eléréséhez használni.

A viszonylag alacsony költségű videofelszerelés azt jelenti, hogy a felhasználó tevékenységeinek későbbi elemzése céljából történő felvételével közvetlen megfigyelésre van lehetőség. A teljes videoelemzés drága és több, a felhasználót és a képernyőt felvevő kamerából álló, speciálisan felszerelt értékelőkészletre van hozzá szükség. A kiválasztott felhasználói műveletek problémáinak felderítésében azonban igen hasznos tud lenni. Annak kiderítésére, hogy a felhasználóknak mely műveletek okoznak nehézséget, más értékelési módszereket kell használni.

A felvételek elemzésével a tervező megállapíthatja, hogy a felület túlságosan sok kézmozgást igényel-e (néhány rendszerrel probléma, hogy a felhasználóknak rendszeresen a billentyűzetről az egérre kell mozgatni kezüket), és hogy van-e szükség természetellenes szemmozgásra. Egy sok fókuszváltást igénylő felület azt jelentheti, hogy a felhasználó több hibát vét, és figyelmen kívül hagyja a képernyő egyes részeit.

Olyan eszközök kódba építése, amelyek használatra vonatkozó statisztikát készítenek, a felületek sokféle tökéletesítését teszik lehetővé. Felfedezhetők a leggyakoribb műveletek, és a felületet át lehet úgy szervezni, hogy ezeket a lehető leggyorsabban lehessen kiválasztani. Például felbukkanó vagy legördülő menük esetén a leggyakoribb műveleteknek a menük tetején kell lenniük, a ritkábban használtaknak pedig az alja körül. A kódfigyelés lehetőséget biztosít a hibalehetőséget magukban rejtő parancsok felfedezésére és módosítására is.

Végül minden programot elláthatunk egy „panasz” paranccsal, amelynek segítségével a felhasználó üzeneteket küldhet az eszköz tervezőjének. Ez azt az érzetet nyújtja a felhasználónak, hogy véleményét figyelembe veszik. A felület tervezését végző személy így az egyedi problémákról is gyors visszacsatolást kaphat.

Ezeknek a viszonylag egyszerű megközelítéseknek egyike sem „bolond-biztos”, és nem valószínű, hogy a felhasználói felületek összes problémáját felismerné. Az itt leírt technikák azonban egy csoport önkéntes segítségével a rendszer forgalomba hozása előtt nagy erőforrásköltségek nélkül alkalmazhatók. A felhasználói felületek tervezésének legnagyobb problémái közül sok felderíthető és kijavítható.

Kulcsfogalmak

- A felhasználói felületek felhasználói jártasságot, konzisztenciát, minimális meglepetést, helyreállíthatóságot, felhasználói támogatást és felhasználói sokféleséget lefedő alapelvei segítik a felületek tervezését.
- Egy szoftverrendszerrel közvetlen manipuláció segítségével, menürendszerrel, űrlapok kitöltésével, parancsnyelveken és természetes nyelven keresztül lehet interakcióba lépni.
- Grafikus információmegjelenítést akkor kell használni, ha trendeket és megközelítő értékeket kell bemutatni. Digitális megjelenítésre csak akkor van szükség, ha az értékeket pontosan kell megadni.
- A felhasználói felületekben színeket következetesen és alig kell használni. A tervezőknek figyelembe kell venniük azt a tényt, hogy az emberek jelentős része színtévesztő.
- A felhasználói felületek tervezésének folyamata magában foglalja a felhasználók elemzését, a prototípus-készítést és a felületek értékelését.
- A felhasználók elemzésének célja, hogy a tervezők ráérezzenek, hogy a felhasználók hogyan is dolgoznak. A felhasználók elemzése során használható technikák: feladatelemzés, interjúk készítése, megfigyelés.
- A felhasználói felületek prototípusának elkészítése többlépcsős folyamat: a korai prototípusokat papíron készítjük, majd egy kezdeti értékelést követően a visszajelzéseket meghallgatva ezt használjuk az automatizált prototípus készítéséhez.
- A felület értékelésének kettős célja van: visszajelzéseket szeretnénk kapni, hogy hogyan lehetne még a felületen fejleszteni, és meg szeretnénk tudni, hogy a felület megfelel-e a használhatóság követelményeinek.

További irodalom

- Human-Computer Interaction, 3rd ed.* A tervezési kérdésekre és együttes munkára összpontosító, általános, jó leírás. (Dix, A. és mások, 2004, Prentice-Hall.)
- Interaction Design.* Ez a könyv a számítógépes rendszerekkel kapcsolatos interakciók tervezésére koncentrál. Nagyrészt ugyanazt tartalmazza, mint a *Human-Computer Interaction*, csak nagyon eltérő szempontból. Mindkét könyv nagyszerűen van megírva, és megéri őket elolvasni. (Preece, J. és mások, 2002, John Wiley & Sons.)
- „Usability Engineering”. Az *IEEE Software* különkiadása, amely a használhatóságra vonatkozóan több olyan cikket is tartalmaz, amelyeket kimondottan szoftvertervezési háttérismeretekkel rendelkező olvasóknak írtak. (*IEEE Software*, 18(1), January 2001.)

Feladatok

- 16.1. A 16.1. alfejezetben azt tanácsoltam, hogy a felhasználók által manipulált objektumokat saját szakterületükön, nem pedig számítógépes szakterületen fogalmazzuk meg. Javasoljon megfelelő objektumokat a felhasználók és rendszerek következő típusai számára:
- automatizált alkatrész-katalógust használó raktári alkalmazott;
 - a repülőgép biztonságfigyelő rendszerét használó repülőgép-pilóta;
 - pénzügyi adatbázist kezelő menedzser;
 - járórkocsi-vezérlőrendszert használó rendőr.
- 16.2. Soroljon fel olyan eseteket, amikor nem érdemes, vagy nem is lehet, következetes felhasználói felületet létrehozni.
- 16.3. Milyen tényezőket kell az olyan „walk-up” rendszerek menüalapú felületének tervezésekor figyelembe venni, mint amilyen a bankok ATM-rendszere? Írjon egy kritikus hangú feljegyzést az ön által használt ATM felületéről.
- 16.4. Javasoljon módokat egy online könyvesbolthoz vagy egy zenei kiskereskedéshez hasonló e-kereskedelmi rendszer felhasználói felületének gyengén látó, illetve kézzremegéssel rendelkező felhasználók számára történő átdolgozására.
- 16.5. Magyarazza el a grafikus információmegjelenítés előnyeit, és mondjon négy alkalmazást, ahol célszerűbb a numerikus információkat grafikusán, mint szövegesen megjeleníteni.
- 16.6. Milyen irányelveket kell követni színes felhasználói felületek esetében? Adjon tanácsot, hogy hogyan lehetne egy ön által használt alkalmazás felhasználói felületében hatékonyabban használni a színeket.
- 16.7. Tekintsük az MS Windows, a Unix, a MacOS és egyéb operációs rendszerek hibaüzeneteit. Adjon javaslatot ezek fejlesztésére.

- 16.8. Írjon interakciós forgatókönyveket az alábbi rendszerek számára:
- Egy webes színházjegyfoglaló rendszer, amelyben színházjegyek rendelhetők, és hitelkártyával lehet fizetni.
 - Ugyanezen jegyek mobiltelefonos felületen keresztül történő megrendelése.
 - Egy szoftverrendszer objektummodelljének elkészítésére és az így előállt modellből történő kódgenerálásra használt CASE-eszközkészlet (8. és 14. fejezet).
- 16.9. Milyen körülmények között használhatjuk az „Óz, a nagy varázsló” prototípus-készítési technikát?
- 16.10. Tervezzon kérdőívet, amely valamely olyan eszközről (például egy szövegszerkesztőről) gyűjt információt, amelynek használatában ön jártas. Amennyiben lehetséges, ossza szét ezt a kérdőívet több felhasználónak és értékelje az eredményeket. Mit árulnak el ezek a felhasználói felületek tervezéséről?
- 16.11. Fejtse ki véleményét arról, hogy etikus-e a végfelhasználók tájékoztatása nélkül szoftverbe épített figyelést végezni.
- 16.12. Milyen erkölcsi kérdésekkel kell a felhasználói felületek tervezőinek szembenézni akkor, amikor megpróbálják összeegyeztetni a rendszer végfelhasználóinak szükségleteit a rendszer fejlesztéséért fizető szervezetével?