

Térinformatika 7.

**A felhasználói felület értékelése, a
térinformatikai szoftver testre szabása**

Végső, Ferenc

Térinformatika 7.: A felhasználói felület értékelése, a térinformatikai szoftver testre szabása

Végső, Ferenc

Lektor: Detrekői, Ákos

Ez a modul a TÁMOP - 4.1.2-08/1/A-2009-0027 „Tananyagfejlesztéssel a GEO-ért” projekt keretében készült. A projektet az Európai Unió és a Magyar Állam 44 706 488 Ft összegben támogatta.

v 1.0

Publication date 2010

Szerzői jog © 2010 Nyugat-magyarországi Egyetem Geoinformatikai Kar

Kivonat

Ebben a fejezetben olyan kérdésekről tárgyalunk, amelyek a térinformatika megjelenésével, testre szabásával és a térinformatikai adatbázis osztott használatával kapcsolatosak. Bemutatjuk a felhasználói felületek fejlődését, korszerű felhasználói felület részeit. Foglalkozunk a térinformatikai szoftver átalakítási lehetőségeivel, úgymint a felhasználói felület átalakítása, új parancsok megírása és a lekérdezések automatizálása. Részletezzük a térinformatikai adatbázis megosztott használatának feltételeit és megoldását

Jelen szellemi terméket a szerzői jogról szóló 1999. évi LXXVI. törvény védi. Egészének vagy részeinek másolása, felhasználás kizárólag a szerző írásos engedélyével lehetséges.

Tartalom

7. A felhasználói felület értékelése, a térinformatikai szoftver testre szabása	1
1. 7.1 Bevezetés	1
2. 7.2 A felhasználói felület	1
3. 7.3 A felhasználói felület értékelése	10
3.1. 7.3.1 Kijelzési formák	10
3.2. 7.3.2 Fogalmazás	10
3.3. 7.3.3 Színek	11
3.4. 7.3.4 Grafika	11
3.5. 7.3.5 Párbeszéd	11
3.6. 7.3.6 Adatbevitel	12
3.7. 7.3.7 Eszközök	12
3.8. 7.3.8 Hibaüzenetek és segítségadás	12
4. 7.4 A térinformatikai szoftver testre szabása	13
5. 7.5 Térinformatikai adatbázis megosztása	15
5.1. 7.5.1 Az ArcStorm - térbeli adatbázis menedzser	16
6. 7.6 Összefoglalás	17

7. fejezet - A felhasználói felület értékelése, a térinformatikai szoftver testre szabása

1. 7.1 Bevezetés

Az ember – számítógép kapcsolat az idők során sokat fejlődött. Az alfanumerikus megjelenítéstől és a parancsok begépelésétől eljutottunk a mozdulatokkal vezérelhető animált menüig. Ez a fejlődés nem hagyta érintetlenül a térinformatikai szoftvereket sem. A kezdeti különállás után a térinformatikai szoftverek objektum szinten beleágyazódtak a korszerű operációs rendszerek világába. Hasonló folyamaton esett át a térinformatikai szoftverek testre szabása is. Kezdetben csak speciális makro nyelven és csak korlátozottan lehetett hozzáférni az adatokhoz, változókhöz. Ma már a térinformatikai szoftvereket ugyanazon a programnyelven fejlesztik, mint bármely más szoftvert. Ezért akár kód szintjén is belenyúlhatunk az eredeti programba. A munkafolyamatok automatizálásában is segítséget nyújtanak a mai térinformatikai szoftverek.

A hálózatok elterjedésével egyre nagyobb hangsúlyt kap a térinformatikai adatbázisok osztott, egyidejű használatának kérdése. Ez a kérdés már régen megoldott az adatbázis kezelők körében, térinformatikai adatbázis speciális adatai miatt (geometria, topológia) azonban új adatbázis adminisztrációs funkciókat kellett kifejleszteni.

2. 7.2 A felhasználói felület

A felhasználói felület az a része a GIS-nek, amelyen keresztül a felhasználó közli a GIS szoftverrel az utasításait. A felhasználói felület meglehetősen változatos. A legegyszerűbb a **parancsfeldolgozó** megoldás (CLI = Command Line Interface). Ez olyan, mint a pl. a Microsoft DOS, begépeljük a parancsot és az esetleges opcióit amit a parancsértelmező végrehajt (Pc Arc/Info).

A felhasználói felület értékelése, a
térinformatikai szoftver testre
szabása

```
Microsoft Windows [verziószám: 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. Minden jog fenntartva.

C:\Users\GEO>ipconfig all

Hiba: felismerhetetlen vagy hiányos parancssor.

HASZNÁLAT:
    ipconfig [/allcompartments] [/? | /all |
        /renew [adapter] | /release [adapter] |
        /renew6 [adapter] | /release6 [adapter] |
        /flushdns | /displaydns | /registerdns |
        /showclassid adapter |
        /setclassid adapter [osztályazonosító] ]

ahol
    adapter          Kapcsolat neve
                    <a * és a ? helyettesítő karakter használható,
                    lásd a példákat>

    Kapcsolók:
    /?               Megjeleníti a súgóüzenetet
    /all             Teljes konfigurációs adatok megjelenítése.
    /allcompartments Minden rekesz adatainak megjelenítése.
    /release        A megadott adapter IPv4-cinének felszabadítása.
    /release6       A megadott adapter IPv6-cinének felszabadítása.
    /renew          A megadott adapter IPv4-cinének megújítása.
    /renew6         A megadott adapter IPv6-cinének megújítása.
    /flushdns       A DNS-feloldási gyorsítótár törlése.
    /registerdns    Az összes DHCP-bérlet frissítése és a DNS-nevek
                    újregisztrálása.
    /displaydns     A DNS-feloldó gyorsítótára tartalmának megjelenítése.
    /showclassid    Minden, az adapter számára engedélyezett
                    DHCP-osztályazonosító megjelenítése.
    /setclassid     A DHCP-osztályazonosító módosítása.

Az alapértelmezés csak az IP-cím, az alhálózati maszk és az
alapértelmezett átjáró megjelenítése minden, a TCP/IP protokollhoz
kötött adapter esetében.

Ha nincs megadva adapternév a Release és a Renew kapcsolóval, akkor a parancs
felszabadítja vagy megújítja minden, a TCP/IP protokollhoz kötött adapter
címberletét.

Ha nincs megadva osztályazonosító a Setclassid kapcsolóval, akkor törli
az osztályazonosítót.

Példák:
> ipconfig          ... Adatok megjelenítése
> ipconfig /all     ... Részletes adatok megjelenítése
> ipconfig /renew   ... minden adapter megújítása
> ipconfig /renew EL* ... minden olyan kapcsolat megújítása,
                    amelynek EL-lel kezdődik a neve
> ipconfig /release *Con* ... minden megfelelő kapcsolat
                    felszabadítása, pl.
                    "Helyi hálózati kapcsolat 1" vagy
                    "Helyi hálózati kapcsolat 2"
> ipconfig /allcompartments ... Minden rekesz adatainak
                    megjelenítése.
> ipconfig /allcompartments /all ... Részletes adatok minden rekeszről

C:\Users\GEO>
```

1. ábra Parancssoros felhasználói felület

Napjainkban egyeduralmat nyertek a **grafikus felhasználói felületek**.

A grafikus felhasználói felület vagy grafikus felhasználói interfész (angolul graphical user interface, röviden GUI) a számítástechnikában olyan, a számítógép és ember közti kapcsolatot megvalósító elemek összessége, melyek a monitor képernyőjén szöveges és rajzos elemek együtteseként jelennek meg. A grafikus felhasználói felületeken alapvető szerepe van a mutatóeszközök, például az egér használatának, amelyekkel a grafikus felület elemei intuitív módon, a fizikai világ egyfajta modelljeként kezelhetők.¹

Egy köztes típus, de jobban hasonlít a GUI-ra a szöveges felhasználói felület (TUI), mely ugyancsak képi eszközöket használ, de karaktercellás szöveges módban, nem pedig képpont alapú grafikus módban. Egy ismert TUI a Norton Commander fájlkezelő programrendszer, de példaként felhozható sok más DOS alkalmazás is.

¹http://hu.wikipedia.org/wiki/Grafikus_felhasználói_felület

A felhasználói felület értékelése, a
térinformatikai szoftver testre
szabása



2. ábra A Norton Commander TUI felülete

A grafikus felhasználó felületet a Stanford Kutatóintézet kutatói (köztük Douglas Engelbart) találták fel, az On-Line Systemben való felhasználás céljából. Az alapötletet nagyban javította és kiterjesztette a Xerox Palo Alto Kutatóközpontja (röviden Xerox PARC), ahol a Xerox Alto számítógépük elsődleges felületévé tették a GUI-t. A legtöbb modern, általános célú grafikus felületet ebből a rendszerből származtatják. Ebből az okból kifolyólag néhányan az interfész ezen fajtáját PARC User Interface-nek (PUI) hívják. Ez az elnevezés azonban nem terjedt el. Néha a GUI-kat WIMP-eknek nevezik, amely az ablakok, ikonok, menük és mutató eszköz szavak angol megfelelőinek betűszava (Windows, Icons, Menus, Pointing device), de ez az elnevezés szintén nem terjedt el. A grafikus felhasználói felület kifejezés általában a WIMP rendszerekkel szinonim, és a legtöbb modern GUI valóban WIMP.

A GUI-t támogató rendszerekre példák: Mac OS X, Microsoft Windows, Linux, rendszerek. Az olyan grafikus felhasználói felületek, melyek nem PUI-k, általában a számítógépes játékokban találhatók meg, továbbá a továbbfejlesztett, virtuális valóság alapú GUI-k is gyakran felszínre kerülnek a kutatóintézetekben. Sok észak-amerikai és európai kutatócsoport jelenleg a Zooming User Interface-en (ZUI) dolgozik, mely a GUI logikus továbbfejlesztése, amely összevegyíti a 3D-mozgást a 2D vagy „2 és fél D” vektorobjektumokkal.²

A leggyakoribb grafikus felhasználói elemek az ablakok, menük, választógombok, jelölőnégyzetek és ikonok, valamint a mutatóeszközhöz kapcsolódó egérkurzor.

² <http://hu.wikipedia.org>



3. ábra Grafikus felhasználói felület³

A GUI-kat általában a parancssoros felhasználói felületekkel állítják szembe, amelyben a felhasználónak parancsokat vagy karakterláncokat kellett begépelnie ahhoz, hogy a számítógépnek feladatokat adjon. Bár több tanulás árán, számos feladat elvégzésében és automatizálásában a parancssori felületek nagyobb hatékonysággal alkalmazhatók. Ezt bizonyítja a több mint 30 év óta töretlen népszerűségnek örvendő Unix-héj vagy a különböző specializált parancssori adatbázis-lekérdező felületek használata, de az AutoDesk AutoCad szoftvere is a mai napig megtartotta a parancssoros bevételi lehetőséget..

Mivel a GUI-k és a TUI-k általában a legtöbb vagy az összes odaillő parancskategóriát megjelenítik, a felhasználók általában könnyebben tanulják meg azok használatát, mint a CLI-két. Ennek ellenére, mivel a megjelenített lehetőségek a felhasználónak szánva lettek kiírva, és ez általában korlátozottabb, mint az összes lehetőség, ezért a szoftver összes funkcióinak kihasználása sok időbe telhet egy GUI-rendszerben. Egy epés megjegyzés a Mac OS X előtti időkből jól összefoglalja ezt: „harminc perc alatt megtanulhatsz Macintosht használni, de hat hónappal később sem tudsz majd semmi többet a Macintosh használatáról.”⁴

A látási vagy mozgási fogyatékos felhasználóknak általában több problémájuk akad a grafikus felhasználói felületben való navigációval, valamint a legtöbb kereskedelmi GUI jobb számítógépet igényel, mint egy CLI, így ritkábbak a GUI-rendszerek a régebbi hardvereken. A megfelelő felület megtervezése a kézi eszközökre (például PDA-alkalmazások) is nagy probléma a felhasználói felületek tervezőinek.

³<http://dveo.com.cn>

⁴ <http://hu.wikipedia.org>



4. ábra A iPhone grafikus felhasználói felülete⁵

Emiatt elterjedt megoldás a grafikus felület összekötése a hangvezérléssel (például képernyő-felolvasó programok gyengén látóknak, hangtárcsázás a mobiltelefonokban; a Microsoft Windows Vista beépítetten tartalmaz hangfelismerő programot). Az OS2/Warp fejlesztői kísérleteztek egy teljesen hangalapú felülettel, ez azonban az operációs rendszerrel együtt véget ért. Napjainkban különféle kutatóközpontok, a NASA-val és a Microsofttal az élen próbálkoznak a létrehozásával, a hangalapú felhasználói felület azonban mind a mai napig nem valósult meg. A felhasználói felületek fejlődésének a számítógépes játékok adtak nagy lökést. Megjelentek a mozgásvezérlésű felhasználói rendszerek, mint a Wii. A Wii a Nintendo által gyártott videojáték-konzol. A konzol korábbi kódneve Revolution (Forradalom) volt.

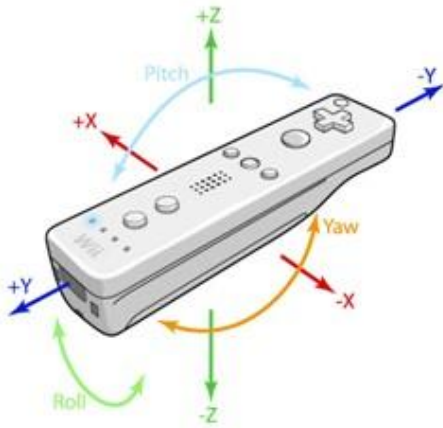


Különleges tulajdonsága, hogy vezeték nélküli távirányító a játék vezérlője, melyet Wii Remote-nak neveznek. Használatakor ezt az eszközt kézben kell tartani és a játék eseményének megfelelően 3D-s mozgásokat lehet vele végezni. A konzol másik sajátossága, hogy WiiConnect24 kiegészítő szolgáltatás tartozik hozzá, ami lehetővé teszi, hogy üzeneteket fogadjunk, és aktuális frissítéseket töltsünk le a konzolra az internet segítségével.⁶

A WiiMote egy egy-kezes vezérlőegység, ami egy 3D-s gyorsulásmérővel és infravörös érzékelővel lett felszerelve. Ez a funkció lehetővé teszi a játékos fizikai mozgásának figyelembevételét a játék közben. A vezérlő egység Bluetooth kapcsolaton keresztül érintkezik a konzollal. A távirányítóhoz tartozik még csuklópánt is annak érdekében, hogy megfelelően kézben tudjuk tartani a vezérlő egységet.

⁵ <http://iphonews.com>

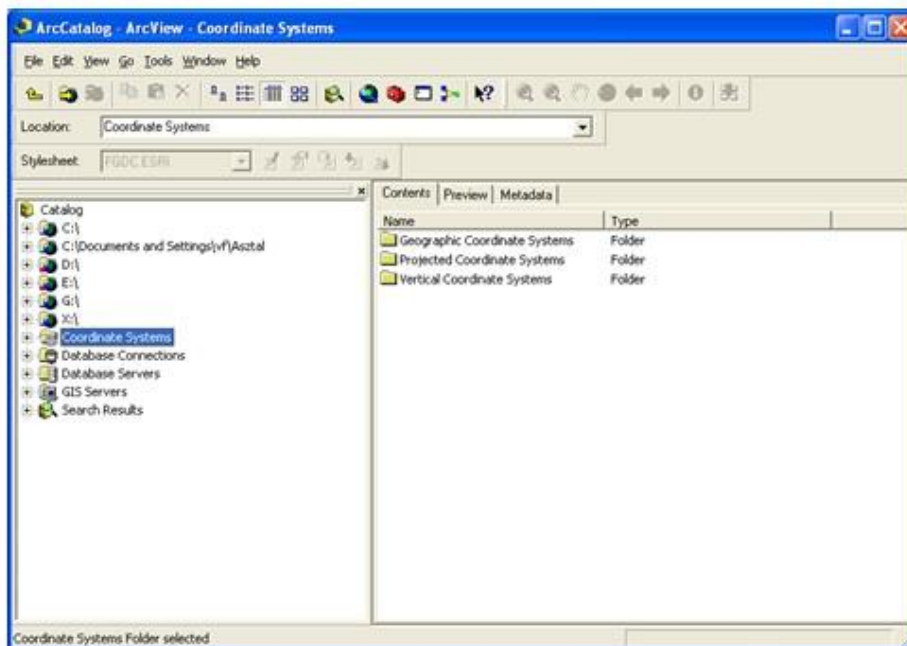
⁶ <http://hu.wikipedia.org>



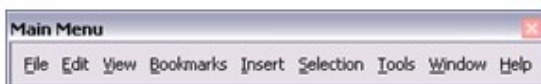
A megfelelő felhasználói felület kiválasztása a szoftver ergonómia feladata. Ezzel a kérdéssel a későbbiekben részletesen foglalkozunk.

A továbbiakban egy térinformatikai szoftveren – ESRI ArcMap – keresztül fogjuk bemutatni a grafikus felhasználói felület fő részeit.

- Osztott ablak

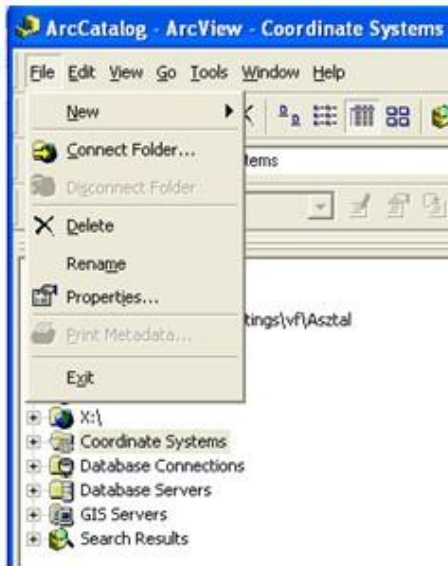


- Főmenü, menü

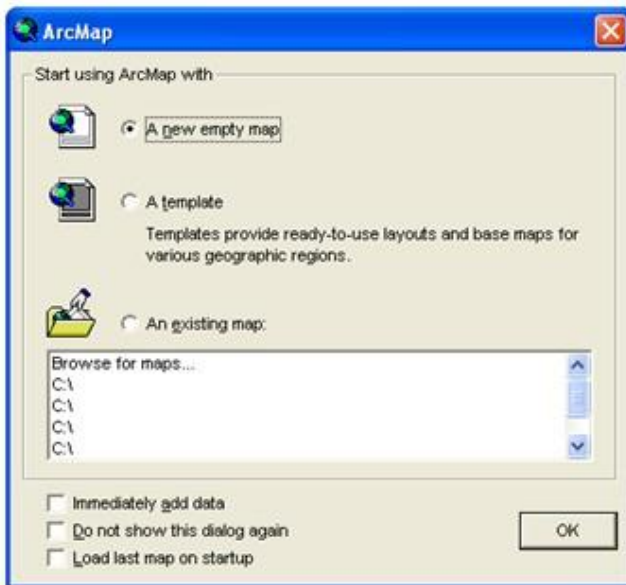


- Legördülő menü

A felhasználói felület értékelése, a
térinformatikai szoftver testre
szabása

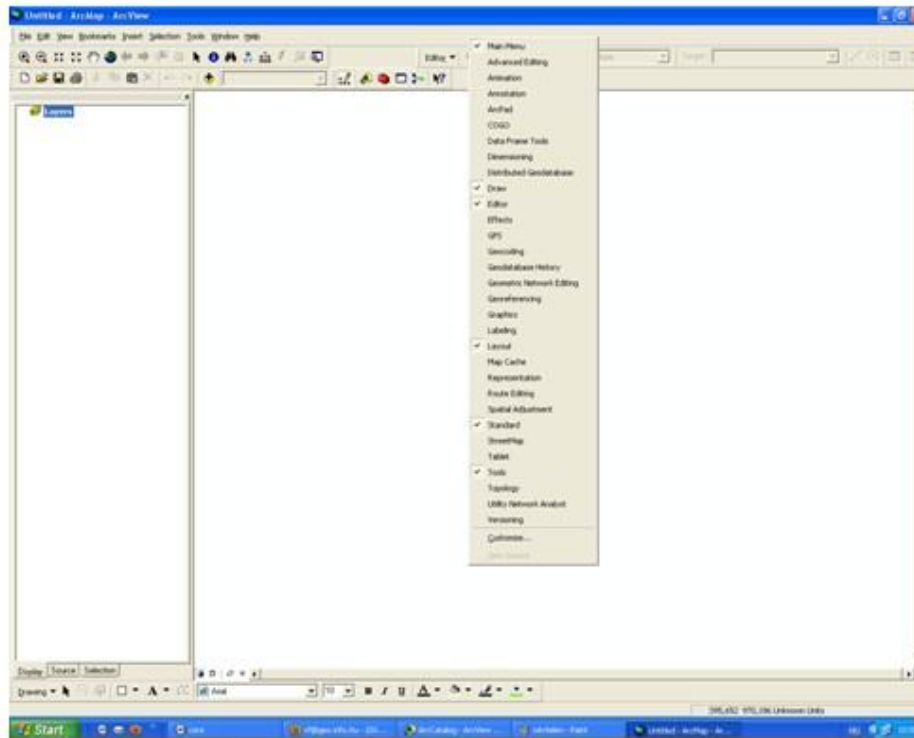


- Párbeszéd ablak

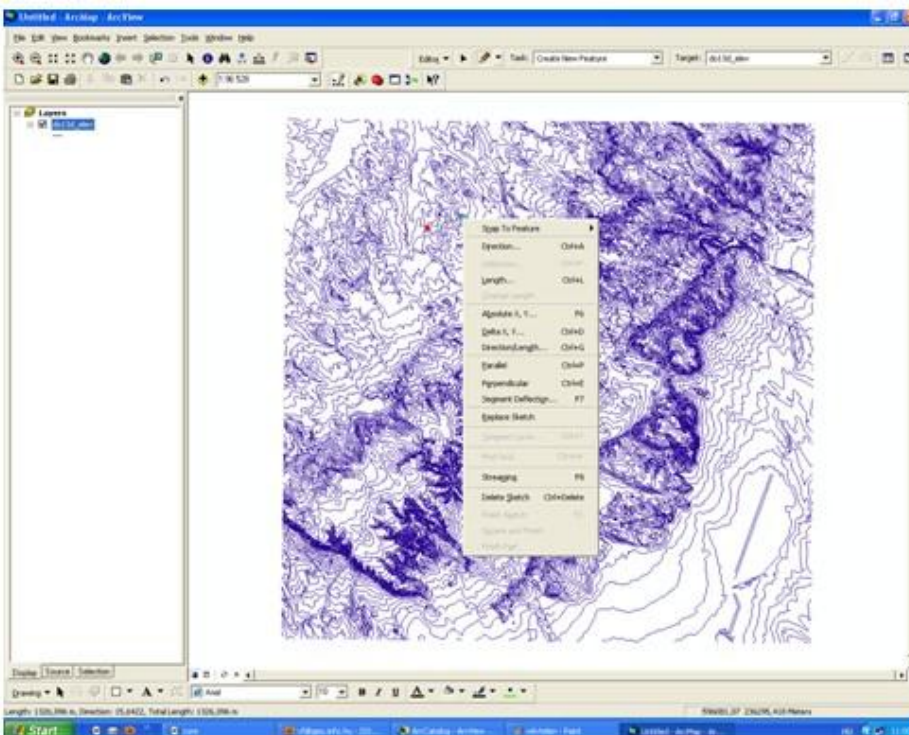


- Lebegő menü

A felhasználói felület értékelése, a
térinformatikai szoftver testre
szabása

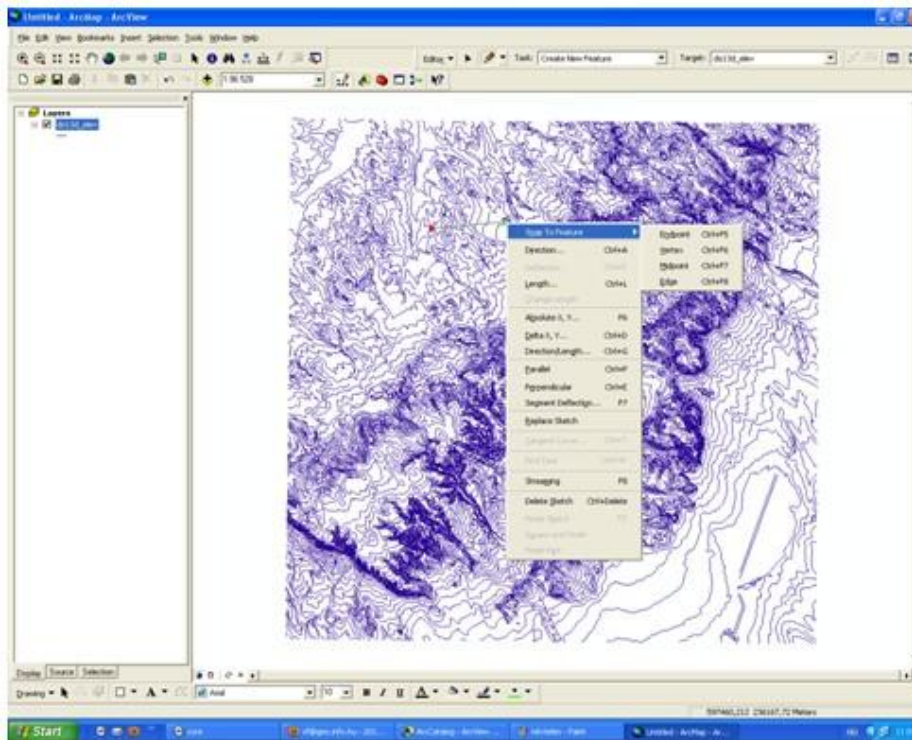


- Helyzet érzékeny menü

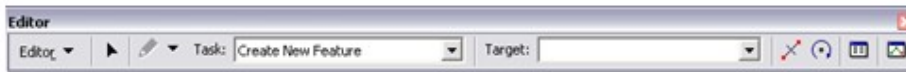


- Helyzet érzékeny menü legördülő listával

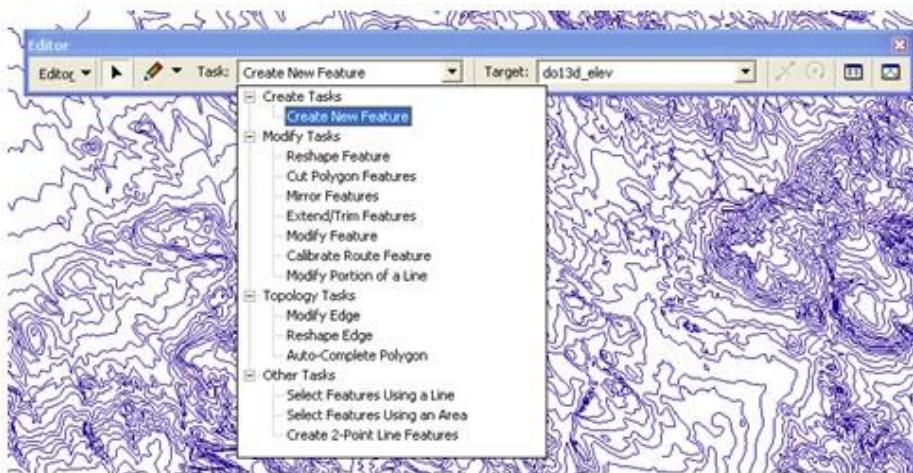
A felhasználói felület értékelése, a
térinformatikai szoftver testre
szabása



- Eszközsáv (pozicionálható)



- Eszközsáv legördülő menüvel



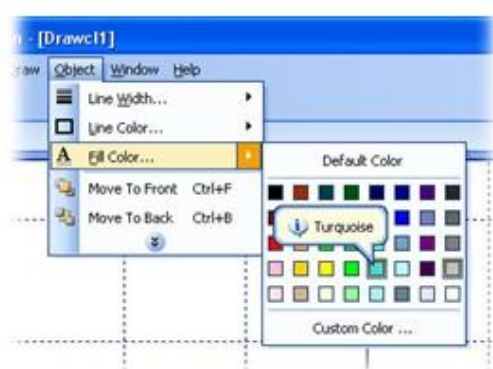
- Eszköz paletta



- Állapotsor, üzenő sáv, rögzített eszköz sáv



- Színválasztó menü



A jó felhasználói felületnek sokféle igényt kellene kielégíteni. Tipikus jelenség például, hogy a gyakorlott felhasználók idővel áttérnek az egér használatáról a rövidítő billentyűkombinációk használatára, mert idegesíti őket a kötött menüpont - elérés. A jó felhasználó felülettel szemben az alábbi követelményeket lehetne megfogalmazni:

- szolgálja ki a kezdő, közepes és gyakorlott felhasználót a saját szintjén
- ne legyen túl sok funkciója
- adjon lehetőséget egy probléma több módon való megoldására

A felhasználó felület tervezésével kapcsolatban nem lehet stabil és mindenre kiterjedő irányelveket megadni. Az alábbiakban néhány téma köré csoportosítva megpróbáljuk összefoglalni a jó felhasználói felület kialakításának elveit.

3. 7.3 A felhasználói felület értékelése

3.1. 7.3.1 Kijelzési formák

- a fogalmakat következetesen használjuk
- a képernyő mezőit ne váltogassuk
- a listákat rendezzük oszlopokba
- ne tegyünk minden oldal tetejére fejléctet egy hosszú listában
- számszerű mezők tartalmát ne kódoljuk át betűkké
- a kiíratások helyét ne váltogassuk
- ne zsúfoljuk a képernyőt
- ha lehet, a listákban különítsük el a csoportokat
- módjával használjuk a kiemelést

3.2. 7.3.2 Fogalmazás

- a rövidítéseket egységesen használjuk

- adjunk a rövidítések mellé szótárat
- használjunk világos, értelmezhető rövidítéseket
- minimalizáljuk a szakmai zsargont
- használjunk közérthető fogalmakat
- egyszerű kifejezéseket használjunk

3.3. 7.3.3 Színek

- a színeket módjával alkalmazzuk
- használjuk a színeket kiemelésre, keresések eredményének kijelzésére
- következetesen használjuk a színeket
- tartsuk be a színek társításának hagyományait
- az adatok megkülönböztetésére használjunk kontrasztokat
- a háttér színe legyen szolid

3.4. 7.3.4 Grafika

- használjuk a grafikát a képernyő zsúfoltságának csökkentésére
- használjunk grafikát az összetett kapcsolatok ábrázolására
- a konkrét számokat ne helyettesítsük grafikával
- használjunk ikonokat a kezelés megkönnyítésére

3.5. 7.3.5 Párbeszéd

- adjunk lehetőséget az adatbevitel megerősítésére
- tájékoztassunk a sikeres végrehajtásról
- ne legyen soha üres ablak vagy monitor
- írjuk ki a választási lehetőségeket
- írjuk ki a szükséges információkat
- jelezzük, hogyan lehet folytatni a munkát
- első helyen legyenek a menüben a leggyakrabban használt parancsok
- az első lépés legyen az első helyen
- adjunk lehetőséget a parancs visszavonására
- csak az aktív parancsokat jelezzük
- használjunk természetes jeleket: nyomógomb, tolóka stb.
- a leggyakoribb gépelési hibákat ismerje fel a program
- engedjünk részparancsokat használni
- a „romboló” parancsokat (törlés, felülírás stb.) többször hagyassuk jóvá

- a beviteli promptot mindig ugyanoda tegyük a monitoron
- egységes stílust alkalmazzunk
- adjunk lehetőséget a felhasználónak saját parancssorok összeállítására
- a felhasználó által bevitt karaktereket írassuk ki a monitorra

3.6. 7.3.6 Adatbevitel

- minimalizáljuk a kurzormozgatást
- használjunk egységes beviteli eljárást
- jelezzük a helyes adatformátumot
- jelezzük az alapértelmezéseket
- használjunk elterjedt mértékegységeket
- csökkentsük minimumra a gépelési szükségletet
- kerüljük az adatok újra bevitelét
- kerüljük az összetett beviteli formákat
- használjunk következetes promptot
- tegyünk lehetővé eltérő beviteli formákat

3.7. 7.3.7 Eszközök

- használjunk egeret a sok kurzormozgást igénylő feladatoknál
- használjunk egeret a kiválasztási és voncsolási művelethez
- ne váltogassuk sűrűn az egér - billentyűzet használatot
- sok dolog kiválasztásához használjunk egeret
- a gyakori funkciókhoz rendeljünk billentyű kombinációkat
- általában a billentyűzetet használjuk adatbevitelre
- rövid mozgásokhoz használjuk a billentyűzet nyilait

3.8. 7.3.8 Hibaüzenetek és segítségadás

- minden adatbevitelt nyugtázzunk
- adjunk lehetőséget javításra
- a hibaüzenetet a beviteli képernyőre írassuk
- emeljük ki a hibaüzenetet
- a régi hibaüzeneteket töröljük a képernyőről
- a hibaüzenet ne legyen semmitmondó
- a hibaüzenet legyen rövid
- a hibaüzenet csak az adott szituációra vonatkozzon

- a hibaüzenet tájékoztasson a lehetséges megoldásokról
- több szintű segítséget adjunk
- a hibaüzeneteket listázzuk
- adjunk valós idejű segítséget
- tegyük lehetővé a segítő szövegben való keresést
- a segítségkérés miatt nem veszhet el adat és nem állhat le a program

4. 7.4 A térinformatikai szoftver testre szabása

A fejlett GIS rendszerek lehetővé teszik, hogy a felhasználó saját, feladat specifikus felületet készítsen vagy készíttessen. Ezt nevezik a **térinformatikai rendszer testre szabásának**. Gyakran előfordul, hogy az alábbi feladatok valamelyikét kellene megoldanunk a térinformatikai szoftver segítségével:

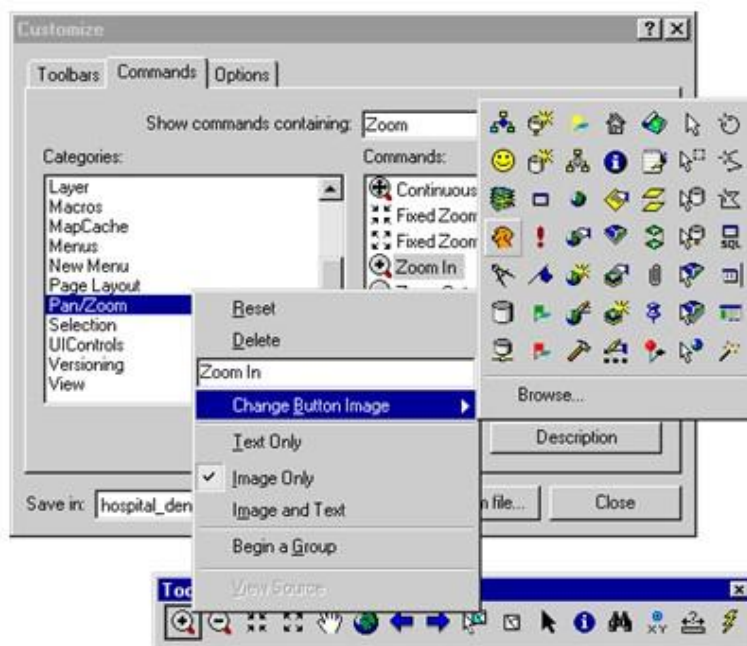
- Eszköztárak elhelyezése
- Felhasználói ikonok létrehozása
- A nem használatos (vagy tiltott – pl. törlés) ikonok, parancsok eltávolítása a menüből
- Stílusok betöltése
- Előre gyártott sablonok készítése és betöltése (pl. EOTR térkép mintája)
- az ismétlődő feladatok automatizálása
- saját parancsok előállítás a program parancsaiból
- speciális környezet előállításához segédfunkciók programozása
- a rendszerhez nem értő felhasználó számára az ő szükségleteihez igazított, menüvezérelt felhasználói felület előállítása.

A fentiek alapján két fő típusa van testre szabásnak: parancsokból makrók előállítása és felhasználói menük megalkotása.

A makró tulajdonképpen célszerű sorrendben egy kötegbe foglalt parancsok sorozata. A menü olyan makrók sorozata, amely a felhasználó igényei szerint készül és "elrejt" a rendszer parancsait és adatbázisait a felhasználó elől. Ez akkor hasznos, ha a felhasználónak sürgősen megismerni a rendszer összes funkcióját, és mindig ugyanazt a jól körülhatárolható feladatot kell megoldania. Meg kell azonban jegyezni, hogy a testre szabás általában korlátozza a felhasználó hozzáférését a térinformatikai rendszerhez.

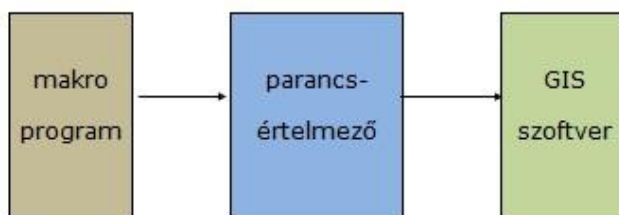
A makrók általában úgy készülnek, hogy egy szövegszerkesztővel egymás után leírjuk a végrehajtani kívánt parancsokat, majd a makró végén utasítást adunk a parancsértelmezőhöz, vagy a menühez való visszatérésre. Ezek után a makrónak nevet adunk (ez tulajdonképpen egy fájlnev) vagy hozzárendelhetjük egy billentyűkombinációhoz, illetve egyes rendszereknél fölvehetjük a menüpontok közé. Néhány térinformatikai szoftver lehetővé tesz automatikus makró készítést oly módon, hogy "figyeli" az általunk kiadott parancsokat, és ezeket kötegelni egy makróba.

A fent említett felhasználói menü tulajdonképpen makrók sorozata, kiegészítve olyan lehetőségekkel, amelyeket általában a programozási nyelvek megengednek. A menü nem más, mint választási lehetőségek sorozata a képernyőn megjelenítve. Ma már követelmény legördülő menü és egér használatának biztosítása a felhasználó részére. A menüben való választás elindíthat egy parancsot, egy operációs rendszer parancsot vagy egy újabb menüt. A menü jól elhatárolható részekből áll: az első rész megjeleníti a képernyőn a felhasználó választási lehetőségeit, a másik rész lehetővé teszi a választás végrehajtását.



5. ábra Felhasználói ikon lecserélése⁷

A makró nyelv működése lényegében azonos a magas szintű programozási nyelvek működésével. A makróban összefoglalt parancsokat, illetve a billentyűzeten adott válaszokat egy parancsértelmező feldolgozza, majd átalakítja a térinformatikai szoftver rendszer parancsaivá, amelyek így végrehajthatók.



A makró nyelv elemei szintén hasonlítanak a programnyelvek elemeihez, és a következőkben összefoglaljuk a jellemző nyelvi elemeket.

Megjegyzések - a programban bárhol elhelyezhető szövegek, amelyeket a parancsértelmező nem hajt végre. Ezek főleg a programozónak hasznosak.

Parancsok - a makró nyelv funkcionális részei, amelyek egy meghatározott műveletet hajtanak végre.

Változók - a legtöbb parancs szimbolikus változókkal dolgozik, amelynek értéke a változó aktuális értékétől függ. A változó különböző típusú lehet: szöveg, egész szám, valós szám, vagy a fentiekből összeállított kifejezés.

Rögzített változók - ezek akkor hasznosak, ha egy változónak csak néhány érvényes értéke lehet és ezt felkínálva a felhasználónak megkíméljük őt a rossz választástól.

Lokális és globális változók - nagyméretű makróknál hasznos lehet, ha ugyanazt a változónevet több helyen használhatjuk anélkül, hogy zavarnák egymást (lokális változó), vagy ha egy változó értéke használható mindenütt a makróban (globális változó).

Érték átadása - ha egy program egy másik program eredményeit dolgozza fel, akkor lehetővé kell tenni az értékátadást.

⁷ http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=An_overview_of_customizing_ArcGIS

Futási kontroll - ezek olyan utasítások, amelyek a program további végrehajtását logikai döntéstől teszi függővé (pl. GOTO (címke) IF (logikai kifejezés)).

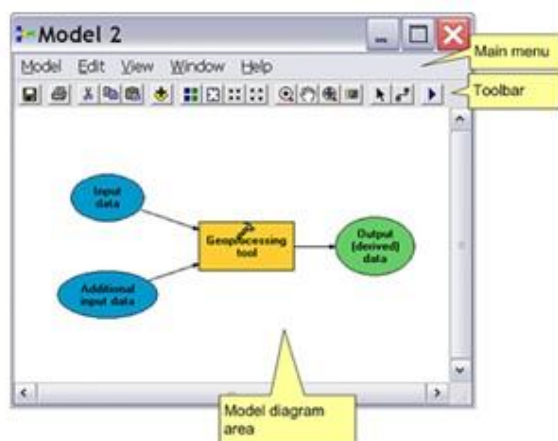
Hurok képzés - egyes programrészek ismételt végrehajtását teszi lehetővé.

A fentiekén túl természetesen a makró nyelv fejlettségétől függően még nagyon sok funkció és kényelmi szolgáltatás képzelhető el.

Korábban a CAD és térinformatikai szoftverekhez kifejlesztettek saját programozási nyelvet (AuoCad Lisp, Mapinfo Mapbasic). Ezeknek az volt az előnye, hogy előre definiált módon fértek hozzá az adatbázis objektumaihoz és a szoftver változóihoz. Hátrányuk az volt, hogy speciális programnyelvet kellett tanulni, a szoftver verzióval együtt elavultak, és az új verzióhoz újra kellett programozni a makrókat. Ráadásul nehézkes volt az objektumok cseréje (export – import) a nem térinformatikai alkalmazásokkal. Napjainkban a legtöbb térinformatikai szoftvert valamilyen fejlesztő nyelven írják (pl. Visual Basic, Visual C ++, Java). Ezeken a nyelven készülnek más szakterületek (operációs rendszerek, irodai alkalmazások stb.) szoftverei is. Ez a tény lehetővé teszi, hogy a térinformatika objektum szinten beágyazódjon a többi alkalmazás közé. A VB nyelven megírt makrókkal tudjuk kontrollálni a felhasználó tevékenységét és az arra adott választ (pl. a párbeszéd ablak melyik gombját nyomta meg, mi történjen az illető gomb megnyomásakor, stb.).

Általában elmondhatjuk, hogy a felhasználói interfész fejlesztése ciklikus feladat. Az első fázisban megtervezzük a rendszert, majd megírjuk a makrókat. Ezután kipróbáljuk az eredményt és első lépésben a programozás formai hibáit szüntetjük meg (szintaktikai ellenőrzés). Ekkor megmutatjuk a felhasználónak, ami után több - kevesebb áttervezés majd programozás és szintaktikai ellenőrzés következik. A folyamat végén aztán előáll a végleges felhasználói felület.

További testre szabási lehetőség a térinformatikai modellezés automatizálása. Ez a lehetőség részletesen ismertetésre kerül a következő félévben, ezért itt csak az alapelvét ismertetjük. A térinformatikai adatbázisok felhasználói általában gyakran ismétlik ugyanazokat a típusú elemzési feladatokat. Egy építhetőségi előadónak például rutinfeladat a kérdéses földrészlet közvetlen szomszédjainak kiválasztása, adataik kikeresése és részükre a határozat egy példányának megküldése egyetértés vagy fellebbezés lehetőségének biztosítására. Az ilyen gyakran ismétlődő elemzési feladatok megoldására való a modellezés automatizálása. Ezt a műveletet a térinformatikai szoftverek vizuálisan támogatják, fogd – és - vidd módszerrel rakhatjuk össze a modellünket.



6. ábra Vizuális modellépítő környezet⁸

A modellépítő alkalmazás nem csak a modell áttekintését könnyíti meg, hanem egyúttal ellenőrzi a modell kapcsolatainak helyességét és így megelőzi a formai hibákat (attribútum táblát nem vehetünk alá bufferelés műveletének). A formális modellt a következő lépésben paraméterezni kell (megmutatni a konkrét adatok helyét). Az elkészült modellt el lehet nevezni és akárhányszor futtathatjuk.

5. 7.5 Térinformatikai adatbázis megosztása

A címben jelzett megosztás annak a helyzetnek a megoldását jelenti, amikor egyszerre több felhasználó szeretne az adatbázishoz hozzáférni és azon műveleteket végezni. Ma az elosztott adatbázisok korában a felhasználók a

⁸ http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=An_overview_of_ModelBuilder

térinformatikai rendszerek esetében is a több felhasználós, hálózatban elérhető változatot részesítik előnyben. Az ilyen típusú használat megsértheti az **adatintegritás** elvét, amit bizonyos intézkedésekkel el lehet kerülni. Például, ha az egyik felhasználó megváltoztat egy objektumot, mialatt a másik műveletet végez rajta, az eredmény hamis lesz mindkét felhasználó számára. Ilyenkor egy megoldás lehet az egyik kérés zárolása a másik művelet befejeződéséig. Az egyidejű felhasználókat az alábbi alapelvek szerint lehet ellenőrizni:

- az adatbázis csak a tranzakció véget érése után változtatható meg
- az egyidejű felhasználók sosem láthatják a befejezetlen tranzakciók hatását
- ha két felhasználó ütközik, a konfliktusukat még az adatbázis megnyitása előtt fel kell oldani

Az egyidejű hozzáférésnek három típusa van:

- védelem nélküli - bárki lekérdezhet és változtathat adatot. Ez nagyon ritka, veszélyes megoldás, de ha megtörténik, a rendszer figyelmezteti a felhasználót, hogy az általa kért adatot valaki éppen használja.
- védett - bárki lekérdezhet adatot, de csak egyvalaki változtathatja.
- kizárólagos - csak egyvalaki kérdezhet le és változtathat adatot

A térinformatikai adatbázisok körében létezik egy olyan művelet, amely nem fordul elő az átlagos adatbázisok esetében. Arról van szó, hogy az adatbázis egy részét tartósan - több órára vagy több napra - kiemelik, megváltoztatják majd visszahelyezik az adatbázisba. Jó példa erre, amikor a térkép egy része megváltozott, amit általában hosszú idő alatt lehet újradigitalizálni vagy szerkeszteni. Mivel ez nagyon hosszú tranzakció, ebben az esetben a fő adatbázist - amelyből a kiemelés történt - zárolni kell, ilyenkor más nem változtathat rajta amíg a megkezdett tranzakció be nem fejeződik. Ez alatt a lekérdezés lehetséges, az eredmény persze a lezárásakor érvényes állapotra vonatkozik. A fő adatbázis azonban ebben az esetben is védtelen marad arra - az egyébként rövid - időre, amíg a módosítások visszahelyezése megtörténik. Gyakori kérdés, hogy mennyi adatot kell védeni a tranzakció során? A legegyszerűbb válasz az, hogy mindazokat, amelyeket a tranzakció érint. A térinformatikában ez sem egyszerű kérdés, mert pl. egy vonalas létesítmény változtatása esetén nehéz megmondani, hogy hány térképszelvényt kell lezárni a változtatás idejére. Az adatvédelmet különböző szinteken lehet megvalósítani. A „legalsó” szint az adategység védelme, amikor csak egy adatot védünk. Ennél magasabb szint a rekordok, majd az adattábla védelme. A következő szint a megjelenítési szint védelme, amikor csak a lekérdezésben szereplő adatokat védjük. Az utolsó szint a teljes adatbázis védelme. Végül meg kell említeni egy ritka esetet, a **kölcsönös kizárást**. Ekkor két felhasználó egy időben a másik felhasználó által igényelt adatokra kérdez rá, így elvileg a végtelenségig várakozhatnak egymásra, sőt az összes többi felhasználó egy idő után rájuk vár. A konfliktust az oldja föl, hogy a felhasználók megunva a várakozást (miután foglalt jelzést kaptak) sorra visszavonulnak, így az ütközés megszűnik.

A nagy adatbázisok egyik kritikus állapota az **adatvesztés**. Mivel az archiválás nem valós időben történik, csak az utolsó mentés állapotát tükrözi. A nagyobb adatbázis-kezelők a jóváhagyott tranzakciókat folyamatosan mentik, így csak a még nem jóváhagyott kérések vesznek el. Az adatvesztés egyik fő oka az adatbázis-kezelő rendszer megszakítása az operációs rendszer, az operátor vagy a hardver hibája miatt. Ez után a hiba után az adatbázis utolsó állapota hamar visszaállítható. A másik jellemző hiba magának az adattárolónak a megsérülése. Ez ritkábban fordul elő, viszont a visszaállítás sokkal hosszabb és az eredménye is bizonytalan.

Az értékes adatbázisokat az **illetéktelen hozzáféréstől** is védeni kell. Az elsődleges védelem az operációs rendszer feladata, amely csak jelszó megadása után enged meg a hozzáférést. A már beengedett felhasználó elöl is elzárhatjuk az adatbázis azon részét, amelyek nem tartoznak rá. Úgy is védekezhetünk az illetéktelen felhasználás ellen, hogy bizonyos felhasználók csak generalizált, összevont adatokhoz juthatnak hozzá. Az utolsó két típusú védekezés az adatbázis-kezelő rendszer feladata.

Az eddig leírt elvek mindig valamely konkrét alkalmazáson keresztül valósulnak meg. Az alábbiakban ismertetjük az ESRI térinformatikai cég megoldását a térbeli adatokat tároló adatbázisok kezelésére.

5.1. 7.5.1 Az ArcStorm - térbeli adatbázis menedzser

A mai GIS rendszerek egyik nagy kihívása, hogy több felhasználó számára lehetővé tegyék a térbeli adatbázisok egyidejű használatát. A korábban kifejlesztett LIBRARIAN modul a nagy földrajzi adatbázisokat részekre bontva kínálta fel a felhasználó számára. Ha egy felhasználó munkába vett egy részt, a szoftver a többi

felhasználót kizárta a kérdéses rész átalakításának lehetőségéből. Ez a módszer sok esetben megfelelőnek bizonyult, a nehézségek akkor jelentkeztek, amikor több felhasználó szeretett volna ugyanazon a területen más-más egyedet megváltoztatni. Ekkor ugyanis az egyik felhasználónak meg kellett várnia, amíg a másik befejezi a munkáját és feloldja a munkába vett rész lezárását. Erre a problémára válaszolva az ESRI kifejlesztette a címben említett ArcStorm (Arc Storage Manager) modult. A modul kulcsfunkciói:

- objektum szintű tranzakció kezelés
- egységes adatkezelés a térbeli és a leíró adatok között szabványos RDBS keretein belül (Oracle, INGRES, INFORMIX és SYBASE)
- hálózattól független adatelérés és adatmegjelenítés (a felhasználó nem tudja, hogy fizikailag hol helyezkedik el az adat)
- az archiválás eredményének átvezetése az adatbázison
- kliens/szerver kiszolgáló mechanizmus

Az ArcStorm fő tulajdonságai:

- objektum-szintű tranzakciók. A szoftver az adatokat az objektumok szintjén kezeli. Ez azt jelenti, hogy egy objektumot anélkül tudunk megváltoztatni, hogy a többi felhasználót kizárnánk más objektumok megváltoztatásából.
- egységes tranzakció-kezelés a térbeli és nem térbeli adatokra nézve egyaránt. A program összehangolja a térbeli adatok és a külső adatbázis-kezelőben (RDBMS) kezelt leíró adatok változtatását, így az adatbázis mindig konzisztens marad.
- a külső adatbázis-kezelő rekordjainak ideiglenes lezárása. Az ArcStorm lezárja a megfelelő oszlopot a külső adatbázis-kezelőben a tranzakció végéig.
- visszaállítási lehetőség. Ha adatvesztés történik valamilyen okból, az ArcStorm visszaállítja az adatbázist a legutolsó konzisztens állapotba.
- az adatbázis szerkezetét csak az ArcStorm Adatbázis Adminisztrátor változtathatja meg (kívülről nem tudunk belemélyülni) ezért az adatbázis integritása sértetlen marad.
- archiválás. Az ArcStorm kezeli az adatbázis történelmére vonatkozó információkat. Az adatváltoztatás ténye rögzítésre kerül, amit visszaállításra vagy áttekintésre használhatunk.
- kliens/szerver adatkiszolgáló rendszer.
- adatdefiníciós nyelv. A fedvények, könyvtárak és külső adattáblák csak az Adatbázis Adminisztrátor közreműködésével definiálhatók.
- közvetlen adatbázis elérés az ArcStorm jelenléte nélkül az ESRI által írt alkalmazások számára (ArcView, ARC/INFO)

A fentiek közül a kliens/szerver rendszer azt jelenti közelebbről, hogy az ArcStorm ellenőrzi minden adatátvitelt az ügyfél és az adatbázis között. Ezzel megőrizhető az adatbázis önállósága és biztonsága. Valahányszor egy újabb ügyfél jelentkezik be a rendszerbe, elindul egy szerver program az ő kiszolgálására. Ezek a programok nagyrészt a kiszolgáló szerveren futnak, de ha túlterhelés fenyeget, néhány funkciójuk áttelepíthető a felhasználó gépére.

6. 7.6 Összefoglalás

Ebben a fejezetben olyan kérdésekről tárgyaltunk, amelyek a térinformatika megjelenésével, testre szabásával és a térinformatikai adatbázis osztott használatával kapcsolatosak. Bemutattuk a felhasználói felületek fejlődését, korszerű felhasználói felület részeit. Foglalkoztunk a térinformatikai szoftver átalakítási lehetőségeivel, úgymint a felhasználói felület átalakítása, új parancsok megírása és a lekérdezések automatizálása. Részleteztük a térinformatikai adatbázis megosztott használatának feltételeit és megoldását.

Önellenőrző kérdések

Milyen változásokon estek át a felhasználói felületek?

Milyen fő részei vannak egy grafikus felhasználói felületnek?

Milyen szempontok alapján értékeljük a felhasználói felületeket?

Milyen lehetőségeink vannak a térinformatikai szoftver testre szabása terén?

Melyek a makro programnyelvek fő komponensei?

Mit jelent a modellezés vizuális támogatása?

Milyen feltételei vannak a térinformatikai adatbázis megosztásának?

Milyen sajátosságai vannak a térbeli adatokat tartalmazó adatbázisok kezelésének?

Hogyan lehet megőrizni az adatbázis integritását több felhasználó egyidejű kérései esetén?

Milyen fő funkciói vannak egy adatbázis menedzsernek?

Irodalomjegyzék

Detrekői Ákos - Szabó György: *Bevezetés a térinformatikába*, Nemzeti Tankönyvkiadó, Budapest, 1995

NCGIA Core Curriculum II.,III., Térinformatikai alapismeretek, Székesfehérvár 1994

David J Maguire, Michael Goodchild, David Rhind: *Geographical Information systems*, Longman Scientific & Technician

ARCSTORM - ARC Storage Manager, ESRI White Paper Series , 1995

ESRI's Approach to interoperability, ESRI White Paper Series , 1995