

Micro-ISIS vagy TEXTAR?

Bakonyi Géza - Kokas Károly

A fejlettebb világban aligha értelmezhető az a kép, amely egy átlagos hazai, gépesedni vágyó könyvtárban fogad bennünket. Értő és nem értő könyvtárosok hajolnak a kapacitásában már eleve kicsinek tűnő PC fölé, hogy rajta és vele elérjék álmukat, az automatizált könyvtári rendszer megvalósítását. Az egyszemélyi munkára kitalált gépet (personal computer) kellene „megtámasztani”, hogy hálózatba kapcsolva, szerzeményezési, feldolgozási feladatokon túl alapja legyen az online katalógusnak és persze a kölcsönzésnek is. A könyvtárosi berkekben megjelenő programokat bizony ezzel az álommal szembesítjük, s aztán megalkuszunk, mert gépesíteni pedig „necesse est”.

Alig három éve, hogy a Tudományos és Műszaki Tájékoztatás hasábjain¹ kezdetét vette a vita - a számítástechnikai szakemberek között. A TEXTAR (akkor még FREETEXT) készítőinek a saját fejlesztés szükségessége melletti érvelésével szemben a Micro-ISIS hazai elterjesztésében jelentős szerepet vállaló szakember, az ISIS egyre újabb és újabb verzióinak plusz szolgáltatásait sorolta. Mivel munkahelyünkön (JATE) a „nagy” ISIS-szel való munkának voltak már bizonyos tapasztalatai, és az egyetem számítógépes szakemberei rendelkeztek UNESCO kapcsolatokkal, talán természetes, hogy a szegedi Egyetemi Könyvtárban a Micro-ISIS használata mellett döntöttünk (egy hosszabb C-64/Superbase, és egy rövid PC XT/dBASE kísérletezés után). Az elmondottakból már következik, hogy ebben az értelemben egy kicsit elfogultak vagyunk. Elfogulatlanok vagyunk viszont mint könyvtárosok, akik őszintén vágyódnak „a lehetséges programok legjobbjára” után. Ezért nem mulasztottuk el - igaz az ISIS-éhez képest talán felszínesnek is mondható - megismerkedést a TEXTAR-ral.² Az összehasonlítás pillanatában a hazai fejlesztésű program 4.03-as, illetve 5.1-es verziójával találkoztunk, míg az általunk is használt ISIS változat a 2.3-as.

Filozófiai problémák: feldolgozó rendszer vagy online katalógus?

Már az első pillantásra látszik, hogy a TEXTAR létrehozóinak szeme előtt egy integrált rendszer megalkotásának kihívásai lebegtek. Mégpedig olyan rendszeré, amely a feldolgozás és az adatbevitel szempontjából közelíti meg a problémát. A Micro-ISIS ezzel szemben nem az adatbázis-építő kényelmét tartotta szem előtt, hanem a programot online katalógusként felfogó felhasználót, bárki legyen is az. A TEXTAR készítői folyamatosan kapcsolatban vannak a hazai (elsősorban) könyvtári-dokumentációs szakemberekkel, s kisebb-nagyobb módosításokkal igyekeznek megoldani a felmerülő kérdéseket. Az ISIS megalkotóitól szükségképpen távolra került a program, a felhasználói visszacsatolás pedig lassú is, kétséges is. Ezért a szoftver főbb paramétereit nagyobb általánosságokban definiálták, megkísérelve a „szülőktől független” életre való felkészítést. Ennek következtében a Micro-ISIS nem csupán általános szöveges adatbáziskezelő program, hanem (kicsit a dBASE családdhoz hasonlóan) egyben adatbáziskezelő nyelv is. Pontosabban: a Micro-ISIS felhasználója képes egy viszonylag ismert programnyelv (PASCAL) tipikusan erre a rendszerre specializált változatával saját programokat írni. Ezekben a programokban „meghívhatja” az ISIS által használt eljárásokat (ez sokszor nagyon leegyszerűsíti a programozói feladatot), és a kész forrásfájlt az ISIS beépített fordítója közvetlenül futtatható programmá alakítja. Ezzel az eljárással a Micro-ISIS a „felismerhetetlenségig” átalakítható, számtalan új funkció beépíthető, s mindezen szolgáltatások egy - akár az egész ISIS-t is eltakaró - menüből vezérelhetők, vagyis az így preparált programmal szembesülő könyvtári szakember (vagy olvasó) egységes rendszernek látja az eredményt. Véleményünk szerint ennek a szolgáltatásnak a hiánya a TEXTAR esetében azt jelenti: nemcsak, hogy két elgondolás

(filozófia) áll egymással szemben, hanem két különböző „súlycsoport” is. Persze azt készséggel elismerjük, hogy az ISIS-nek a fentiekben jelzett átalakításához általában nem elegendő a könyvtárosi szakismeret. Kétségtelen az is, hogy az ISIS ezen rugalmas felülete szinte kínálja a folytonos átalakítást, ami nem feltétlen előny pl. egy egységes regionális társulási viszony felől szemlélve.

Ezen a ponton talán elnézést is kellene kérnünk a hosszúra nyúlt bevezetőért, ami semmi mást nem volt hivatott igazolni, mint azt, hogy összehasonlító vizsgálatunk bevallottan koncepciózus. Az az idő, amíg a könyvtáros feláll az íróasztala mellől, hogy leüljön a számítógép terminálja elé, valóban sokkal rövidebb, mint amennyit az itt leírtak elolvasása megkövetel. Ez azonban csak látszólag van így. Valójában dimenziókon ugrik át: egyszerű „kétkezi” bibliotekariusból, számítógépesített könyvtári rendszert használó könyvtárossá lényegül át. Ebből következik, hogy a két könyvtári program használhatóságának összehasonlítása nem sokat ér, ha csupán a különböző funkciók meglétének vagy hiányának, alkalmazhatóságának vagy alkalmazhatatlanságának kipipálására szorítkozik (végül is - különösen egy könyvtáros számára - minden elsajátítható). Többről van szó: a számítógép alkalmazásának kétfajta viszonyáról a számítógépesített könyvtár ideáljához. Most jutottunk el ahhoz a pillanathoz, amikor a könyvtáros helyet foglal a villódzó képernyő előtt, és szembetalálja magát a konkrét problémákkal:

- az egyedi alkalmazás létrehozása (adatbázis-definíció);
- adatbevitel;
- adatkarbantartás;
- adatvisszakeresés;
- adatszolgáltatás;
- hálózati alkalmazás;
- barátságossági tényezők (help-képernyők, dokumentáció stb.)

A továbbiakban ezeken a területeken fogunk vizsgálni - most már a könyvtáros képernyő elé ülésének pillanatából kiindulva.

Az adatbázis(ok) definiálása

Valljuk meg, ebből az igencsak alapvető szempontból egyik rendszer sem igazán felhasználóbarát. Sőt, a TEXTAR előzetesen editorral kicentizett definíciói kifejezetten barátságatlanok. A logikája mindkét rendszernek hasonló: hozzávetőleg négy szempont határozza meg az adatbázist, hogy minimálisan működőképes legyen. Az előzetesen megtervezett adatmodell alapján

- 1) meghatározzuk a rekord szerkezetét,
- 2) létrehozuk a beviteli űrlapo(ka)t,
- 3) megírjuk a megjelenítő és/vagy nyomtatási formátumo(ka)t, majd
- 4) kijelöljük, hogy mely mezők s milyen módon vesznek részt a visszakeresést lehetővé tevő indexfájl fölépítésében. Ez a négy funkció minden kényelmetlensége ellenére az ISIS-ben a programból hívható, s a program futása közben módosítható is. Ez a tervezési munka a TEXTAR esetében előre elvégzendő, s ráadásul mindenfajta program-kontroll nélkül, vagyis elkövetett hibáink csak a próba futtatáskor derülnek ki. Ha a négy definíciót módosítani akarnánk, akkor is ki kell lépni a programból.

Itt beszélnünk kell még az időtényezőről is, aminek az adatbázis-definíción túl is jelentősége van. A könyvtári munkák jó része nagy mennyiségű mechanikus munkával jár, ráadásul sokszor hirtelen adódik egy elvégzendő feladat, melynek gyorsan kell az eredménye. Például: az ajándékozni készülő alapítvány számára elegánsan nyomtatott dezideráta listát kell adni. Ehhez gyorsan létre kell hozni egy nem kompatibilis adatbázist az egyik hálózati

könyvtárunkban, betöltve oda a mi - központi -katalógusunkból leválogatott rekordokat, de más adatszerkezettel. Úgy véljük, hogy ezek és a hasonló feladatok az ISIS-ben egyszerűen és gyorsan megoldhatók, így segítik kiküszöbölni az „amire nekiállunk, papíron már meg is van” típusú (önbecsapó) reakciók megjelenését. Az időtényező szempontjából a TEXTAR is rendelkezik egy hallatlan előnnyel: az indexelés (mindkét programnál) időt rabló folyamata bármikor felfüggeszthető, s a munka később folytatható. A Micro-ISIS legfeljebb három szegmensben végzi el a munkát, de egyébként a folyamat maga csak erőszakos beavatkozással szakítható meg.

Adatbevitel

Az előbbieken felvázolt kétféle filozófia különbözősége itt is megnyilvánul. A TEXTAR esetében az adatbeviteli munkalap megpillantásától kezdve azonnal kiderül, hogy olyan szoftverrel van dolgunk, amely kifejezetten könyvtári típusú adatfeldolgozás céljából íródott, a következő megszorítások figyelembevételével:

a) Elsődleges cél egy központi adatbázis (egyrészt az ún. szolgálati adatbázis, másrészt az Új Könyvek adatbázis) kialakítása és az erre épülő, hagyományos értelemben vett adatszolgáltató hálózat kiépítése (ami viszont nem adatbevitelre, hanem a parancsszavas visszakeresésre korlátozódik). Már itt megjegyezzük az ISIS azon hiányosságát, hogy még ezzel az egyszerű hálózati lehetőséggel sem rendelkezik a PC-és verzió (ellentétben a VAX-variációval).

b) Hangsúlyozottan támogatja a könyvtáros által történő adatbevittelt: vagyis a címleírás precizitását maximálisan támogatja az indexelt mezők esetén, illetve a már egyszer leírt mű adatainak felhasználásakor.

c) A lehetséges visszakeresési szempontokat már az adatbevitelnél igyekszik a lehető legerőteljesebben meghatározni: az ún. kapcsolódó mezők alkalmazása esetén, külön munkalapot bocsát a feldolgozó rendelkezésére, ahol a kapcsolat minősége rögtön meghatározható.

d) Az adatellenőrzés és a karbantartás fontossága már a rekord beviteli űrlapjának kitöltésekor megnyilvánul, hiszen az invertált állomány az adatrögzítés után azonnal módosul. Ugyanezt támogatja a alapértelmezési értékek széles körű használata.

Ezzel szemben az ISIS nem törődik különösképpen azzal, hogy milyen kataszter készül, inkább az információ visszakeresésre helyezi a hangsúlyt. Ennek következtében az ISIS esetében az adatbevitellel kapcsolatban az előzőekben felsoroltakhoz a következőket írhatjuk:

a)+d) Az ISIS-ben létrehozott adatbázisok között nincs hierarchikus viszony: elsősorban az egyenrangúság, a mellérendeltség jellemzi őket. Viszont külön program írása nélkül nem is kapcsolhatók össze az adatfájlokban tárolt adatok (ezért pl. kölcsönzési nyilvántartásra csak körülményesen alkalmazható), - ugyanakkor a különböző (akár eltérő szerkezetű) adatbázisok közötti adatsere nem jelent problémát.

b)+c) Az adatbeviteli űrlap egyszerű, lineáris megoldású; de a már meglévő adatokat nem használhatjuk fel. Viszont azzal, hogy a funkcióbilentyűk programozhatók, elérhetjük, hogy egyes mezőket más mezőkbe, illetve más rekordokba átmásolhassunk. Az adott adatbázis többi rekordjával is teremthetünk kapcsolatot, ez azonban nem input-központú, hanem erőteljesen az outputra koncentrált (ami abból is látszik, hogy a nyomtatási formátumnyelv használatával történik az összekapcsolás). Ennél bonyolultabb kapcsolatokat, az ISIS-ből külön is futtatható és a már említett tezaszusz-programmal hozhatunk létre. A kapcsolódásokat elősegítheti még a szinonimafájl létrehozása (any-file), amely persze megint csak az adatvisszakeresés biztonságát növelő tényező.

Elmondható tehát, hogy a TEXTAR-ban az alaposan átgondolt módon (és lassan!) bevitt adatmennyiség önmagánál lényegesen több információt kelt életre, de ezzel meg is nehezíti a keresést. Az ISIS a gyors, egyszerű munkát támogatja, a többlet információ a kereséskor, illetve találatkor jelentkezik.

Információ-visszakeresés

A visszakeresésnél éppen fordított a helyzet, mint az adatbevitelnél: ez lényegesen egyszerűbb és kifinomultabb eszközökkel történik az ISIS-ben, mint a TEXTAR-nál. Itt mutatkozik meg a „filozófiai” különbség másik aspektusa, nevezetesen, hogy az ISIS az adatbevitellel szemben közömbösen viselkedik, de előnyben részesíti az információkeresőt.

A TEXTAR esetében láthatjuk, ami az adatbevitelnél előny volt (a több index használata stb.), most már hátránnyá vált. Pl. ha az olvasó csak a cím egyes szavaira emlékszik, akkor ezt az ISIS két lépésben „megoldja”. A TEXTAR esetében a rejekció-projekció műveletei, illetve a névváltozatok lehetséges megjelenése miatt, ez öt lépés lesz, ráadásul a technika kifinomult ismeretét tételezi fel.

Már ebből is érzékelhető, hogy a TEXTAR információkereső rendszere más elveken alapul, mint az ISIS-é. Legjobban talán az ISI által kifejlesztett Social Sciences Citations Index CD-ROM változatánál használt keresőprogramhoz hasonlítható, de annál jóval körülményesebb. Az ISIS keresési stratégiája a jelenlegi verzióra már teljesen kimunkáltnak mondható (különösen a tezaurusszal együtt). A kereső technika mindazt nyújtja, amire szükségünk lehet: az összes Boole-algebrai műveletet (akár igen sok halmaz között is!), csonkolások és viszonyítások, feltételek (if .. then .. else), függvények, limit-jelek, szabad szövegkeresés (a nem indexelt mezőkben és a már leválogatott halmazokban is). Előnyös az is, hogy megjelenítésnél és rendezésnél ugyanaz a formátum-leíró nyelv használható. Igaz ugyan, hogy az ISIS egy egységes indexet használ, de az invertálásnál megadott prefixek biztosíthatják a kvázi több indexfájl meglétét (ugyanígy definiálhatjuk, hogy a keresés mely mezőkből indexelt fogalmakra vonatkozzék kizárólagosan).

Mindenképpen meg kell említeni, hogy a TEXTAR rendelkezik egy sajátos parancsnnyelvel is, amely lehetővé teszi a bonyolultabb keresési profilok elmentését, majd előhívását. Az eljárás elvileg világos, de kidolgozottságában még a nagyobb szövegszerkesztők makro-nyelvének hatékonyságát és sokoldalúságát sem éri el.

A karakterkészlet problémájáról is szólnunk kell, mert ebből fontos különbségek adódnak. Mindkét program két (s csodák csodája, majd azonos) kódtáblát használ, egyiket a rendezéshez, másikat a megjelenítéshez. Az ISIS-nél kézenfekvőbb az ékezet nélküli term-dictionary használata, ami persze „zajjal” jár, bár tapasztalatunk szerint segít kiszűrni a helyesírási különbségeket. A TEXTAR is tolerálja ezt, de érzékenyebben reagál az eltérő alakokra.

Adatvédelem és relációs adatbázis

Az igazi relációs adatbázis felé vezető úton az ISIS lemaradt a fejlődésben a TEXTAR mögött, sőt - struktúrájából adódóan - már eleve vesztes, bár tezaurusz funkciójának jobb megismerése és használata enyhíthet e gondon. Ebből adódóan hiányzik a már egyszer bevitt adatok (pl. kiadók neve stb.) többszöri felhasználása, pusztán egy másik fájlra való hivatkozással. Ez persze megnöveli az ISIS helyigényét is. Hiszen egy több 10 000 rekordos könyvtári adatbázisban számtalanszor ismétlődnek nevek, intézmények ugyanazon alakban. Az ISIS-ben így az ún. authority-fájl előnyei közvetlenül nem aknázhatók ki: sem helytakarékosági megoldásként, sem egységesítési eljárásaként. (Persze áthidaló megoldás született ez utóbbira: a könyvtárunk online katalógusát kezelő ISIS-mutációban pl. ezt egy

ablakban megjeleníthető, az eddig bevitt 100 000 rekord alapján egységesített „névsorral” kívánjuk pótolni, melyből a helyes névalakot a kurzorral átviheti az adatbevivő az új rekordba.)

A TEXTAR előnyei között szokták emlegetni a rendkívül helytakarékos megoldásait. Ezt általánosságban elismerve, azért figyelniük kell arra is, hogy a számtalan jó tulajdonsággal járó relációs adatkezelés miatt teljességre törekvő kapcsolatok kiépítése esetén, akár igen sok fájlt (és helyet) is használhat. Ugyanis minden, relációban részt vevő adatot típusonként külön fájlba helyez (szerzők, kiadók stb.) és ezeket az állományokat külön indexeli, majd ezen indexek a masterfájl egy-egy segédfájlján keresztül hivatkoznak az alaprekordokra. Így pl. 4 kiemelt adatmező projekciója és rejekciója összesen 12, az addigi állományhoz képest új fájl létrejöttét eredményezi. A relációs adatbázis-szerkezetnek teljesen „bolond-biztosnak” kell lennie, hiszen a kapcsolt fájlból történő törlés akár több ezer rekordot is tönkretesz. A TEXTAR e szempontból biztonságos. Az ISIS lineáris adatbázisából csak logikailag törölhető könnyedén az adatok, fizikai eltávolításuk (s a hely felszabadítása) csak az egész adatbázis karbantartásával oldható meg. A relációs megoldásnak van még egy szembevetendő előnye: a TEXTAR sokkal gyorsabban s „magától értetődőbben” indexel, illetve aktualizál, mint az ISIS, így a bevitel után már rögtön felfrissített index támogatja a következő rekord kitöltését. Persze ezért fizetni is kell. Az ISIS igen jól tolerálja a rekordszerkezet utólagos átalakítását, viszont ezt a TEXTAR-nál csak az adatok teljes export-import művelete árán tehetjük meg, ami igen időigényes s nem is mindig biztonságos eljárás.

Adatcsere

A TEXTAR adatcsere-formátuma jól kidolgozott, legfeljebb azt vethetjük szemére, hogy közvetlenül nem tud igazodni a nemzetközi szabványhoz (ISO 2709), pontosan a relációs jellege miatt. Az ISIS export-import lehetőségei egyszerűbbek ugyan, de igen fontos előnyt jelent, hogy a találati halmaz szerinti többszörösen rendezett exportálásra is mód van, másrészt az egészen más adatszerkezetű adatbázisokból is könnyedén tudunk átvinni adatokat, ha azok kellően szegmentált felépítésűek. Ugyancsak kellemes tulajdonság, hogy az ISIS az export-import funkcióban módot ad a karakterkiosztás adatcsere közbeni megváltoztatására is. A TEXTAR vagy nem rendelkezik ezekkel a lehetőségekkel, vagy külső eszközökkel oldhatók csak meg a problémák.

Megjelenítés

Mindkét program előnyös tulajdonsága a cserélhető képernyő-, illetve nyomtatási formátum, de az ISIS, főként a nyomtató-vezérlés terén, többet tud. Ijesztő bonyolultságúnak tűnő hieroglifái lehetővé teszik a közvetlen többhasábos nyomtatást, a nyomtató opcióinak, sőt fontkészleteinek vezérlését (vastagítás, kurziválás stb.) Ezeket a vezérlő karaktereket kívánságra a nyomtatandó fájlba is behelyezi, így az ilyen típusú állományok akár a DOS háttérnyomtató funkciójával (egyéb munkák közben) is a nyomtatóra küldhetők.

Az ISIS első pillanatra hátrányosnak tűnő, a bevezetőben hangsúlyozott általánosabb meghatározottsága előnyként is értelmezhető. Könyvtári munkánk során igen sokszor használtuk kitűnő index-készítési funkcióit, sokoldalú (már csak a kapcsolati feladatok miatt is alaposan átgondolt) formátumnyelvét. Főleg folyóiratrepertóriumok elkészítésénél, különféle, összetett mutatók készítésénél hasznos a beépített rendezési lehetőségek nagy száma és programozhatósága. Az ISIS-t a készítői eleve úgy tervezték, hogy azzal az adatbázisból közvetlenül, igen finoman vezérelt rendezések után, kitűnően szerkesztett nyomtatott anyagot tudjon létrehozni a felhasználó. S még egyszer hangsúlyozni kell, az ISIS-ből való kilépés nélkül (beépített editorral) a szükséges paraméterek változtathatók. A

TEXTAR e téren főleg azért marad el, mert nem ismeri a felhasználó számára is látható rekordszámozást, s így az nem is tudja kihasználni az ebből származó előnyöket.

Programbarátság: (a felhasználó egyedül marad...)

Külföldi lapok tesztjeiben, ebben a rovatban szokták említeni a „hot-line” telefon foglaltságát, felvette-e a kagylót valaki éjjel kettőkor stb. Sajnos, hazai viszonylatban erről még nem beszélhetünk.

Mint említettük, mindkét program ilyenén komfortja kívánnivalókat hagy maga után. Az ISIS dokumentációja alaposabb, igaz viszont, a gyártó által elkészített ún. online help-képernyők hiányoznak. A TEXTAR a képernyő alján állandóan jelenlévő instrukciós sorral segíti a munkát. Az ISIS apró előnye, hogy az adatbázis beviteli űrlapjának tervezésekor gondolhatunk a későbbi felhasználóra: egy, a definiált help-billentyű megnyomásakor bejelentkező rövid utasítás meghagyásával, mégpedig mezőnkénti lebontásban. Ügyes megoldás, hogy a hibaüzeneteket a program úgymond „saját-típusú” adatbázisban tárolja (három nyelven a menükkel együtt!), s ezek átírhatók, javíthatók.

Ezeknél azonban van egy lényegesen fontos szempont: az UNESCO-program teljes egészében kötődik a nagyvilágban kialakult online hagyományokhoz, tehát a máshol megtanultak kitűnően alkalmazhatók (vagy megemlíthető az IIF által támogatott nagygépes ISIS-gyakorlat felhasználhatósága is). S ez fordítva is igaz. Megfigyeltük, hogy az ISIS használatában már némi gyakorlatra szert tett kollégáink milyen gyorsan sajátítják el a legkülönbözőbb online és CD-ROM adatbankokban való keresés módját. A TEXTAR ezzel szemben sokszor speciális megoldásokkal él.

Végül is??

Természetesen nem tudunk s nem is kívánunk a címben feltett (ál)kérdésre válaszolni. Ugyanis a valódi kérdés inkább így hangzik: mit akarsz a számítógépeddel csinálni a könyvtárban? Erre talán már a fentiekből kiderült válaszuk. A TEXTAR-t - ha megszorításokkal is - alkalmasnak tartjuk kisebb (100.000 kötet alatt) könyvtárak integrált rendszerének ellátására, főleg az adatbevitelt támogató finom megoldásai miatt. Ezt külön alátámasztja a közművelődési könyvtárakat adatokkal ellátó Gazdasági Társulás³. Míg az ISIS-t mint könyvtárban is kitűnően hasznosítható általános szöveges-adatbáziskezelőt helyeznénk el a kínálati palettán.⁴

De bármelyiket is választjuk, ne feledjük: ezek személyi számítógépre irt programok, s így főként a személyes munkát támogatják, de egyik sem fogja megoldani a nagykönyvtárak több százézes rekordállományának hatékony kezelését, a napi többszáz kölcsönző folyamatos regisztrálását egyetlen integrált rendszerként. Azt sem szabad elfelejtenünk, AMI EGYSZER GÉPEN VAN MÁR, NEM VÉSZ EL, LEGFELJEBB ÁTALAKUL...

Jegyzetek

1. ERDŐS Iván - BISZAK Sándor: A FREETEXT szöveges információkereső rendszer. = Tudományos és Műszaki Tájékoztatás, 35. évf. 1988. 3.sz. 91-97.p.; JACSÓ Péter: A Micro-ISIS lehetőségei újabb változatainak tükrében. = Tudományos és Műszaki Tájékoztatás, 35. évf. 1988. 3.sz. 104-105.p.; Vö. legújabban: GERŐ J. Péter: TEXTAR. Egy hazai fejlesztésű szöveges adatbáziskezelő rendszer. = Könyvtári Figyelő, 36.évf. 1990. 3-4.sz. 194-222.p.
2. Köszönetet mondunk a szentendrei Pest Megyei Könyvtár két kitűnő munkatársának, Bazsó Gábornak és Kabai Ilonának, akik segítettek bennünket a TEXTAR-ral való megismerkedésben.

3. A szokásos TEXTAR elleni érvet, hogy igen drága, az ISIS meg lényegében ingyen beszerezhető, nem kívántuk bevenni szempontjaink közé.
4. vö.GERŐi.m.218.p.