

Bánki Zsolt – András Áron

Országos Széchényi Könyvtár

Változó feltételek – változó igények

Tapasztalatok és trendek az OPAC fejlesztés terén az Országos Széchényi Könyvtárban [1]

Az OPAC fogalma kibővült a virtuális, integrált katalógus irányába: helyi és távoli adatbázisokat szolgáltató közös, egységes felületen, úgy, hogy igyekeznek „eltakarni” a forrásadatbázisok strukturális eltéréseiből fakadó különbözőségeket. Az interfészek az információs portálok stílusának irányába fejlődnek az intelligens navigációk kialakításával, előválasztási lehetőségek kiterjesztésével. Szükségessé vált, hogy különböző igényeknek is megfeleljenek: például dokumentumot, illetve bibliográfiai adatokat kereső felhasználók számára az eltérő információkat relevánsan jelenítsék meg. A korszerű OPAC-ot fel kell készíteni valamennyi dokumentumtípus szolgáltatására: kéziratok, apró nyomtatványok, mikrofilmek, mozgóképek stb. Bemutatjuk a „rekordtérkép” ötletét is, és ismertetjük a vegyes (adatbázis, html és xml lapok) forrásokból épített virtuális katalógus megvalósításának technikai lehetőségeit – a ZEBRA segítségével.

Az Országos Széchényi Könyvtár (OSZK) 2000-ben indította útjára új integrált rendszerét, az AMICUS-t, és vele párhuzamosan korszerű OPAC-ját, a LibriVisiont [2]. Az azóta eltelt három év eseményei, fejlesztései, felhasználói tapasztalatai megmutatták azokat az irányokat, követelményeket, amelyek felé az informatikai fejlesztőknek tovább kell lépniük az OPAC korszerűsítése területén. Először ezeket a fejlesztési terveket mutatjuk be, majd ismertetjük a ZEBRA szoftver alkalmazását az OSZK-ban, amelynek segítségével különböző adatbázisaink egységes felületen való közös lekérdezhetőségét kívánjuk megoldani.

Fejlesztési terveink kiindulópontjai:

- 2002-ben felmérést végeztünk az OSZK-ban épülő online adatbázisokról. A vizsgálódás eredménye azt mutatta, hogy 2002-ben a nemzeti könyvtár integrált rendszerén kívül még háromféle adatbázis-kezelőben (MicroISIS, MsAccess, FolioViews) több mint 20 adatbázis épült [3].
- Az AMICUS adatbázis tartalmi bővülése.
- Hároméves felhasználói és fejlesztői tapasztalat a LibriVisionnel.

Az „egy OPAC – több adatbázis” elv kialakulása

A kilencvenes évek adatbázis-fejlesztési koncepciójának egyik alaptétele, amelyet az OSZK-ban is

így vallottunk, hogy egy intézmény gyűjteményének számítógépes feldolgozása egy közös nagy rendszerben történjen, illetve a korábbi kisgépes rendszerekben feldolgozott adatok előbb-utóbb bekerüljenek a központi rendszerbe. Az „egy könyvtár – egy rendszer” elvnek kétségkívül nagy előnyei vannak: a közös, egységes (többnyire MARC) adatstruktúra, a technikai műveletek generális kezelése, és (ennélfogva) a szolgáltatás egyszerűsége. Ez a felfogás annyira általánosan elfogadott volt, hogy még a hazai közös katalógus tervezésekor is a közös adatbázis mellett döntöttek, és a MOKKA nehézkes megvalósulásának az is egyik oka volt, hogy hatalmas feladatot jelentett a különböző rendszerekből érkező adatok betöltése a közös adatbázisba.

Az OSZK-ban elvégzett felmérés alapján tapasztalatunk azt mutatja, hogy a többféle PC-s rendszerben (MicroISIS, MsAccess, FolioViews) élő kis adatbázisok output formátumai, rekordszerkezetek, mezőstruktúrái alkalmatlanok a korrekt MARC konverzióra, és a MARC formátumot kezelő integrált rendszerekbe való betöltésre. Ugyanakkor ezek az adatbázisok jórészt OSZK-s állományokat tükröznek, illetve olyan nemzeti bibliográfiai feldolgozások (eRMK Szinnyei József: Magyar írók élete és munkái, Magyar Könyvészet 1712–1920, Latin nyelvű kötetes kéziratok katalógusa), amelyek szolgáltatása szintén nemzeti könyvtári feladat.

Az elmúlt években számottevően felgyorsult az internet hálózati sebessége, és széles körben elterjedt a Z39.50 protokoll használata, amely távoli adatbázisok lekérdezését teszi lehetővé közös felületen keresztül. Ez az eszköz még nem tette lehetővé a Z39.50-et nem ismerő kisebb adatbázis-kezelők elérését, ma azonban van olyan szoftverünk, amely képes szöveges állományokat indexelni, és Z39.50 gateway-en keresztül szolgáltatni. Ez a ZEBRA, amely imponáló sebességgel kezel hagyományos szövegfájlokat és XML-lapokat [4].

A Z39.50 megjelenése önmagában is új helyzetet eredményezett. A szolgáltatáshoz már nincs szükség feltétlenül közös adatbázisra, egységes felületről, egyetlen keresőkérdéssel elérhetünk különböző adatállományokat, és saját OPAC-unknak megfelelő formátumban tudjuk őket megjeleníteni. A ZEBRA segítségével – viszonylag egyszerű munkával – ugyanezt az eredményt tudjuk biztosítani azoknál a korábbi adatbázisoknál is, amelyek elérhetetlennek bizonyultak a közös lekérdezés számára [5]. Meggondolva azt, hogy mekkora terhet jelentett a könyvtár-informatikusoknak az ilyen-olyan szerkezetű PC-s adatbázisok egységesítése, konvertálása, már látjuk az új megoldás jelentőségét. Mindehhez tegyük még hozzá, hogy az intézmények igyekeznek megőrizni integrált rendszerük „tisztaságát”, amelyet a kényszerű konverziók igencsak megviselnek.

Ezek után kimondhatjuk: fel kell adnunk az „egy könyvtár – egy rendszer” elvet, és helyette az „egy OPAC – több adatbázis” felfogást kell alkalmaznunk, hangsúlyozva, hogy minden informatikai rendszer gerincét az integrált rendszernek kell alkotnia, és új adatbázisok építésébe nem, vagy csak nagyon indokolt esetben szabad kezdeni. Megoldási javaslatunk elsősorban a már korábbról örökölt adatbázisokra, illetve egyes speciális állományok retrospektív konverziójára vonatkozik.

A differenciált lekérdezés követelménye

Információs oldalak – adatbázis-választás

A LibriVision jelenleg kétféle típusú adatbázis elérését teszi lehetővé: a Z39.50 protokollal rendelkező adatbázisokét, és a webes felülettel rendelkező adatbázisokét. A második csoportot azonban nem tudjuk közös kereséssel lekérdezni, hanem a LibriVision felületről átléphetünk az adott, beköttöt adatbázisához, ott elvégezhetjük keresésünket, és visszatérhetünk a kiinduláshoz.

Amint az előzőekben jeleztük, az OPAC-nak több adatbázis közös lekérdező eszközévé kell fejlődnie. Ehhez azonban feltétlenül szükséges, hogy a rendszer világos, informatív eligazodást nyújtson az elérhető adatbázisokról, logikai adatbázisokról, azok jellegéről, adattartalmáról, valamint megfelelő, egyszerű módon, például jelölőnyezetek segítségével tetszőlegesen összeállítható legyen azon célobjektumok köre, amelyeket a későbbiekben megfogalmazandó kérdéssel el akarunk érni. Mindez egy áttekinthető, tartalmilag csoportosított, sok szempontú oldal létrehozásával érhető el, amelynek mintegy az OPAC belépési pontjaként el kell igazítania a felhasználót, hogy milyen szolgáltatásokat vehet igénybe.

A keresés előzetes szűkítése

Az AMICUS adatbázis indulásakor a Magyar Nemzeti Bibliográfia Könyvek Bibliográfiája által feldolgozott dokumentumokat tartalmazta 1976-tól kezdődően, illetve a Könyvtári Intézet Szakkönyvtára gyűjteményének egy részét, valamint egy OSZK-s különgyűjtemény által feldolgozott videodokumentumokat.

Mára ez a kör jelentősen kibővült a központi katalógus funkciót ellátó Nemzeti Periodika Adatbázissal, az 1989-től az OSZK állományába bekerülő külföldi könyvek adatbázisával (egykori KATAL adatbázis), és elindult a Szabad Európa Rádió gyűjteményének, és a Térképtár anyagának a feldolgozása is. Időközben megtörtént az Időszaki Kiadványok Bibliográfiájának a betöltése a rendszerbe, és elkezdődött a periodika állomány retrospektív konverziója. Felkészült a belépésre a Zeneműtár és a Kézirattár is. Ez a helyzet, ha megfontoljuk, hogy hányféle gyűjteményrészről, dokumentumtípusról van szó, rendkívül nagy feladatot ró az adatbázis szervezésére, és a visszakeresés feltételeinek intelligens kialakítására, hogy a bőség ne okozzon zavart, és az adatbázis ne nőjön áttekinthetetlen konglomerátummá, amelyben csak a tájékoztató szakemberek ismerik ki magukat. Meg kell teremteni annak a világos lehetőségét, hogy az egyes részadatbázisokat, illetve dokumentumtípusokat elkülönítve lehessen keresni. Mely eligazodási pontok jelenthetnek megoldást?

Az AMICUS ún. logikai adatbázisokat kezel. Ezeknek sajátos céljai és tulajdonságai vannak. Jelen vizsgálódásunk szempontjából az bír jelentőséggel, hogy önálló logikai adatbázisba van szervezve a Törzsgyűjtemény, az NPA és a Könyvtári Intézet Szakkönyvtárának állománya is. Ezen adatbázis-

részek önálló kereshetőségét biztosítani kell, és lehet is az OSZK OPAC-jának új, jelenleg fejlesztés alatt álló verziójában, hiszen a logikai adatbázisok önállóan lekérdezhetőek az AMICUS-ból.

Nehezebb feladatot jelent azonban az, hogy a Törzsgyűjtemény önmagában is többféle gyűjteményrészt és dokumentumtípust (könyvet, periodikumot, térképet, mikroformátumot, elektronikus dokumentumot stb.) egyesít magában. Ebben az esetben a logikai adatbázisra való szegmentálás nem elegendő. A LibriVision – bár némelyek bonyolultságára hivatkozva ezt gyengéjeként emlegetik – rendkívül részletes keresésre ad lehetőséget. Lehetőség van a kódolt MARC mezők (pl. 007 – kódolt fizikai jellemzők, 008 – meghatározott jellemzők és információs adatok) lekérdezésére is, amelyet a LibriVision felületen dinamikusan megjelenő indexsűgők feloldanak és értelmeznek. Ezen a ponton kínálkozik az a megoldás, hogy a keresőfelületre olyan előválasztó opciókat vetítsünk ki, amelyek mögött egy-egy dokumentumtípust definiáló MARC kódra vonatkozó kérdés húzódik meg. Így a felhasználó egyszerűen, közérthetően tud választani, az őt érdeklő dokumentumtípusra szűkítve a keresést anélkül, hogy elbújtatott indexekben kelljen kutakodnia.

„Rekordtérkép”

Az egyes különgyűjteményi dokumentumok számítógépes katalogizálási szabályzatainak kidolgozásakor szembesültünk azzal a ténnyel, hogy a speciális elvek szerint szervezett gyűjtemények, pl. a kéziratári fondok, vagy a produkciók köré csoportosított színháztörténeti dokumentumok bonyolult kapcsolatrendszerrel rendelkeznek. Feltárásuk néha három-négy hierarchiaszintet is tartalmaz. Az OSZK már eddig is alkalmazta a MARC-okban meglévő rekordkapcsolatokat, amelyekkel ugyan ki lehet fejezni az egyes rekordok közötti viszonyokat, de egy bizonyos pont után már áttekinthetlenné kezd válni a rekordkapcsolatok szövevénye, arról nem is beszélve, hogy a rendszer egy adott rekordnak egyszerre csak a közvetlen alá- és fölérendelt kapcsolatát mutatja meg, így soha nem láthatjuk egyben pl. egy fond teljes struktúráját. A hagyományos fondjegyzékhez szokott kutatók számára, akik jogosan szeretnék egyben áttekinteni az adott állományt, hátrányt jelent ez a megoldás. Ez az igény szülte meg az ún. „rekordtérkép” ötletét, amelynek lényege, hogy a találatok megjelenítésekor egy gomb segítségével új ablakban meg lehet nyitni az adott rekordhoz kapcsolódó összes kapcsolat ágrajzát, hasonlóan a fájlkezelők

könyvtárstruktúrájához, vagy a weblapok oldaltérképéhez. Így megjelennek az összefüggések, és a teljes struktúrát egységesen át lehet tekinteni. Ennek az ötletnek a megvalósítása jelenleg még csak kísérleti stádiumban van.

A szolgáltatások bővítésének szükségessége

Három évvel ezelőtt a LibriVision indításakor úgy gondoltuk, hogy olyan korszerű rendszer birtokosai lettünk, amely hosszú időre kényelmesen és színvonalasan biztosítja az OSZK számítógépes szolgáltatásainak elérését a helyi és távoli felhasználók számára. Ma már azonban látjuk, hogy egy percet sem ülhetünk ölbe tett kézzel, mert a LibriVision teljesíti ugyan azokat az elvárásokat, amelyeket a bevezetésekor kívántunk tőle, de a fentebb leírt szempontok és a rendszer mélyebb megismerése, amely hiányosságokat is feltárt, megkívánja az új verzió alapos átdolgozását, fejlesztését.

Gateway és katalógus

Horváth Ádám, az OSZK informatikai főigazgató-helyettese a 2002-es Networkshopon a virtuális katalógusról tartott előadásában egy olyan OPAC képét vázolta fel, amely kihasználja a Z39.50 protokoll biztosította lehetőséget, és adott esetben a hazai, illetve az európai (nemzeti) könyvtári adatbázisok lekérdezésének központi eszközévé válhat [6]. Ennek a célnak a LibriVision megfelel.

Hasonló funkciójú felhasználást látunk az Ausztrál Nemzeti Könyvtárban, ahol a LibriVision ezres nagyságrendű könyvtárak adatbázisainak közös számítógépes katalógusaként működik. A szoftver szerkezete is arról árulkodik, hogy a fejlesztők is elsősorban erre a feladatra szánták. Az OSZK ugyanakkor ezt az eszközt használja helyi katalógusként, saját állománya szolgáltatására is.

A két felhasználói körnek – akik távolról, interneten keresztül keresnek, vagy a könyvtárba látogató olvasónak – nyilvánvalóan más szempontjai vannak. A távoli felhasználót elsősorban az érdekli, hogy egy dokumentum mely könyvtárban található meg, vagy irodalomkutatást végez, és a bibliográfiai adatokra van szüksége, és csak másodsorban fontos számára az az információ, hogy egy bizonyos dokumentumnak mi a helyi státusa, elhelyezése, raktári jelzete, kölcsönözhetősége. A helyi felhasználót (olvasót) pedig primer módon a do-

kumentumhoz való minél gyorsabb, egyszerűbb hozzáférés motiválja, mi sem bosszantja jobban, mint egy eldugott raktári jelzet, amit három „fölösleges” kattintással lehet csak előbányászni.

Ezt a feszültséget a két funkció között a LibriVision remekül példázza. A szoftver ugyan logikusan épül fel, amikor együtt kezeli és jeleníti meg a MARC bibliográfiai rekord adatait, és külön oldalon közli az állományadatokat, az elhelyezést, kihelyezést, raktári jelzetet, a forgalmazhatóságra vonatkozó információt. Így azonban nehézkes a megtalálásuk különösen azoknál a találatoknál, ahol a raktári jelzet a kapcsolódó rekordnál található, és a kapcsolódó rekordhoz még el is kell jutni (például sorozat – sorozat tagja, vagy többkötetes közös adatai – kötet adatok esetén).

Szükségesnek tartjuk, hogy olyan OPAC felületet tudjunk konfigurálni, amely plasztikusan alakítható a kettős felhasználási cél szerint. Jelenleg a LibriVision ún. OSZKAT (OSZK Katalógus) felületén már az ún. „rövid címkés” megjelenítésben is szolgáltatni tudjuk a raktári jelzetet, de ez a rendszer sebességének rovására megy, és távolról nem vehető igénybe.

A felület megtervezésénél azt is figyelembe kell venni, hogy a tájékoztató szakember és kutató, valamint az átlag felhasználó más-más igényvel közelít az OPAC-hoz. Ezért legalább kétféle felületet kell elérhetővé tenni. A „kezdők” elől el kell rejtetni a részleteket, és ha ezzel információvesztést okozunk is, egyszerű, világosan áttekinthető kezelőfelületet kell biztosítani, míg a „haladók” részére minden választási lehetőséget meg kell jeleníteni.

Digitális dokumentumok szolgáltatása

Külön kezelendő feladat az online dokumentumok szolgáltatása az OPAC-on keresztül. Az interneten megjelenő és szabadon hozzáférhető dokumentumok egyre nagyobb vonzerőt jelentenek a felhasználók számára. Otthonról, munkahelyről gyorsan hozzáférhetnek, kereshetnek bennük. A LibriVision és az Amicus lehetővé teszi, hogy az online adatforrások is megjelenhessenek a bibliográfiai adatok között, ugrópontok (linkek) segítségével biztosítva a könnyű továbbhaladást. Aki azonban csak ezekre kíváncsi, annak az összes többi találat közül kell kikeresnie az online dokumentumokat. A már említett előválasztási lehetőségeknek arra is ki kell terjedniük, hogy mely elektronikus dokumentumok szabadon hozzáférhetőek. Ez

utóbbi már nehezen megoldható feladat. Az ilyen minősített metaadatok jelentik a könyvtári rendszerek óriási előnyét a „gyári” internetes keresőrobotokkal szemben.

Virtuális katalógus építése egyes (adatbázis, html és XML lapok) forrásokból

Fejlesztési elképzeléseink után az INDEXDATA nevű dán cég ZEBRA nevű ingyenes, nyílt forráskódú Z39.50 szerverét mutatjuk be, illetve azt, hogyan egészíthetjük ki ezzel a szoftverrel integrált könyvtár-informatikai rendszerünk szolgáltatásait. Ez az eszköz az integrált könyvtári rendszerbe nem illeszthető, viszonylag rossz struktúrájú, vagy hiányos szerkezetű rekordok bibliográfiai rendszerekben való kereshetővé tételére alkalmas Z39.50 protokollon keresztül (pl. html oldalak, vagy a katalógizálás érvényes szabályait figyelembe nem vevő olyan adatállományok, mint az OSZK Levelestár FolioViews-ban feldolgozott katalógusa).

Átfogó kép a Z39.50 protokoll tulajdonságairól és felhasználásáról

Tekintsük át, hogy milyen előnyökkel jár a Z39.50 protokoll használata [7]. Manapság adatainkat rendszerint relációs adatbázis-kezelő rendszerekben tároljuk, így gyorsan hozzájuk férhetünk, könnyen és többnyire ütközésmentesen módosíthatjuk őket, különféle kimutatásokat készíthetünk belőlük. A modern relációs adatbázis-kezelő rendszerekben az adatok többé-kevésbé azonos (szabványos) SQL utasításokkal is elérhetők. Mindennek ellenére a relációs adatbázis-kezelő rendszerben tárolt adatbázisok struktúrája szinte minden esetben eltérő: különbözőek a táblanevek, a táblákban szereplő oszlopok (az oszlopok nevei, az oszlopokban tárolt adatok típusai) stb. Ezért ha két különböző adatbázisból szeretnénk adatokat kinyerni, és a kinyert adatokat közös felületen, egységesen megjeleníteni, akkor szinte biztos, hogy két eléggé eltérő (ráadásul bonyolult adatbázisok esetén nagyon komplikált) SQL utasítást kell kiadni.

A Z39.50 protokoll ezen a problémán igyekszik segíteni. A Z39.50 szervert úgy lehet a legegyszerűbben elképzelni, mint egy már létező adatbázis-kezelő rendszer elé illesztett interfészt, amely a hozzá érkező kéréseket a mögötte álló adatbázisnak és adatszerkezetnek megfelelő formátumú kérésre alakítja. Az adatbázis felől érkező válaszokat pedig a hozzá befutott kérésnek megfelelő alakra hozva adja vissza. A protokoll használatával egységes kérést lehet intézni egy vagy több

Z39.50 szervertől: mutassa meg azokat a tételeket például MARC21 formátumban, amelyeknél a szerző nevében szerepel az „Arany” kifejezés, és a címben szerepel a „Toldi” szó és olyan kiadványok, amelyek „1950 után” jelentek meg. Ezt a kérést elméletileg változatlan formában fel lehet tenni két teljesen különböző Z39.50 szervertől. A gyakorlatban nem ilyen jó a helyzet, mert nem biztos, hogy mindegyik Z39.50 szervert támogatja a MARC21 formátumot, vagy képes az így megadott megjelenési évre keresni.

Kicsit mélyebben szemlélve a Z39.50 protokoll leíró specifikációt, észrevesszük azt is, hogy például a címre keresés paraméterei, egyszerű esetben egy attribútum – USE (1) attribútum – meghatározott számértékének (4), és a keresendő kifejezésnek megadásából áll:

```
find @attr1=4 „Toldi”
```

Létezik ugyan szabványos, előre definiált keresőkifejezés-készlet (USE attribútumok) a bibliográfiai rendszerekben történő keresésekhez (pl. BIB-1), de ez nem követeli meg teljes felhasználását, sőt lehetőséget nyújt a bővítésre, illetve végső soron az egész ajánlást fel lehet cserélni egy alapjaiban más rendszerrel. A címre keresést a 82-es USE attribútum megadásával is lehet kezdeményezni. Bár ez eltér a BIB-1 ajánlástól, a Z39.50 specifikáció keretein belül marad, azaz az adatbázist cím alapján bárki le tudja kérdezni, feltéve, ha tudja, hogy az adott adatbázisban a címre kereséskor a 4-es USE attribútum helyett a 82-est kell használnia:

```
find @attr1=82 „Toldi”
```

Láthatjuk tehát, hogy a Z39.50 protokoll használata sem egyértelmű. Mégis a modern könyvtár-informatikai rendszerek (Aleph, Amicus, Corvina, Olib, Voyager stb.) egytől egyig támogatják ezt a protokollt, azaz ezen a protokollon keresztül (megfelelő beállítások mellett) le tudják kérdezni egymás adatbázisait, illetve bármilyen más adatbázist, amelynek az adatai Z39.50 protokollon keresztül elérhetőek. Ilyenek például a ZEBRÁ-val létrehozott adatbázisok.

A ZEBRA szervert

A ZEBRA szöveges adatforrások (például: normál szöveg, címkékkel ellátott szöveg, MARC fájl, XML/SGML fájl) indexelésére, és Z39.50 protokollon

történő keresésére és megjelenítésére (MARC, XML, SUTRS stb. formátumban) alkalmas, megfelelő stabilitással és sebességgel. Ebből következően kiválóan alkalmas arra, hogy segítségével a már meglévő Z39.50 protokollt támogató könyvtár-informatikai rendszerünkhöz (OPAC-unkhoz) új adatforrásokat kapcsoljunk.

Mit érdemes ZEBRÁ-val szolgáltatni?

Az OSZK-ban jelenleg a következő állományokat dolgozzuk fel ZEBRÁ-val:

- Az ARCANUM Adatbázis Kft. által Folio Views-ban rögzített, elsősorban muzeális gyűjteményeink retrospektív elektronikus katalógusait. Ezek XML-szerű anyagok, de meglehetősen rosszul strukturáltak, és nem felelnek meg az érvényes könyvtári szabványok, szabályzatok követelményeinek.
- A MEK webről elérhető (html formátumú) állományjegyzékét.
- Az OSZK saját honlapjait.

Bár a ZEBRA sok lehetőséget kínál fel a források értelmezésére, mi azt a módszert követjük, hogy szükség szerint, egy-egy PERL script segítségével értelmezzük a forrásfájlokat, módosítjuk az adatokat, és a végeredményt egy SGML-szerű fájlba írjuk. A folyamat végeztével a ZEBRA az újonnan keletkezett SGML fájlt fogja indexelni, illetve az abban szereplő adatokat fogja visszaadni. Az SGML formátum a legalkalmasabb arra, hogy mind a ZEBRA, mind pedig a rendszer működését felügyelő szakember a legtöbb információt nyerje ki a forrásadatokból.

Adatok indexelése és szolgáltatása a ZEBRA segítségével

A ZEBRÁ-val kapcsolatos rendszergazdai tevékenységek közül a legfontosabb és legbonyolultabb feladat az egyes adatbázisok konfigurálása az indexeléshez és a szolgáltatáshoz. A folyamat azért összetett, mert minimális igények esetén is legalább 3-4 konfigurációs fájl tartalmát ajánlott átnézni, illetve módosítani, míg bonyolultabb esetekben a konfigurációs fájlok száma könnyen 6 fölé emelkedhet, sőt előfordulhat, hogy ezek közül jó néhányat nekünk kell létrehozni. Természetesen a „konfigurációs fájlok” egymásra épülnek, így az sem mindig triviális, hogy milyen sorrendben nézzük át őket. Tapasztalataink alapján a következő sorrendet alakítottuk ki:

1. zebra.cfg
2. TESZT.tag

3. TESZT.abs
4. TESZT.att
5. TESZT-usmarc.map
6. string.chr

Magyarázatra szorul, hogy mik azok a TESZT*.* fájlok, és miért éppen a TESZT szóval kezdődik a nevük. A válasz a ZEBRA sajátos viselkedésében található: SGML fájlok indexelésekor a szoftver úgy jár el, hogy a legfelsőbb szintű tag nevével megegyező nevű „.abs” kiterjesztésű konfigurációs fájlban keresi az indexelés paramétereit, illetve az indexeléshez és a szerver funkciók működéséhez szükséges konfigurációs fájlok neveit, ezért indexelendő adatbázisunknak a következőképpen kell kinéznie:

```
<TESZT>
<SZERZO>
<VEZETEKNEV>Andrási</VEZETEKNEV>
<KERESZTNEV>Áron</KERESZTNEV>
</SZERZO>
<CIM>Z39.50 használati útmutató</CIM>
<URL>http://www.oszk.hu/1.html</URL>
</TESZT>
<TESZT>
<SZERZO>
<VEZETEKNEV>Palyik</VEZETEKNEV>
<KERESZTNEV>Katalin</KERESZTNEV>
</SZERZO>
<CIM>Amicus rendszeradminisztrátori kézikönyv</CIM>
<URL>http://www.oszk.hu/2.html</URL>
</TESZT>
```

Elvileg elégséges lenne, ha csak az „.abs” kiterjesztésű fájlunk neve egyezne meg SGML „adatbázisunk” legelső elemének nevével, ha azonban több adatbázist akarunk beindexelni, és mindegyikhez egyedi konfigurációs fájlok tartoznak, akkor célszerű az összetartozó fájloknak hasonló (egységes) nevet adni.

zebra.cfg

```
profilePath: [konfigurációs_(tab)_fajlok_elérési_útvjai]
attset:      bibl.att
attset:      explain.att
recordType:  gsr.sgml
lockDir:     lock
setTmpDir:   tmp
keyTmpDir:   tmp
# ... [egyéb beállítások például a memóriakezelésre
vonatkozóan]
```

A zebra.cfg fájlban állíthatjuk be, hogy a ZEBRA mely könyvtárakban keresse az indexeléshez, és a rekordszolgáltatáshoz szükséges konfigurációs fájlokat, illetve hogy az ezen folyamatok által ideiglenesen létrehozott fájlok mely könyvtárakba kerüljenek. Itt rendelkezhetünk a memóriahasználatról is.

TESZT.tag

type 4	[sgml_tagnév]	[típus]
# tag [sorszám]		
tag 1	SZERZO	structured
tag 2	VEZETEKNEV	string
tag 3	KERESZTNEV	string
tag 4	CÍM	string
tag 5	URL	string

Ahhoz, hogy az indexelendő fájlunkban lévő adatokat értelmezni tudjuk, egyedi azonosítókkal kell ellátnunk a benne szereplő (számunkra fontos) adatelemeket (XML taget), illetve meg kell határoznunk a típusukat is. Ezt végezhetjük el a TESZT.tag fájlban.

Az egyedi azonosítók (amelyeket később a TESZT.abs fájlban használunk fel) két részből állnak:

- a konfigurációs fájl elején a „type” kulcsszó után szereplő számból,
- a később felsorolt tag kulcsszavak után szereplő számokból.

Vagyis a <SZERZO> tag által jelölt adatra a TESZT.abs fájlban (4,1)-ként fogunk hivatkozni, míg a <SZERZO><VEZETEKNEV> tag által jelölt adatra (4,1)/(4,2)-ként.

TESZT.abs

```
reference GILS-schema
attset TESZT.att
tagset TESZT.tag
maptab TESZT-usmarc.map
esetname B @
esetname F @
all any
# elm ([type], [zebra_elnév] [index_neve:
[sorszám]) index_típusa]
elm (4,1) szerzo -
elm (4,1)/(4,2) vezeteknev nev:w
elm (4,1)/(4,3) keresztnev nev:w
elm (4,4) cím !:w,!:p
elm (4,5) url -
```

Az indexelés „mikéntjét” a TESZT.abs fájlban határozhatjuk meg. A TESZT.tag fájlban megadott azonosítók felhasználásával megmondhatjuk, hogy az egyes adatokat hogyan indexelje a rendszer (szavanként és/vagy kifejezésenként stb.), az indexekhez neveket rendelhetünk, amelyek alapján a TESZT.att fájlban hivatkozhatunk rájuk (ott definiáljuk, hogy egy meghatározott Z39.50 use attribútum beérkezésekor milyen indexben keressen a ZEBRA). Ha két adatelemhez ugyanazt az indexnevet rendeljük, akkor a ZEBRA ennek megfelelően mindkét adatelem tartalmát beteszi egy közös indexbe. Arra is lehetőségünk van, hogy egy adatelem tartalmát egyszerre több indexben is szerepeltessük.

A fent említett indexelési lehetőségen kívül itt adhatjuk meg azt az azonosító nevet (a fent látható mintában [zebra_elvezés] ennek az oszlopnak a neve), amely alapján az adatelemet a megjelenítéskor használt *.map fájlban (jelen példában a TESZT-usmarc.map fájlban) azonosíthatjuk. Ebben a fájlban határozzuk meg az indexeléshez, illetve a rekordszolgáltatáshoz szükséges többi konfigurációs fájl nevét is.

Az „all any” direktíva megadásával „létrejön” egy „any” nevű index, amely tartalmazza az összes általunk definiált indexet. Példánkban ha az any indexben keresnénk, akkor az azt jelentené, hogy egyszerre keresünk a vezetéknevek, keresztnévek és címek között.

Az 'esetname' kulcsszó segítségével rendelkezhetünk arról, hogy a rövid (Brief) és teljes (Full) megjelenítéskor mely adatelemeket küldje vissza a szerver. A @ paraméter megadásával minden adatelem szerepelni fog a szerver által küldött válaszban, míg a @ helyére egy fájlnevet írva a csak fájlnev által jelölt fájlban szereplő 'simpleelement' tagek paraméterei szerepelnek a válaszban, amelyek nagyjából megegyeznek az 'elm' tagek paramétereivel, azaz a TESZT.tag fájlban megadott azonosítók sorozatából áll. Példaképpen nézzünk meg egy lehetséges TESZT.b.est fájl tartalmát:

```
simpleelement (4,1)/(4,2)
simpleelement (4,1)/(4,3)
simpleelement (4,4)
```

TESZT.att

reference Bib-1		[index_neve]
# att [Z39.50_use_attribútum]		nev
att 1003		cim
att 4		any
att 1016		

A TESZT.att fájlban adjuk meg, hogy a TESZT.abs fájlban meghatározott indexek milyen Z39.50 use attribútumon keresztül legyenek elérhetőek. A fent látható példában ha „név”-re szeretnénk keresni, akkor az 1003-as use attribútumot, és persze egy nevet kell megadni a keresőkérdésben. Pl.:

```
find @attr1=1003 „Arany”
```

Fontos megjegyezni, hogy a TESZT.att fájlban szerepelnie kell az összes TESZT.abs fájlban létrehozott indexnek, egyébként hibaüzenetet kapunk.

TESZT-usmarc.map

targetname usmarc		[MARC_kód]
targetref usmarc		
# map [elvezés]		
map vezetéknev		/(3,100)/(3,a)
map keresztnév		/(3,100)/(3,j)
map cím		/(3,245)/(3,a)
map url		/(3,856)/(3,u)

A TESZT-usmarc.map fájl nevéből következően ahhoz nyújt segítséget, hogy a felhasználó által kért rekordot a ZEBRA USMARC formátumban küldje vissza. Tapasztalatunk szerint a ZEBRÁ-n belül semmi nem ellenőrzi, hogy az itt megadott beállítások megfelelnek-e a USMARC előírásainak, vagy sem, így akár HUNMARC formátumot, vagy bármi mást is be lehet állítani USMARC címén. Erre szükség is lehet, ugyanis a targetname, illetve targetref direktívák HUNMARC-ra állításától a ZEBRÁ-nak elvileg támogatnia kellene a HUNMARC formátumot, ám próbálkozásaink során arra a következtetésre jutottunk, hogy míg DANMARC és USMARC beállításokkal jól működik a rendszer, addig a HUNMARC formátumot ismeretlennek tekinti.

Igen érdekes, bár számunkra kissé kódós az a mód, ahogy ebben a konfigurációs fájlban a

TESZT.abs fájlban megadott és általunk [zebra_ elnevezés] néven jelölt adatelem azonosítóhoz (a TESZT-usmarc.map fájlban [elnevezes]-ként szerepel) hozzárendeljük a MARC hívójelet és az almezőkódot. Példánkban a „vezeteknev”-vel jelzett adathoz a „100”-as marc hívójel „\$a” almező kódját rendeltük, de miért kellett ezt „/(3,100)/(3,a)” formában megadni egy egyszerű „(100,a)” helyett, ez egyelőre rejtély.

string.chr

A ZEBRA nyugat-európai szoftver, ezért alapesetben az ASCII kódtábla szerint rendezi az indexelendő adatokat. A magyarországi könyvtárakban viszont az MSZ 3401 *Bibliográfiai tételek betűrendbe sorolásának szabályai* című szabvány szerinti rendezés használatos, ezért ezt a rendezést külön specifikálnunk kell a ZEBRA számára a string.chr fájlban.

```
lowercase {0-9}{a-z}
uppercase {0-9}{A-Z}
space {'001-040}'!#$%&'\()*+,-./:;<=>?@[\\]^_`{|}~
# map (mit) mire
map (á) a
map (é) e
map (i) i
map (ó) o
map (ö) o
map (õ) o
map (ú) u
map (ü) u
map (ű) u
map (Á) A
...
```

A lowercase és uppercase kulcsszavak után fel kell sorolni az indexelendő rekordokban előforduló karaktereket olyan sorrendben, ahogyan azokat rendezni szeretnénk. Ha az MSZ 3401 szabvány előírásait követjük, akkor itt bele is botlunk abba a problémába, hogy egyes karaktereket, mint például az 'e' és az 'é', a rendezés szempontjából egyenértékűnek kell tekintenünk. Természetesen ez is definiálható a ZEBRA számára a map kulcsszó segítségével, csak arra kell ügyelni, hogy a mapelni kívánt betű (a példában a „mit” oszlop elemei) ne szerepeljen a lowercase és uppercase listában, de az a betű, amelyre mapelni szeretnénk (a példában a „mire” oszlop elemei), feltétlenül szerepeljen.

A példában szereplő mapelésnek az lesz az eredménye, hogy ékezetesen és ékezet nélkül is tu-

dunk keresni, illetve böngészni, de böngészéskor csak ékezet nélküli alakban jelennek meg a kifejezések.

Indexelés indítása

```
# zebraind -d [adatbázis_neve]
update [indexelendő_fájlok_helye]
zebraidx -d TESZT
update records
```

Ha az előzőekben megemlített konfigurációs fájlokat beállítottuk, akkor elindíthatjuk az indexelést. Az indexelő programnak megadható paraméterek közül a két legfontosabb, hogy mit indexeljen, és hogy milyen „logikai adatbázisnéven” legyen elérhető az így keletkezett index.

ZEBRA szerver indítása

```
# zebrasrv -l [log_fájl_neve] tcp:[ip]:[port]
zebrasrv -l log/index.log tcp:@:9999
```

Az indexelés befejezése után el kell indítani a ZEBRA szervert. Az indító parancsban célszerű megadni, hogy milyen IP címen, és milyen porton legyen elérhető a szerver. Emellett a szerver által készített logot is tanácsos valamilyen fájlba írni.

Saját kliens létrehozása

Nemcsak egy Z39.50 szerver áll ingyenesen a rendelkezésünkre, hanem kliensek és klienskészítő eszközök egész sora. Az INDEXDATA például a következő jelentősebb eszközöket kínálja:

- YAZ és YAZ++: C, illetve C++ függvénykönyvtár plusz parancssori Z39.50 kliens;
- PHP/YAZ (PHP extension): PHP alapú (főleg) webes Z39.50 kliens fejlesztéséhez;
- ZAP! (Apache modul): szintén webes kliens fejlesztéséhez;
- IRTCL (Tcl/Tk alapú GUI): platformfüggetlen grafikus környezetben futó desktop alkalmazáshoz.

A Z39.50 protokoll jövője

A Z39.50 protokoll fejlődése szerencsére nem állt meg. A Library of Congress vezetésével kidolgozták a Z39.50 szabvány legújabb változatát (Z39.50 Initiative Next Generation, röviden ZING [8]),

amely a mai informatikai trendnek megfelelően webszolgáltatásként képzelel el a Z39.50 klienszerver funkciókat, és ennek megfelelően új eszközökkel és fogalmakkal (például XML, SOAP, WSDL) gazdagítja a Z39.50 protokoll eszköz- és fogalomkörét. Már megjelentek az első JAVA és egyéb nyelveken készült implementációk, amelyek közvetve elérhetők a Library of Congress weboldaláról.

Irodalom

[1] http://nws.iif.hu/nws2003_video/index.html (A 2003. évi Networkshop konferencián elhangzott előadás szerkesztett változata)

- [2] BÁNKI Zsolt: LibriVision az OSZK új webes katalógusa. = Az előadás elhangzott a 2001. évi Könyvfesztiválon.
- [3] BÁNKI Zsolt: ONLINE szolgáltatások a Nemzeti Könyvtárban. = MERCURIUS 2002: Országos Széchényi Könyvtár, 2002. p. 7–9.
- [4] <http://www.indexdata.com/>
- [5] HEGYI Ádám – SÁNDOR Ákos: Folyóirat-indexelés ZEBRÁ-val. = <http://nws.iif.hu/ncd2001/docs/eloadas/10/index.htm> (Az előadás elhangzott a 2001. évi Networkshop konferencián)
- [6] HORVÁTH Ádám: Virtuális katalógus építése az OSZK-ban. = <http://nws.iif.hu/ncd2002/> (Az előadás elhangzott a 2002. évi Networkshop konferencián)
- [7] <http://www.loc.gov/z3950/agency/document.html>
- [8] <http://www.loc.gov/z3950/agency/zing/>

Beérkezett: 2003. VIII. 6-án.

Rendezvénynaplár

IADIS WWW/Internet 2003, nemzetközi konferencia

Algarve (Portugália), 2003. november 5–8.
Szervező: IADIS (International Association for Development of the Information Society)
E-mail: secretariat@iadis.org
URL: <http://www.iadis.org/icwi2003>

Digitális tájékoztatás, a VRD 5. éves konferenciája

San Antonio (Texas, USA),
2003. november 17–18.
Szervező: VRD (Virtual Reference Desk)
E-mail: vrldconf@vrd.org
URL: www.vrd2003.org

INFOtrend nemzetközi informatikai és telekommunikációs konferencia

Budapest, 2003. november 27–29.
Szervező: Hungexpo Rt.
Tel.: 263-6065
E-mail: infotrend@hungexpo.hu
URL: www.hungexpo.hu

Content Management Europe 2003, konferencia és kiállítás

London, 2003. december 2–4.
Szervező: Vernon Tolson
Business Development Manager
Tel.: +44 1932 730735
E-mail: Vtolson@imark.co.uk
URL: <http://www.cme-expo.co.uk>

Digitális könyvtárak. Nemzetközi konferencia (International conference on digital libraries: knowledge creation, preservation, access and management)

Újdelhi, 2004. február 24–27.
Szervező: ICDL 2004 Secretariat, TERI,
Darbari Seth Block, Habitat Place, Lodhi Road,
New Delhi – 110 003
India
Fax: +99 11 24682133
URL: <http://www.teriin.org/events/icdl>