Adatbázis-alapú webhelyek archiválása és rekonstruálása

A jelenlegi webhelyek jó része statikus jellegű, vagyis előre elkészített HTML fájlok alkotják őket, s ezeket esetleg Flash formátumú vagy JavaScript nyelven írt dinamikus elemek egészítik ki. Megőrzési szempontból az ilyen site-ok viszonylag egyszerűen begyűjthetők, tetszőleges mélységig, olyan robotokkal, mint például a HTTrack, vagy egyszerűen elkérhető az egész fájlrendszer a webmestertől – tehát ezeknek már kialakult archiválási technológiája van. [1] [2] Ám a W3Techs 2012-es felmérése [3] szerint a webhelyek mintegy 30%-át már valamilyen tartalomkezelő rendszer (CMS = Content Management System) szolgáltatja, melyek mögött általában egy adatbázis működik, ami a tartalom nagy részét – többnyire a szöveges elemeket és részben a formázást is – tárolja. Hogy a felhasználó számára hogyan jelenik meg az ebből a háttéradatbázisból „röptében", dinamikusan generált tartalom, az függhet a felhasználó böngészőjétől és egyéb paraméterektől is.

Egy ilyen CMS-alapú rendszert nem tudunk a megszokott módon jól archiválni, mert az a nézet, amit a robot begyűjt, csak egy a többféle lehetséges közül, továbbá az adatbázist sem lehet rekonstruálni a letöltött weboldalak alapján. Ilyenkor még az adatbázis és a statikus (pl. multimédia) állományok elkérése sem elegendő, mert ugyan így fájlszinten megvalósul az archiválás, de az eredeti tartalomszolgáltatás ezekből még nem állítható vissza. A különböző adatbázis-kezelő rendszerek beépített exportfunkciója is többnyire olyan bináris vagy gyártófüggő formátumba ment, amiből nem építhető vissza az adatbázis egy másfajta adatbázis-kezelőt használó archívumban. Az igazi megoldás az összes adat olyan formátumba való konvertálása volna, amely független az eredeti szerveren a háttérben futó adatbázisrendszertől. A DeepArc nevű eszközzel [4] – egy megfelelően kialakított XML sémát használva – az adatbázisok tartalma XML fájlokra képezhető le, így biztosítható a rendszerfüggetlenség. Ennek a megoldásnak az a hátránya, hogy a séma definiálásához alaposan ismerni kell az archiválandó adatbázist. Továbbá a DeepArc-ot, vagy más hasonló leképező eszközt az eredeti szolgáltató gépen is telepíteni kell és ott hosszú ideig elérhetőnek kell maradnia. Az előbbiekben vázolt bizonytalan helyzet és egy egységes és egyszerű adatbázis-archiválási formátum hiánya komoly problémát jelent a web hosszú távú megőrzésével foglalkozó intézményeknek.

A dinamikus webhelyek archiválásának jelen cikkben bemutatott modelljében minden archivált tartalmat egységes XML formátumra konvertálnak, amiből automatikusan rekonstruálható az eredeti site. Az XML fájlok helyességét és hibátlanságát egy XML séma segítségével bármikor ellenőrizni lehet az archívumban. A DeepArc-féle megoldásoktól eltérően ennél a megoldásnál az XML séma nem függ az archiválandó webhelytől, nem kell belülről ismerni az azt működtető rendszert, és nem kell az eredeti szerveren a rendszergazdának telepítenie egy célszoftvert.

Az archiválási folyamat során háromféle adat keletkezik *(1. ábra)*:
- a webhelyet leíró metaadatok,
- az XML formátumra leképezett statikus fájlok,

az XML formátumra leképezett adatbázisadatok.

A metaadatok egyrészt az eredeti szerver jellemzőit (pl. az operációs rendszer típusát), másrészt az archiválásra vonatkozó információkat (pl. a letöltés időpontját) tartalmazzák.

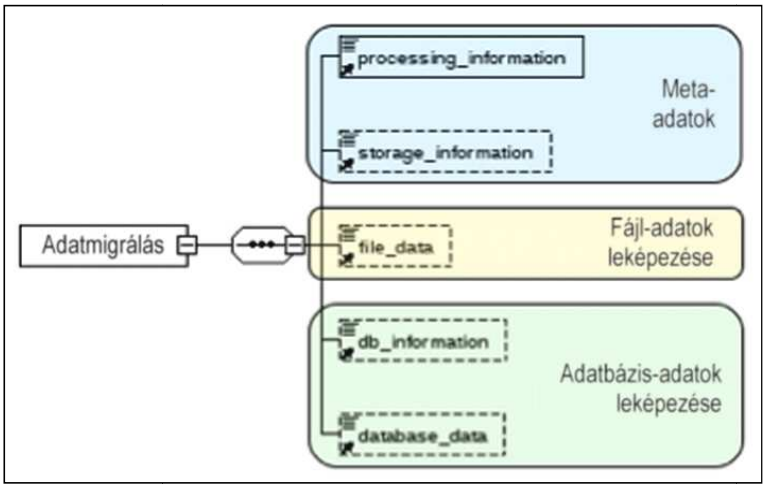
A statikus állományok esetében azok tartalma az ábrán látható *file_data* elemen belül egy *file* nevű elembe kerül (BASE64 kódolással, mivel az XML fájlokban nem lehetnek bináris karakterek), az eredeti fájlnév és relatív elérési útvonal pedig a *file* elem attribútumaként van eltárolva.

A háttéradatbázisból származó adatok leképezése és eltárolása már némileg bonyolultabb folyamat: A dinamikus webhelyekhez jellemzően relációs adatbázisokat használnak és SQL nyelven zajlik

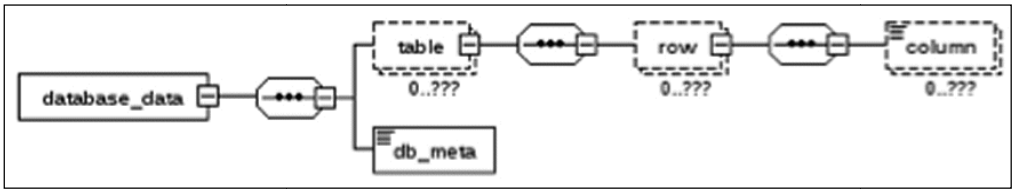
velük a kommunikáció. Az eredeti ISO 9075 szabványt azonban sok gyártó kibővítette, így a rendszerfüggetlenség eléréséhez ezeket a bővítéseket szükség esetén mappelni kell standard SQL elemekre. Az adatbázis begyűjtésekor az első lépés a táblák és a köztük levő relációk kielemzése. (A körkörös hivatkozások például gondot okozhatnak és ezért ezekkel külön foglalkozni kell.) Az eredeti adatbázis típusa, gyártója és verziószáma a *db_information* nevű elemben tárolódik, míg maguk az adatok az XML fájl *database_data* szekciójába kerülnek, különböző elemekre bontva *(2. ábra)*. A *db_meta* elem az archivált adatbázis jellemzőit tartalmazza (pl. méret, táblák és táblasorok száma). A táblákban levő adatok *table* elemekbe kerülnek, melyek mindegyike akárhány *row* elemet tartalmazhat, ezek pedig *column* elemekből állnak. Az eredeti adatbázistáblák egy-egy cellájában található adatok tehát ilyen *column* elemekként lesznek eltárolva az archív XML fájlban, és mivel egyes adatbázisokban akár képek vagy egyéb bináris adatok is tárolhatók, ezért a begyűjtési folyamat során ezek is átesnek egy BASE64 konverzión.

Az eredeti tartalom visszanyerése az archív XML fájl feldolgozásával történhet; erre a célra az ingyenes XML értelmezők (*parser*-ek) is használhatók. A *parser* soronként végigolvassa az XML állományt és különféle műveleteket végez el az egyes elemek tartalmával azok attribútumainak függvényében. Az archivált statikus fájlok esetében például nevük és relatív útvonaluk kiolvasása után dekódolja és létrehozza őket az adott néven és helyen. Az adatbázistartalmak esetében pedig a táblák visszaállítását végzi el soronként haladva. A *table* elemen belül az *Element-Start* címke jelzi, hogy egy új táblát kell létrehozni a megfelelő oszlopokkal és relációkkal.

A fenti modellt megvalósító hardver- és szoftverfüggetlen archiválási technológia többféle programozási nyelven is elkészíthető. Az első prototípus Java nyelven íródott, mivel ez sokféle platformon elérhető és jó az XML feldolgozási képessége. Az eredmény egy kisméretű applet, melyet ki lehet még egészíteni driverekkel a különféle adatbázisokhoz. Az alkalmazás használatához szükség van a JVM (Java Virtual Machine) jelenlétére az eredeti szolgáltató gépen, de a későbbi verziókban még ez az előfeltétel is kiküszöbölhető lesz. Az archiválandó site jellemzői egy egyszerű konfigurációs fájlban adhatók meg. (Grafikus felület még nem készült ehhez a konfigurálási fázishoz.) Ezzel a technológiával automatizált, ütemezett mentések készíthetők adott webhelyekről.



1. ábra **Az archiváláshoz használt XML séma vázlata**



2. ábra **Az adatbázisadatok tárolási szerkezete az XML fájlban**

A szoftver, bár még csak prototípus, lényegében egy teljes értékű, működőképes archiváló eszköz, melyet már különböző platformokon (Windows, Linux, Openembedded OS) és különféle feladatokkal teszteltek sikeresen. A tesztek közt volt eltérő adatbázis-kezelők közötti migráció: például MySQL-ből PostgreSQL-be, SYBASE ASE-ből PostgreSQL-be, illetve MySQL-be (és visszafelé is), továbbá adott gyártó különböző rendszerei közötti adatcsere is. A tesztelés még olyan adatbázis-kezelőkre is kiterjedt, amelyeket nemigen használnak webszolgáltatáshoz (MS Access és SQLite). Az archivált site-ok egy része statikus, másik része CMS-alapú volt, illetve akadtak köztük Java Servlet Container (Apache Tomcat) által szolgáltatott webhelyek is. A begyűjtött adatok mennyisége a néhány Mbyte-tól a több Gigabyte-ig terjedt – utóbbi egy csaknem 200 táblából álló SYBASE ASE adatbázis volt, amelynek a legnagyobb táblája kb. 1 millió sorból állt.

Egy CMS-alapú, kb. 2800 statikus fájlból (összesen nagyjából 60 Mbyte) és egy 100 táblás, 15 000 soros adatbázisból álló webhely archiválása, majd helyreállítása nagyjából 40 másodpercet vett igénybe (az idő persze nagyban függ a hardver teljesítményétől). Egy lényegesen nagyobb méretű teszt is sikeresen futott le: ennek során egy több Gbyte-nyi statikus fájlrendszert és egy 15 millió soros, kb. 15 Gbyte-os adatbázist dolgoztak így fel – tehát várhatóan ez az archiválási technológia a legtöbb webhely esetében használható lesz. Korlátot csak az archív XML dokumentum mérete jelent, de mivel a legtöbb korszerű fájlrendszerben (pl. EXT4, NTFS, XFS) több terabyte vagy exabyte a lehetséges legnagyobb fájlméret, így ez sem valódi korlátozás.

A szokásos archiválási formátumok (mint például a ZIP) többnyire bináris fájlok, amelyeknél akár egyetlen meghibásodott bit használhatatlanná teheti az egész állományt. Ennél az XML-alapú archívumnál viszont a sérülés csak az adott elemet teszi tönkre, vagyis a visszaállításkor legfeljebb egyetlen fájl vagy az adatbázis egy sorának egyetlen oszlopa lesz hibás.

Mivel a BASE64 kódolásnak köszönhetően az XML állomány csak nyomtatható karaktereket tartalmaz, egy adott site különböző idejű mentéseinek gazdaságos tárolásához a Git vagy más hasonló verziókezelő rendszer (VCS = Version Control System) használható. A VCS csak az archív dokumentum egyes verziói közötti eltéréseket tárolja, és képes visszaállítani a dokumentum állapotát bármelyik elmentett időpontban, amiből azután rekonstruálható az eredeti webhely adott idejű pillanatképe, és megjeleníthető a felhasználó számára is, például egy virtuális szerveren futó Archive Content Viewer alkalmazással.

A „kézzel” indított, illetve az automatikus, időzített archiválási, valamint az archív állományból való rekonstruálási folyamatokhoz ellenőrzőlisták állnak rendelkezésre, amelyek megkönnyítik a szükséges beállítási lépések helyes elvégzését. További hasonló listák, konfigurálási segédeszközök, mintapéldák és a szoftverdokumentáció készítése folyamatban van. (Ezek közvetlenül a szerzőtől megkaphatók.) A további tervek közt szerepel egy grafikus kezelőfelület a kevésbé gyakorlott szakembereknek, amellyel parancsok begépelése nélkül is elindítható az archiválási vagy a visszaállítási folyamat; valamint egy olyan eszközkészlet, amely megkönnyítené az archiválási lánc felállítását és karbantartását (beleértve a VCS integrálását is); továbbá egy Archive Content Viewer, ami szintén egy hasznos továbbfejlesztés volna.

Hivatkozások

[1] BROWN, A.: Archiving websites: A practical guide for information management professionals. Facet Publishing, London, 2006, ISBN: 978-1856045537.

[2] PINSENT, E. et al.: The preservation of web resources handbook. ULCC, London, 2008. *http://jiscpowr.jiscinvolve.org/wp/files/2008/11/powrhandbookv1.pdf*

[3] Q-Success: Usage of content management systems for websites, 2012. *http://w3techs.com/technologies/overview/content_management/all*

[4] *http://deeparc.sourceforge.net*

/RUMINAEK, Michael: Archiving and recovering database-driven websites. = D-Lib Magazine, 19. köt. 1–2. sz. 2013.
http//www.dlib/org/dlb/january13/rumianek/01rumianek/html

(Drótos László)