

# Interaktív televíziós alkalmazások átvitelének vizsgálata az Antenna Hungária DVB-T platformján

KRÉMER SZABOLCS

Antenna Hungária Rt., Fejlesztési és Informatikai Ágazat  
kremersz@ahrt.hu

**Kulcsszavak:** DVB, MHP, interaktív televízió, adatkörfolyam, karusszel

A televíziózás része mindennapi életünknek, mely most forradalmi változáson megy át. A digitalizálás révén a szélesebb műsorválaszték mellett a néző az interaktív alkalmazások segítségével többlet-információkhoz juthat. A cikk célja az interaktív alkalmazások átviteléhez szükséges rendszer bemutatása mellett az átvitel során alkalmazott adatstruktúrák és sávszélesség-optimalizálási eljárások ismertetése.

## 1. Bevezetés

Az interaktív televíziós alkalmazások átvitelének vizsgálata előtt érdemes tisztázni, hogy az interaktivitásnak ebben a televíziós környezetben milyen jelentései és lehetőségei vannak. A hagyományos analóg műsor-szórásban a hang és kép mellett teletext információkat is átvisznek. Ezek az adatok minden készülékhez eljutnak függetlenül attól, hogy a néző kérte vagy sem. A néző a távvezérlő segítségével meghatározhatja, hogy a továbbított oldalak közül melyiket kívánja megtekinteni. Ezt a lehetőséget a digitális televíziós rendszerekben lokális vagy alap-interaktivitásnak nevezik. A digitális rendszerekben a teletext szolgáltatás analógiájára megvalósított „szuper-teletext” rendszerekben a néző lehetőségei hasonlóak.

Az interaktivitás következő lépcsőfokán a néző a vevőberendezésben lévő visszirányú csatorna (GPRS vagy analóg modem) segítségével kérhet adatokat. Ezeket az adatokat csak az adott néző számára küldik el, annak modemes vagy GRPS kapcsolatán keresztül. Az interaktivitás harmadik fokán a néző a teljes körű Internet elérés minden előnyét élvezheti. A jelenleg alkalmazott rendszerek csak az alap-interaktivitást teszik lehetővé.

Az interaktivitás második lépcsőfokát megvalósító rendszerek fejlesztése folyamatban van és a közeljövőben várható az elterjedésük. A teljes Internet elérés megvalósításának fő akadálya jelenleg a vevőberendezés technológiája. Léteznek PC alapú vevőberendezések is amelyek képesek az Internet elérésre is, de a hagyományos vevődekóderekhez képest magas az áruk, így széles körű elterjedésük a közeljövőben nem várható.

Az itt tárgyalt rendszer a digitális televíziós környezetben az alap-interaktivitás megvalósításához szükséges elemeket tartalmazza, és a mindenkire eljutott információ hatékony továbbítási módszerével foglalkozik. Ezek a módszerek felhasználhatók az interaktivitás következő szintjeinek megvalósításakor is a hatékony információtovábbításhoz.

## 2. DVB és MHP rendszerek célkitűzése

A DVB Projekt keretében már számos nyílt szabványt definiáltak. Ezek segítségével a műsorszórás és interaktív szolgáltatások megvalósíthatók minden átviteli hálózaton: műholdon, kábelen, földfelszínen, mikrohullámon. Az utóbbi években a házi multimédiás platform (MHP) specifikációinak harmonizálása a DVB Projekt legfontosabb tevékenysége. Ezen belül kiemelkedik az alkalmazások interoperabilitása, letölthetősége, skálázhatósága és bővíthetősége. Az MHP (Multimédia Home Platform) első változatát, az 1.0 verziót 2000. elején fogadta el a DVB Konzorcium. A cikk írásakor e szabvány a negyedik változatánál tart, melyet a DVB Projekt MHP 1.0.3 néven tart nyilván.

Az MHP fő célja, hogy a jelenlegi 'vertikális piac' helyett 'horizontális piac' jöjjön létre. A 'vertikális piac' azt jelenti, hogy mindegyik szolgáltató saját platformot üzemeltet fejállomással, feltételes hozzáférési rendszerrel, átviteli eljárással, vevőkészülékkel (Set Top Box-al, továbbiakban STB). 'Horizontális piac' esetén az előbb felsorolt funkciók mindegyike jól definiált interfészekkel rendelkezik és sokféleképpen megvalósítható. A 'horizontális piac' körülményei között bármely digitális tartalomszolgáltató bármilyen előfizetői terminál – egyszerű és sokfunkciós STB, integrált tv-készülék, multimédiás PC – részére szolgáltatathat. A lehetséges szolgáltatások közé tartozik a fizetős tv (Pay TV), az elektronikus műsorkalauz (EPG), a tőzsdei árfolyamok és az Internet-szolgáltatás. 'Horizontális piac' esetén az MHP kulcsfontosságú eleme az alkalmazási programozási interfész (API), amely különböző szolgáltatók alkalmazásai és a gyártóspecifikus hardver és szoftver implementációk között teremt platformfüggetlen kapcsolatot.

Az MHP kétségkívül egy komplett átviteli rendszer egyik kulcseleme. Ennek a rendszerelemnek harmonikusan kell illeszkednie a műsorszóró átviteli láncba. Ez az eszköz nemcsak az alkalmazásokat tartalmazó adatfolyam (objektumkarusszel) hatékony menedzselésért, hanem az MHP eseményeknek a műsorokhoz való szinkronizálásáért is felelős.

A cikk első szakasza az MHP szabvány azon fontosabb elemeit emeli ki, amelyeket az MHP alkalmazásokat kijátszó karusszelszerverben alkalmazni kell. Ugyancsak bemutatjuk az MHP alkalmazás beágyazását megvalósító protokollstruktúrát, melyen keresztül bemutatható, hogy a DVB milyen eszközöket biztosít a tervező számára az karusszelszerver által létrehozott objektumkarusszelek optimalizálásához.

A második és harmadik szakasz egy műsorszóró átlomásba integrálható MHP karusszelszerver megvalósításának részleteivel foglalkozik. Ismertetünk néhány adatot arra nézve, hogy az objektumkarusszel kialakításával és kijátszásával milyen összefüggésben áll az az idő, ami alatt egy vevő-dekóder befogja (retrieve) a teljes alkalmazást. Fontos kérdés a vevőberendezések kérdése. Bemutatunk néhány problémát a jelenleg használt vevőberendezésekkel kapcsolatban, valamint a készülékek várható fejlesztésének irányvonalát.

### 3. Az MHP platform áttekintése

Az MHP platform központi eleme a DVB-J technológia. A DVB-J platform a Sun Microsystems Inc. által kifejlesztett Java™ platformon alapul, amit a DVB MHP projekt keretében kiegészítettek a televíziós technika szempontjából fontos osztálykönyvtárakkal. A DVB-J felhasználja a Java virtuális gép (JVM) technológiáját, amelynek specifikációja az informatikusok körében már néhány éve ismert. A JVM hardverfüggetlen bájtkódot dolgoz föl. Így az alkalmazások könnyen átvihetők egyik platformról a másikra (portabilitás).

A DVB-J alapvető API-ja a Java osztálykönyvtárakra épül (Java Class Libraries). Ezekbe a következő osztályok tartoznak: java.lang, java.util, java.io, java.net. Emellett szintén a Sun által kifejlesztett JavaTV és a DAVIC (Digital AudioVisual Council) szervezet által kidolgozott HAVi (Home AudioVisual Interface) specifikáció elemei is szerepelnek benne. Az MHP specifikációban a DVB-J mellett a HTML egy módosított változata is jelen van DVB-HTML néven, ami a HTML nyelv szintaktikájának megtartása mellett számos új adattagot (tags) és jellemzőt is specifikál.

A DVB-HTML specifikációja azonban még nem tekinthető véglegesnek, mivel a használat szempontjából számos fontos elemet nem definiál (például script beágyazás, futtatás). Ezért jelenleg az MHP alkalmazások átvitele alatt a fent leírt DVB-J technológián alapuló alkalmazások kód és adatfájljainak átvitelét értjük. A DVB által adatátvitelre használt protokollokat mutatja az 1. ábra.

Az MHP alkalmazások fájljainak átvitelére a DSM-CC (Digital Storage Media Command and Control) objektumkarusszel alkalmazható. Az objektumkarusszel adatstruktúrája az adatátvitelre kifejlesztett adatkarusszel segítségével vihető át. Az adatkarusszel leképzését az MPEG-2 transzport adatfolyamba a DSM-CC szabvány specifikálja. Egy speciális adatstruktúra, a szekció (section) teszi lehetővé, hogy a kódolt audió

Alkalmazás		
DVB-MPE (MultiProtokoll Enkapszuláció)	DSM-CC adat- karusszel	DSM-CC objektum karusszel
		DSM-CC adat karusszel
DSM-CC szekció		
MPEG-2 transzport adatfolyam		

1. ábra  
A DVB műsorszóró csatornán használható  
adatátviteli protokollok

és videó mellett más természetű adatokat is hordozzon az MPEG-2 adatfolyam. Az MHP alkalmazás közvetlenül is hozzáférhet az DSM-CC szekcióihoz, így más protokollok által hordozott adatokat is felhasználhat. Az adatkarusszel és az MPE protokollok jellemzője, hogy az adatokat nyers formában hordozzák valamilyen keretelési algoritmus szerint, de nem foglalják azokat összefüggő egységbe. Ezt az egységbe foglalást biztosítja az objektumkarusszel, ami többek közt fájlrendszert is tud kezelni. Az MPE az IP (Internet Protokoll) által hordozott adatok átvitelére van optimalizálva. Az objektumkarusszel DSI (Download Server Initiate) protokollüzenete adja meg az étvitt fájlrendszer referencia (root) pontját. A DII (Download Info Indication) üzenetek tartalmazzák a fájlrendszer elemeinek azonosításához szükséges információkat valamint a későbbiekben ismertetésre kerülő prefetch és cacheelési paramétereket. Az adatokat DDB (Download Data Block) üzenetekben viszik át. A DDB logikai mérete maximálisan 64kbyte lehet, a DSM-CC szekció logikai mérete 4kbyte, amelyeket 188 byte-os MPEG-2 transzport adatcsomagokban kerülnek átvitelre.

Az objektumkarusszelben a Broadcast Inter-ORB Protocol (BIOP) szerinti üzeneteket lehetséges továbbítani, amelyek közös struktúrája a BIOP általános objektum üzenetre épül. A BIOP üzenetek a következő információkat hordozhatják:

- *CORBA string*
- *BIOP fájl:*  
egy fájl azonosítását szolgáló fejléccet, valamint a fájl bájtoit tartalmazza
- *BIOP alkönyvtár:*  
a fejléc egyéb tartalmán túl tartalmazza az alkönyvtár környezetét, az azonosítását, nevét, és a tartalmát.
- *BIOP ServiceGateway:*  
Az objektumkarusszelben lévő könyvtárstruktúra kiindulási (root) pontját adja meg a Service Gateway. Ennek az üzenetnek a formátuma azonos a BIOP alkönyvtár üzenet tartalmával, eltekintve attól, hogy itt a típus információ "srg" a "dir" helyett.
- *BIOP interoperábilis objektum referencia:*  
az objektumkarusszelben lévő objektumok azonosítására szolgáló üzeneteket tartalmazza, ideértve a saját karusszel mellett a más adatfolyamban lévő karusszeleket is

- *BIOP bitfolyam:*

Az azonos multiplexben lévő bitfolyamok leírására szolgáló üzenetek, amelyekben a TS-ek nevét, azonosítóját, a benne lévő szolgáltatás nevét és azonosítóját, valamint a videó, audió és adat azonosítóját tartalmazza. It viszik át továbbá az időalap (Normal Play Time) értelmezésére vonatkozó információkat. Ennek az üzenetnek az adatfolyam események alkalmazása esetén van jelentősége.

- *BIOP bitfolyam esemény:*

Ez az üzenet egy bitfolyamban az adatfolyam események leírására szolgál, a műsorfolyamokhoz szinkronizált alkalmazások esetén alkalmazhatók.

Az MHP alkalmazásokat alkotó fájlok BIOP fájlüzenetként kerülnek átvitelre.

#### 4. Az MHP objektumkarusszal paraméterek optimalizálása

Az MHP paraméterek optimalizálásának a célja, hogy a felhasznált adatfolyam sávszélessége a lehető legkisebb lehessen amellett, hogy az alkalmazások a vevőkészülékek számára elfogadható gyorsasággal álljanak rendelkezésre. Mivel az előbb említett két paraméter, azaz az alkalmazások átvitelére használt sávszélesség és az alkalmazás adatfolyamból való kinyeréséhez szükséges idő fordított arányban van, ezért meg kell találni a megfelelő egyensúlyt e két paraméter között.

A felhasznált adatfolyamban (objektumkarusszal) az alkalmazások szoftver elemei (fájlok, könyvtárak, adatfolyamok) helyezkednek el. Ezek mellett található meg az alkalmazások jelzésére egy speciális adatstruktúra, az AIT (Application Information Table) és az alkalmazások működésének szinkronizálására szolgáló műsorfolyam-események (stream events).

Az AIT olyan struktúra, amely ciklikusan továbbítva egy vevőkódér számára lehetővé teszi az MHP alkalmazások megtalálását a transzport adatfolyamban. Az AIT-et periodikusan újra küldik, általában ennél kevesebbet, 1-3 másodpercet szokás választani. A jelenlegi karusszelszerverek általában önállóan generálják az AIT-t a megadott paraméterek alapján. A karusszelszerver tipikusan TCP/IP hálózat felett távoli eljárásírással vezérelt (Remote Procedure Call, RPC). Az RPC a vezérlő rendszer számára hozzáférést biztosít a kijátszó berendezés API-jához, amivel a konfiguráció és az adatok feltöltése is lehetséges.

Az objektumkarusszal lehet statikus, azaz változatlan tartalmú és lehet dinamikus, amikor tartalma valamilyen esemény hatására (például egy szavazási alkalmazás esetén ha az aktuális szavazatszámok) megváltoznak.

Az alkalmazások szállításához szükséges sávszélesség és a letöltési idő közötti kapcsolat szemléltetésére egy egyszerű példát adunk meg. Ha az alkalmazások részére 512 kbit/s áll rendelkezésre, és az alkalmazás

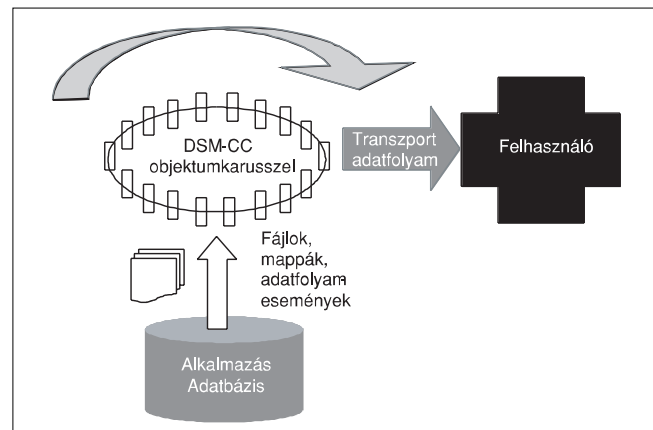
mazás 8 db 100 kb-ot méretű képet és 224 kb-ot Java .class fájlt tartalmaz, akkor a letöltési idő 1024kb-ot/512kbit/s=16 másodperc. Kérdés, hogy a felhasználók mennyi időt hajlandók várni az alkalmazás letöltésére. A tervezőknek ezeket a körülményeket mindenképpen figyelembe kell venniük. Az összefüggésből következik, hogy az objektumkarusszal mérete a letöltési idő nagysága miatt kritikus. Az alkalmazások mérete is minél kisebb kell, hogy legyen a gyors betöltés érdekében.

Néhány fontos tanácsot sorolunk most fel a DVB-J alkalmazások fejlesztéséhez:

- minden lefordított Java osztálynak van egy fejléce, ezért célszerű az osztályok számát minimalizálni
- a Java csomagok elnevezési szabályai miatt bonyolult, többszintű könyvtárstruktúra alakulhat ki, ehelyett törekedni kell az egyszerű könyvtárstruktúrára, és használni kell az AIT-ben a „class path” mezőt alternatív könyvtárak használatához
- a több különböző, de ugyanazon transzport adatfolyamon belüli közösen használt Java osztályokat célszerű egy közös, erre a célra létrehozott objektumkarusszalban átvinni, amit minden alkalmazás használhat
- a feleslegesen nagy felbontású képeket (például true color) kerülni kell, a képernyő adottságainak megfelelő színpaletta használata előnyös („MHP-safe” paletta)

Ahhoz, hogy a felhasználóknak ne kelljen túl sokat várniuk az alkalmazások indulására, optimalizálni kell az objektumkarusszal szerkezetét is. Ki kell választani az alkalmazás indulásához minimálisan szükséges fájlokat, és ezek együttes letöltésének elsőbbséget kell adni. Ezek a fájlok gyakrabban kerülhetnek átvitelre, mint az induláshoz nem feltétlenül szükséges fájlok (például a képeket tartalmazó fájlok a programot tartalmazó fájloknál fele vagy harmada gyakorisággal is kerülhetnek átvitelre.) Harmadik lehetőség a ciklikus átvittel csökkenteni a nagyobb méretű fájlok átvitelére használt sávszélességet így az alkalmazás indításához használható sávszélesség nagyobb lehet. Ennek az átvitelnek a lényege, hogy hasonlóan a teletext rendszerekhez, az objektumkarusszal első körbefordulásakor az első nagyobb méretű fájl kerül átvitelre a kö-

2. ábra Az adatátviteli rendszer felépítése



vetkező körbeforduláskor a következő. Ily módon, bár a nagyobb fájlok (tipikusan képet tartalmazó fájlok) betöltésére akár több köridőt is kell várni (maximálisan annyit, ahány ilyen nagy objektumot viszünk át külön körökben), de az alkalmazás elindulása meggyorsítható.

Csökkentheti az elérési időt az induláshoz szükséges komponensek az STB-be való előzetes letöltése (pre-loading). Ez a paraméter az egyes adatmodulokra megadható, de a vevőkészülék nem köteles ezeket az információkat figyelembe venni.

Az objektumkarusszal felépítése során a fájlok csoportosítása is fontos, mivel az adatfolyam 64kbájtos adatcsomagokat visz át. Ezekben a csomagokban helyezkedhetnek el fájlok, vagy ha egy fájl ennél a méretnél nagyobb, akkor darabolva több adatcsomagban kerül átvitelre. Mivel az adatcsomagok dekódolásához szükséges erőforrások nagysága egyenes arányban áll a dekódolandó adatcsomagok számával, ezért célszerű egy-egy adatcsomagba a tartalmilag összetartozó (például az egy időben együtt megjelenő tartalmat hordozó) fájlokat rendezni. Ezzel dekódolási idő takarítható meg. Az alkalmazás indulása szempontjából fontos fájlokat tartalmazó modulokat célszerű a többinél gyakrabban átvinni, így az elérési idő tovább csökkenthető.

Az objektumkarusszal tartalmának frissítésekor is célszerű a frissülő komponenseket önálló adatcsomag(ok)-ba rendezni, így a frissítés egyszerűen ezen adatcsomag(ok) cseréjéből állhat. A fájlok modulokba rendezését, a modulok kijátszási ütemezését és frissítési eljárását általában a vezérlőrendszer adja meg a karusszal-szervernek egy XML (Extensible Markup Language) fájl formájában. A STB az objektumkarusszal lokálisan tárolhatja (cache-elheti) a gyorsabb elérés és feldolgozás érdekében. A memóriában lévő fájlokat a STB frissíti az objektumkarusszalben lévőkkel megadott időközönként. A frissítési gyakoriság háromféle lehet:

- transzparens cache-elés:
  - 0,5 másodpercenként frissül a fájl
- fél-transzparens cache-elés:
  - 30 másodpercenként frissül a fájl
- statikus cache-elés: a fájl nem frissül, csak egyszer, az alkalmazás indulásakor kerül betöltésre.

A frissítési időköz fájlonként nem, csak modulonként adható meg, a modul verziószám segítségével. Ha egy modul új verziószámot kap, akkor az STB frissítéskor a modul tartalmát betölti a cache-be. A gyakran frissülő fájlokat külön modulba szervezve és a modul frissítését transzparensre állítva az éppen futó alkalmazás mindig a legfrissebb adatokhoz juthat, megtakarítva a közvetlen adatfolyamból való olvasás-késleltetési időt. A frissítési idő mellett a fájloknak prioritás is adható.

A jelenlegi STB-k esetén a számított elméleti elérési időnél lényegesen (minimálisan a négyszeresét) több időt fordítanak a dekódolásra. A cache funkciók megvalósítása sem teljes az STB-kben, ezért az újraolvasás az adott fájl adatfolyamból jelent csak biztos megoldást. Ez a STB-kben rendelkezésre álló erőforrások (demultiplexer chip feldolgozási sebessége, CPU számítá-

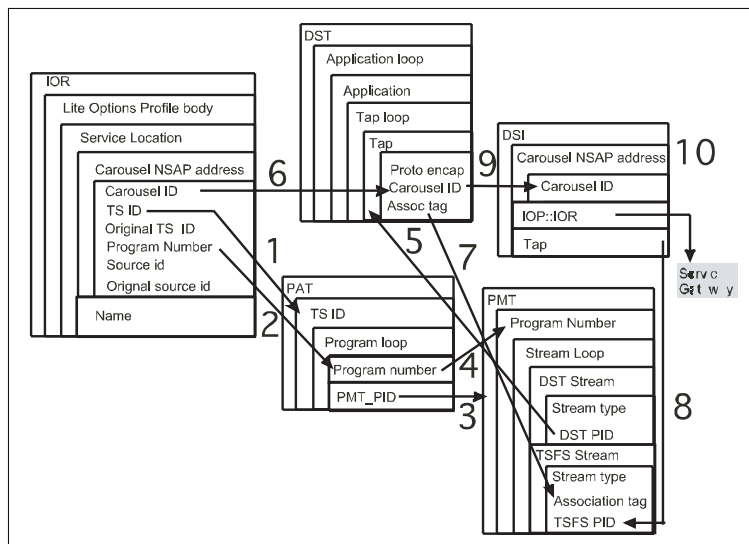
si kapacitása) szegényes voltának tudható be, ami még fontosabbá teszi az objektumkarusszal optimalizálását. A fájlrendszer regenerálás komplexitásának bemutatására nézzük a fájlrendszer referencia pontjának (root) kinyeréséhez szükséges műveleteket (3. ábra):

- 1) A vevőberendezés a tárolt csatornatáblázat alapján, a NSAP cím alapján meghatározza a TS azonosítóját (TSID), a TS-re hangol és PAT táblát kiolvassa.
- 2) A PAT (Program Association Table) táblából meghatározza a program sorszámnak megfelelő bejegyzést.
- 3) A PMT (Program Map Table) PID alapján lekérdezi a PMT szekciót.
- 4) A PMT szekció lekérdezéskor ellenőrzi, hogy az ott szereplő program sorszám azonos-e a lekérdezett program sorszámmal.
- 5) A PMT-ben lévő adatfolyamból meghatározza a DST információ helyét.
- 6) A DST-ben meghatározza a carousel ID-nek megfelelő TAP-et.
- 7) A TAP association tag adatmezőjének megfelelő azonosítójú TSFS adatfolyam kinyeréséhez szükséges információt kiolvassa a PMT-ből.
- 8) A PMT-ből kiolvasott adatfolyam azonosító érték határozza meg azt a program komponenst, ami tartalmazza azt a DSI üzenetet, amiben a Service Gateway információ megtalálható.
- 9) Kiolvassa a DSI üzenetet.
- 10) Az IOP::IOR üzenet dekódolásával meghatározza a ServiceGateway információt.

Látható hogy az információ kinyeréséhez öt adatstruktúra elemzése szükséges, amelyek kinyerése és létrehozása is szükséges a fent felsorolt műveletek elvégzése előtt.

A 6-8MB közötti érték komoly korlát egy szupertext alkalmazás számára, mivel a felhasználó a hagyományos teletext rendszerekben közel 800 oldalnyi információt olvashat. A szupertext rendszerekben ha feltételezzük, hogy az oldalakon a hagyományos oldalaknak megfelelő mennyiségű szöveges információ mellett egy hozzávetőlegesen a TV képernyő területének negyedét kitevő grafikus információ (kép) is elhelyezkedik akkor ez 16MB méretű fájlrendszert eredményez, ami meghaladja az korlátként említett 6-8MB-ot.

Megoldást jelenthet a problémára az információ-mennyiség csökkentése vagy külön objektumkarusszalokba történő szervezése. Például a témák szerint elkülönülő tartalom kerülhet különböző objektumkarusszalokba, így bár egy objektumkarusszal mérete korlátozott, a felhasználó mégis a hagyománnyal megegyező oldalszámú szupertext oldalhoz férhet hozzá. Ekkor a néző, ha azonos témán belüli oldalakat nézeget, akkor gyorsan férhet hozzá a következő oldalhoz, hiszen a STB tárolta az aktuális objektumkarusszal. Ha azonban vált a témák közt, akkor ki kell várnia az objektumkarusszal teljes visszafejtéséhez és a tároláshoz szükséges időt. Természetesen lehetséges az adatok más szempontok szerinti csoportosítása is,



3. ábra  
A ServiceGateway info kinyeréséhez szükséges műveletek

például a népszerű sokak által olvasott oldalak, vagy a kevesek számára érdekes speciális információkat tartalmazó oldalak kiválasztása.

Fontos megemlíteni, hogy a gyakorlatban a bitsebesség növelésével a letöltési idő csak egy, az alkalmazott STB-től függő meghatározott értékig csökkenthető. Ezen bitsebesség érték fölé nincs értelme növelni az átvitelre használt sáv szélességet, mivel az alkalmazás betöltése nem lesz gyorsabb. Ez abból adódik, hogy várakozási időben ekkor már a karusszal dekódolásához kellő idő dominál, és az adatkinyerés gyorsításának hatása itt már nem érzékelhető. Általános szabályként elmondható, hogy 1 MB alatti objektumkarusszal méreteknél 512kbit/s fölé nem érdemes menni, és általában 1-2Mbit/s a maximális érték, amit érdemes egy karusszal kijátszására fordítani.

A objektumkarusszal blokkjainak tartalma Zlib (RFC 1951) eljárással tömöríthető, ami elvben csökkentené az objektumkarusszal méretét. Ezt azonban általában nem használják, mivel az adatmennyiség jelentős részét adó (jpeg és gif formátumú) grafikus információk alig tömöríthetőek (rendszerint alig néhány százalék nyereség érhető el) ezért ez csupán a dekódolási időt növelné. Látható hogy az átvitel optimalizálásához a tartalom előállítás, alkalmazásfejlesztés és a kijátszórendszer közti együttműködés nélkülözhetetlen. Ez fontos szempont lehet a jövő alkalmazás -és tartalomfejlesztési illetve hálózatüzemeltetési együttműködések kialakításakor.

### 5. A vevőberendezések problémái, fejlesztésük irányvonala

A jelenleg az MHP-s vevőkészülékek a minimálisan előírt erőforrásokat és funkciókat tartalmazzák, ami azal magyarázható, hogy a gyártók igyekeznek a fejlesztési időszakban a berendezések árát a minimálisra szorítani. Léteznek más operációs rendszerrel (például

MediaHighway) rendelkező STB-k, amelyek operációs rendszerét utólag az MHP alkalmazások futtatására is alkalmassá tették. Ezek az STB-k tartalmazhatnak a minimálisnál jelentősen több erőforrást, és a velük elérhető szolgáltatások köre is szélesebb lehet. Ezen termékek MHP kompatibilitása nem teljes, többnyire csak az adott gyártó fejlesztőeszközeivel készített alkalmazások futnak rajta.

A jelenleg elért szolgáltatások grafikai és számítási igénye igen magas a jelenleg a STB-kben alkalmazott erőforrásokhoz képest. Emiatt a jövőben a STB-kben az erőforrások bővülése várható, ami a számítástechnika jelenlegi fejlődési ütemét tekintve több nagyságrendű is lehet. A PC világával ellentétben, ahol az erőforrások teljesítményének kétszereződése hozzávetőlegesen évente teljesül a STB-k világá-

ban ez a folyamat sokkal lassabb. Az elmúlt három év alatt a számítási kapacitás háromszorozódásának a tárolókapacitás kétszereződésének lehettünk tanúi.

Az ismertett MHP rendszer interpreteres voltából adódó hordozhatóság előnye mellett rontja a helyzetet, hogy a JAVA jellegű rendszer teljesítménye messze elmarad a konkurens megoldások bináris, közvetlenül futtatható C alapú kódjától. A problémát enyhítheti a JIT (Just In Time) módszerek alkalmazása, ami gyorsíthatja az MHP alkalmazások futtatását, de az erőforrások nagymértékű növelése így sem kerülhető meg.

A cikk írásakor a gyorsabb STB-kben 180MHz körüli processzor, 32MB RAM és 8MB Flash az általános, ilyen konfiguráció ára az olaszországi piacon 200-250 Euro körül mozog, de kapható nagyobb kapacitású box is 350MHz-es processzorral 80MB RAM-al 16MB Flash-el. A cikkben ismertett műszaki lehetőségek és megoldások magas erőforrásigénye miatt ezek hasznosítása csak a jelenleginél több erőforrással rendelkező STB-kkel valósítható meg maradéktalanul. A jövőben remélhetően az erőforrások az említettnél gyorsabb ütemű bővülése és az árak esése hamar bekövetkezik, ami lehetővé tenné az MHP széles körű elterjedését és az interaktív alkalmazásokon keresztül a modern televíziózás térhódítását.

### Irodalom

- [1] MPEG-2 Systems (ETSI/IEC 13818-1)
- [2] DSM-CC (ETSI/IEC 13818-6)
- [3] DVB-SI (ETSI/IEC 300 468)
- [4] Multimedia Home Platform 1.0.3, ETSI ES 201801 V1.1.1, July 2003.
- [5] Sun Microsystems: Java Platform Specification, 2001.
- [6] A. Arthurs, A. Collins: Delivery of Interactive Data Services by Multiple Networked Content Providers, IBC 2002.
- [7] B. Pichot ext al.: Efficient Broadcast of MHP Applications, IBC 2001.
- [8] Bartley Calder: DVB-J Platform Overview. Sun Microsystems, IBC 2000