

# **híradástechnika**

VOLUME LIX.

# **2004/10**

*Október*



**Minőség és biztonság**

**Méretezési módszerek**

**Műsorszórás**

**A Hírközlési és Informatikai Tudományos Egyesület folyóirata**

# Tartalom

KOCKÁZAT VAGY BIZTONSÁG 1

## MINŐSÉG ÉS BIZTONSÁG

**Győri Sándor**

Jelzésekódolás többszörös hozzáférésű VAGY csatornán 2

**Lengyel László, Levendovszky Tihamér**

Aspektus-orientált programozás 8

**Kovács László, Vass Dorottya, Vidács Attila**

Szolgáltatásminőségi paraméterek előrejelzésének javítása outlierok detektálásával és eltávolításával 13

**Dénes Tamás**

Latin és bűvös négyzetek a játékos alkalmazásoktól a biztonsáig 19

## MÉRETEZÉSI MÓDSZEREK

**Szödényi Ákos**

Optikai szűrők hatása optikailag átlátszó WDM hálózatok méretére 25

**Kovács Dániel László**

Intelligens rendszerek egységes tervezése 29

## MŰSORSZÓRÁS

**Krémer Szabolcs**

Interaktív televíziós alkalmazások átvitelének vizsgálata az Antenna Hungária DVB-T platformján 39

**Szegedi Péter**

VLAN címke alapú cross-connect funkció a videóhálózatokon 44

**Nagy Beatrix Havaska**

Beszélgetés Bendzsel Miklóssal, a Magyar Szabadalmi Hivatal elnökével 49

Interjú Hetényi Péterrel, a Siemens Rt. Felügyelő Bizottságának elnökhelyettesével 53

Címlap: A gyors változások nem csak az informatikában, hanem a művészetben is nyomon követhetők (Szabotka Imre: Almaszüret, 1930)

### Főszerkesztő

ZOMBORY LÁSZLÓ

### Szerkesztőbizottság

Elnök: LAJTHA GYÖRGY

BARTOLITS ISTVÁN  
DIBUZ SAROLTA  
GÖDÖR ÉVA

GYŐRI ERZSÉBET  
HUSZTY GÁBOR  
JAMBRIK MIHÁLY

KÁNTOR CSABA  
MARADI ISTVÁN  
PAKSY GÉZA

PAP LÁSZLÓ  
SALLAI GYULA  
TORMÁSI GYÖRGY

# Kockázat vagy biztonság

*lajtha.gyorgy@ln.mata.v.hu*

**A** mentőautó vezetője nap mint nap nagy kockázatot vállal. Piroson áthajt a kereszteződésben úgy, hogy a sarkon álló gyalogosoknak égne a haja. A vezető számítja arra, hogy a kereszt irányban, zöld jelzésen érkező gépkocsik vezetői nem halálkárosultak, nem szól olyan hangosan a zene, hogy minden külső hangot elnyomna és reméli, hogy időben meg tudnak állni. A vezető minden alkalommal vállalja a kockázatot, mert ezzel esetleg emberéleteket tud megmenteni, bízik a saját reflexeiben és gyors döntéseiben. Számos más területen is kell kockázatot vállalnunk, nem mindig ilyen nagyot és nem minden nap.

A fejlesztésnek is van kockázata. Itt azonban nem közvetlen emberéletekről, hanem pénzről, vagy munkahelyekről van szó. Ragyogó ötletek buktak meg és okoztak milliárd dolláros veszteségeket. Ha az elmúlt húsz évet tekintjük csak át akkor látjuk, hogy a videotextbe fektetett fejlesztési pénzek mind elvesztek. Ennek a kutatásnak éllovasa volt a Philips és csak a vezérigazgató gyors döntésének köszönhető, hogy a cég nem ment tönkre. Ugyanis kijelentette, hogy ha leírjuk a fejlesztési költségeket és nem kezdjük meg a gyártást, akkor pontosan tudom mennyi a cég vesztesége. De ha elkezdjük propagálni a terméket és felszerszámozzuk, akkor arra az egész vállalat rámehet. Hasonló nagy bukás volt a földi mobilállomásokról elérhető szatelithálózat az Irídium.

Voltak olyan fejlesztések is, melyek a maguk idejében sikeresek voltak, azonban az idő túlhaladta azokat. Nem volt több mint tíz év az ISDN virágzásának kora. Hasonlóan rövid ideig élt a DAB elnevezésű digitális rádiórendszer, melyet a DRB váltott fel. Jellemző történet a személyhívó esete is, mely az első mobil információközlési lehetőség volt és megszületése után Szingapúrban, Hong-kongban sorban álltak az emberek a készülékért és az ezzel együtt járó előfizetésért. Magyarországon is két vállalkozás indult személyhívóra, de egyik sem volt sikeres, mert hamar áttértek az emberek a mobilra.

Mindez nem azt jelenti, hogy nem szabad a fejlesztés kockázatát vállalni, mert ha nem vállaljuk az újdonságok bevezetését, akkor a konkurencia az összes vevőt, felhasználót elhódítja és termékeink, szolgáltatásaink lassanként múzeumba kerülnek. Gyorsan változó világunkban tehát az lenne a cél, hogy minimalizáljuk a kockázatot. A kutatási és fejlesztési eredményeket gondosan megvizsgálva és az ismert kockázatelemzési eljárásoknak alávetve lehet a siker valószínűségét számszerűsíteni és megjósolni. Érdemes megfontolni a Bell Laboratórium fejlesztési stratégiáját, ahol egy-egy ötlet kidolgozására több csoportot hoztak létre, akik egymástól függetlenül dolgoztak. Különböző megoldásokat tártak a vezetés elé, akiknek módjuk volt bevonni üzletembereket, felhasználókat annak érdekében, hogy együttesen megvizsgálják melyik megoldás a legperspektivikusabb és mikor legkisebb a kockázat.

Jelen számunkban több forgalmi problémára keresünk a megoldásokat. Szinte egyértelmű, hogy ezek alkalmazásával inkább remélhetjük, hogy a távközlési szolgáltatások sikeresek lesznek. Ebbe a sorozatba tartozik az optikai szűrők hatásának vizsgálata és a jelzésekódolással kapcsolatos gondolatok. Az intelligens rendszerek tervezése pedig általánosságban igyekszik kialakítani a legjobb tervezési módszereket, figyelembe véve számos lehetőséget. Két cikkünk a műsorszórással foglalkozik, ahol a minőség javításnak jelentős perspektívái vannak.

A sort a Siemens egyik vezetőjével készült interjú zárja, ahol megláthatjuk, hogy a folytonosság és a fejlesztés hogyan erősítheti egymást. Az újdonságok stabil háttérre építve megvédik a céget a katasztrófától. A kockázattal tehát szükséges, kockázat nélkül nincs siker, de kell valamilyen biztonságos háttér, ami a szakmai tudásban, tapasztalatban és a nyugodt gazdálkodásban realizálódhat.

*Dr. Lajtha György*

# Jelzésekódolás többszörös hozzáférésű VAGY csatornán

GYÓRI SÁNDOR

BME, Számítástudományi és Információelméleti Tanszék  
gyori@szit.bme.hu

Reviewed

**Kulcsszavak:** többszörös hozzáférés, kombinatorika, kódkonstrukciók, becslési módszerek

A jelzésekódolást többszörös hozzáférésű VAGY csatornán vizsgáljuk, amikor az összes  $T$  közül legfeljebb  $M$  felhasználó aktív. Bemutatunk gyakorlati alkalmazási példákat. A matematikai probléma megegyezik az extrémális kombinatorika  $M$ -fedésmentes halmazrendszerek témakörével. Alsó és felső korlátokat adunk az  $n(T, M)$  minimális kódszóhosszra, és néhány kódkonstrukciót is ismertetünk. A jelenleg ismert legjobb korlátok  $1 \ll M \ll T$  esetén:

$$\frac{1}{2} \frac{M^2}{\log M} \log T \leq n(T, M) \leq \frac{1}{\ln 2} M^2 \log T,$$

tehát meglehetősen távol van egymástól az alsó és felső korlát. Ismertetünk néhány hatékony kódkonstrukciót.

## 1. Csatornamodell

A  $T$ -felhasználós többszörös hozzáférésű VAGY csatorna – melyet először Cohen, Heller és Viterbi [6] vezetett be – bináris bemenetekkel ( $x_i$ ,  $1 \leq i \leq T$ ) és bináris kimenettel rendelkező determinisztikus csatorna, melynek kimenetén pontosan akkor jelenik meg 0, ha az összes bemenete 0. A csatorna kimenete tehát a bemenetek Boole-algebra szerinti összegeként adódik (1. ábra):

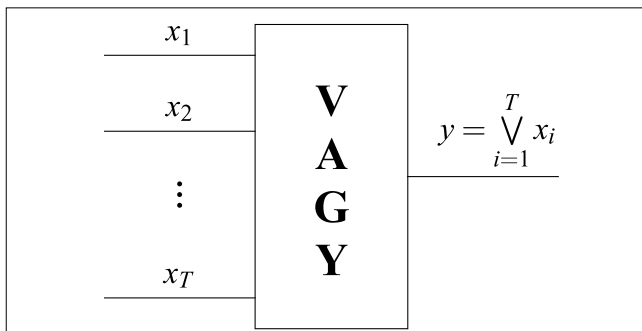
$$y = \bigvee_{i=1}^T x_i = x_1 \vee x_2 \vee \dots \vee x_T.$$

VAGY csatornaként modellezhető például egy olyan kommunikációs csatorna, amelyen a ki/bekapcsolás (OOK, On/Off Keying) modulációs eljárást alkalmazunk, vagyis az 1-es bitnek megfeleltetünk egy jelalakot, míg a 0-s bitet a konstans 0 jelszint jelenti. Ekkor a demoduláció az a döntés, hogy az összes aktív felhasználó a 0 jelszintet adta-e a csatornába.

Egy többszörös hozzáférést lehetővé tevő kódnak általában három feladatot kell megoldania [16]:

- az aktív felhasználók azonosítása (detection, identification),
- kódszavaik szinkronizálása (synchronization),
- az üzenetek hibavédelme (decoding).

1. ábra Többszörös hozzáférésű VAGY csatorna



Folyamatosan aktív felhasználók esetén a maximális, egységnyi kihasználtság a legegyszerűbb módon, az időosztás segítségével is elérhető. Részleges felhasználói aktivitás esetén a hatékony kommunikáció megvalósítása nehéz és még korántsem megoldott probléma.

A továbbiakban a jelzésekódolás (signature coding) esetét fogjuk vizsgálni. Minden felhasználónak van egy saját  $n$ -bites kódszava, amelyet jelzésre használhat, viszont a csupa 0 kódszót (a konstans 0 jelszintet) használja, ha éppen nem aktív.

A kódolási probléma a következő: keressünk olyan kódot, amely garantálja, hogy ha a  $T$  felhasználó közül egyidejűleg legfeljebb  $M$  felhasználó akar jelezni (aktív), akkor a VAGY csatorna kimeneti vektorából egyértelműen meghatározható az aktív felhasználók halmaza. Az erőforrás hatékony kihasználása érdekében határozzuk meg azt a minimális kódszóhosszat, amely mellett a probléma még megoldható.

A kódosztásos többszörös hozzáférésű (CDMA) rendszerekben a felhasználók általában csak egy-egy kommunikációs ciklus idejére kapnak kódot (session code), tehát itt nincs azonosítási feladat. Esetünkben viszont minden felhasználó előre megkapja az alkalmazandó kódot, amely a rendszer teljes élettartama alatt állandó.

### 1.1. UD és ZFD kódok

Az UD és ZFD kódokat a VAGY csatornán való kommunikációhoz Kautz és Singleton [14] vezették be.

#### 1. definíció (UD kód)

$T$  darab  $n$  hosszú kódszóból álló kód  $M$ -edrendben egyértelműen dekódolható (Uniquely Decipherable,  $UD(T, M, n)$ ), ha a legfeljebb  $M$  kódszóból álló összegek mind különbözőek.

Formálisan ez azt jelenti, hogy egy  $C = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$  kód  $UD(T, M, n)$  tulajdonságú, ha bármely  $A, B \subset \{1, 2, \dots, T\}$  halmazra, melyekre igaz, hogy  $|A| \leq M, |B| \leq M, A \neq B$ , teljesül az alábbi

$$\bigvee_{i \in A} \mathbf{x}_i \neq \bigvee_{j \in B} \mathbf{x}_j.$$

(Bináris vektorok összegén a koordinátánkénti Boole-összegek vektorát értjük.)

**Példa: dokumentumkereső rendszer.** Egy könyvtári nyilvántartásban a dokumentumokról (könyvek, folyóiratok stb.) különböző tulajdonságokat tárolnak. A keresés ezek alapján történik, vagyis el kell döntenünk, hogy egy adott dokumentum rendelkezik-e egy tulajdonsággal.

A vizsgálat megkönnyítésére a dokumentumok fejrészében hatékonyan szeretnénk tárolni jellemző tulajdonságaikat. Amennyiben a lehetséges tulajdonságok száma  $T$  és egy adott dokumentum ezekből legfeljebb  $M$ -mel rendelkezhet, a szükséges információ egy bináris vektorba kódolható, melynek hossza legalább

$$n = \log \sum_{m=0}^M \binom{T}{m} \simeq M \log T,$$

hiszen bármely, legfeljebb  $M$  tulajdonság együttesét meg kell tudnunk különböztetni.

Amennyiben a dokumentumok és tulajdonságaik bővíthetnek, akkor az ezeket leíró bináris vektort célszerű UD kódként előállítani, mivel egy újabb tulajdonság esetén csak az annak megfelelő kódszó VAGY kapcsolatát kell vennünk a régi Boole-összeggel. Legyen  $\mathbf{x}_i$  az  $i$ -edik tulajdonsághoz rendelt bináris vektor. Ekkor az  $i_1, \dots, i_m (m \leq M)$  tulajdonságokkal rendelkező dokumentum leíró vektora az alábbi Boole-összegként adódik:

$$\mathbf{y} = \bigvee_{j \in \{i_1, \dots, i_m\}} \mathbf{x}_j.$$

Így  $UD(T, M, n)$  kód alkalmazásával a leíró vektorból meg tudjuk határozni, hogy a dokumentum mely tulajdonságokkal rendelkezik [14 és 5].

Az UD tulajdonság egyrészt nehezen ellenőrizhető, másrészt a dekódolás során kimerítő keresést igényel a legfeljebb  $M$  kódszóból álló összegvektorok halmazában. Ezek kiküszöbölésére bevezetünk egy könnyebben kezelhető tulajdonságot.

Azt mondjuk, hogy két,  $n$  hosszúságú bináris vektor közül  $\mathbf{z} = (z_1, \dots, z_n)$  fedi  $\mathbf{y} = (y_1, \dots, y_n)$  vektort, ha

$$\mathbf{z} \geq \mathbf{y},$$

vagyis  $z_i \geq y_i (i = 1, 2, \dots, n)$ .

Az előző példában a leíró vektor kódolása egyszerű és gyors volt. Szeretnénk egy hasonlóan egyszerű dekódoló eljárást is. Vegyük észre, hogy tetszőleges  $j \in \{i_1, i_2, \dots, i_m\}$ -re

$$\mathbf{y} \geq \mathbf{x}_j.$$

Ez az UD kódok egy speciális esetéhez vezet.

## 2. definíció (ZFD kód)

$T$  darab  $n$  hosszú kódszóból álló kód  $M$ -edrendben ZFD tulajdonságú (Zero False Drop,  $ZFD(T, M, n)$ ), ha a

legfeljebb  $M$  kódszóból álló összegek kizárólag az őket alkotó vektorokat fedik.

Formálisan ez azt jelenti, hogy ha valamely  $k$ -ra

$$\mathbf{y} \geq \mathbf{x}_k.$$

akkor

$$\mathbf{x}_k = \mathbf{x}_{i_j}$$

valamely  $i_j$ -re. A dokumentumkereső rendszer példájában tehát ZFD tulajdonság esetén egy  $\mathbf{y}$  leíró vektorral rendelkező dokumentum pontosan akkor rendelkezik a  $k$  tulajdonsággal, ha

$$\mathbf{y} \geq \mathbf{x}_k.$$

Ez valóban gyors dekódolást tesz lehetővé, hiszen a tulajdonságok számának függvényében lineáris időben végrehajtható, míg csupán az UD tulajdonság alapján történő dekódolás során az összes lehetséges kombinációt meg kellene vizsgálnunk.

A ZFD tulajdonságot egy dekódolási szabállyal definiáltuk, ezért a  $ZFD(T, M, n)$  kód egyben  $UD(T, M, n)$  is. Mennyit veszünk azzal, hogy ha UD kód helyett ZFD-t használunk?

## 1. tétel (Kautz és Singleton [14])

Egy  $UD(T, M, n)$  kód  $ZFD(T, M-1, n)$  tulajdonságú, és egy  $ZFD(T, M, n)$  kód  $UD(T, M, n)$  tulajdonságú.

### Bizonyítás

1. Legyen a  $C$  kód  $UD(T, M, n)$ . Ha  $C$  nem lenne ZFD  $(T, M-1, n)$  tulajdonságú, akkor lennének olyan  $\mathbf{x}_M \notin \{\mathbf{x}_1, \dots, \mathbf{x}_{M-1}\}$  kódszavai, hogy

$$\mathbf{x}_M \leq \mathbf{x}_1 \vee \mathbf{x}_2 \vee \dots \vee \mathbf{x}_{M-1},$$

vagyis

$$\mathbf{x}_1 \vee \mathbf{x}_2 \vee \dots \vee \mathbf{x}_{M-1} \vee \mathbf{x}_M = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \dots \vee \mathbf{x}_{M-1},$$

ami ellentmond annak, hogy  $C$   $UD(T, M, n)$ .

2. Tegyük fel, hogy a  $C$  kód  $ZFD(T, M, n)$ , de nem  $UD(T, M, n)$ .

Ekkor léteznek olyan  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\} \neq \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L\}$ ,  $K, L \leq M$  kódszóhalmazok, amelyekre

$$\mathbf{x}_1 \vee \mathbf{x}_2 \vee \dots \vee \mathbf{x}_K = \mathbf{y}_1 \vee \mathbf{y}_2 \vee \dots \vee \mathbf{y}_L.$$

Mivel a két halmaz nem egyenlő, létezik egy olyan kódszó, amely csak az egyikben van benne. Legyen ez  $\mathbf{x}_j$ , amely tehát nincs benne az  $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L\}$  halmazban, ugyanakkor

$$\mathbf{x}_j \leq \mathbf{y}_1 \vee \mathbf{y}_2 \vee \dots \vee \mathbf{y}_L,$$

ami ellentmond a feltételezésünknek. ■

Jelölje  $n_{ZFD}(T, M)$  és  $n_{UD}(T, M)$  a  $ZFD(T, M)$  illetve az  $UD(T, M)$  kód minimális hosszát.

## 1. következmény

A ZFD és UD tulajdonságú kódosztályok között az alábbi tartalmazási kapcsolat áll fenn:

$$ZFD(T, M, n) \subseteq UD(T, M, n) \subseteq ZFD(T, M-1, n) \subseteq \dots,$$

a minimális kódszóhosszakra pedig:

$$n_{ZFD}(T, M) \geq n_{UD}(T, M) \geq n_{ZFD}(T, M-1) \geq \dots$$

**Példa: jelzésekódolás.** Tekintsünk egy  $T$ -felhasználós többszörös hozzáférésű VAGY csatornát, ahol minden felhasználó rendelkezik egy  $n$  hosszúságú bináris vektorral (kódszóval). Ha egy felhasználó aktív, elküldi a saját kódszavát, ellenkező esetben a csupa 0 kódszót adja a csatorna bemenetére.

A VAGY csatorna kimenetéből, vagyis az aktív felhasználók kódszavainak Boole-összegéből meg kell tudnunk állapítani, hogy mely felhasználók voltak aktívak. Amennyiben egyidejűleg legfeljebb  $M$  felhasználó lehet csak aktív, úgy egy  $UD(T, M, n)$  kóddal megoldható a probléma, illetve az egyszerű dekódolhatóság érdekében használhatunk  $ZFD(T, M, n)$  kódot is.

**Példa: riasztás.** Fűzzünk fel  $T$  darab tűzjelző készüléket egy közös vezetékre. Amikor egy készülék jelezni kíván, OOK modulációval elküldi a kódszavát a vezetéken. Ha az egyidejű jelzések száma nem haladhatja meg az  $M$ -et, akkor az aktív állomások azonosíthatók a vezetéken megjelenő jelből.

**Példa: bejelentkezés.** Tekintsünk egy mobil távközlési rendszert, amelyben sok, viszonylag ritkán aktív felhasználó akar korlátozott számú csatornán keresztül kapcsolatot teremteni. Az adáshoz a felhasználóknak be kell jelentkezniük a rendszerbe. Ehhez elküldik az egyedi kódszavukat a bázisállomásnak egy VAGY csatornán keresztül, amelynek kimenetéből a bázisállomás megállapítja, hogy mely felhasználók aktívak, és mindegyikhez dedikált csatornát rendel, amelyen már ütközés nélkül kommunikálhatnak.

Napjainkban a mobil távközlőhálózatok a bejelentkezéshez véletlen hozzáférést használnak visszacsatolással. Ez a módszer kiváltható a VAGY csatornán alkalmazott jelzésekódolással, s így megtakarítható a visszacsatolások feldolgozása.

**Példa: mérésadatgyűjtés.** A fogyasztók elektromosenergia-felhasználását szeretnénk automatikusan összegyűjteni. Az elektromos hálózat használható többszörös hozzáférésű VAGY csatornaként. Minden fogyasztásmérő órához egyedi kódszót rendelünk, amelyet az mindig elküld a közös csatornán keresztül egy-egy egység, például 1 kWh elfogyasztása után.

**Példa: csomagküldés ütközéses csatornán.** A részelt ALOHA modellhez hasonlóan tekintsük a többszörös hozzáférésű ütközéses csatornát azzal a lényegi különbséggel, hogy nincs visszacsatolás, azaz az adó nem értesül arról, hogy a csomagja ütközött-e vagy sem.

Mindegyik adó rendelkezik egy egyedi kódszóval (protokoll sorozattal), az  $i$ -edik az  $\mathbf{x}_i$ -vel, és amennyiben az  $i$ -edik adónak van elküldendő csomagja, úgy azt az  $\mathbf{x}_i$  1-eseinek helyén ismételve elküldi, remélve azt, hogy a csomag legalább egyszer sikeresen átmegy.  $T$  felhasználó esetén, amelyek közül egyidejűleg legfeljebb  $M$  aktív,  $ZFD(T, M-1, n)$  kód alkalmazásával ez teljesíthető, vagyis minden  $\mathbf{x}_i$  protokoll sorozatnak lesz olyan 1-ese, amelyet a másik  $M-1$  protokoll sorozat nem fed le.

## 1.2. Kapcsolat a halmazrendszerekkel

A ZFD problémát többen és sokszor vizsgálták extrémális halmazrendszerek kapcsán. Legyen  $U$  egy  $n$ -elemű alaphalmaz.  $\binom{U}{k}$  módon jelöljük az  $U$  halmaz  $k$ -elemű részalmazainak halmazát ( $0 \leq k \leq n$ ), míg  $2^U$  pedig  $U$  hatványhalmazát jelenti:

$$(2^U = \bigcup_{k=0}^n \binom{U}{k})$$

Az  $U$  halmaz részalmazainak egy  $\mathcal{F}$  rendszere a hatványhalmaz egy részalmaz (  $\mathcal{F} \subseteq 2^U$  ).

### 3. definíció

(Fedésmentes halmazrendszer, [10 és 11])

Az  $\mathcal{F}$  halmazrendszer  $M$ -fedésmentes, ha

$$F_0 \not\subseteq F_1 \cup \dots \cup F_M$$

fennáll minden különböző  $F_0, F_1, \dots, F_M \in \mathcal{F}$  halmazra.

Keressük az  $\mathcal{F} \subseteq 2^U$   $M$ -fedésmentes halmazrendszer maximális  $T$  méretét, ha  $|U| = n$ . Ez a probléma megegyezik a ZFD kódok minimális hosszának meghatározásával.

A paraméterek megfeleltetése a következő. Legyen  $U = \{1, 2, \dots, n\}$ , és egy  $F \in \mathcal{F}$  halmazhoz rendeljük az  $\mathbf{x}_F$  bináris kódszót, amelynek  $i$ -edik pozícióján pontosan akkor áll 1-es, ha  $i \in F$ .

A halmazrendszer  $T$  számossága játssza a lehetséges felhasználók számának szerepét, az  $M$ -fedésmentes tulajdonság felel meg az  $M$ -edrendű ZFD tulajdonságnak és az  $U$  alaphalmaz  $n$  mérete pedig a kódszóhossznak.

## 2. ZFD kódok

### minimális hosszúra vonatkozó korlátok

Ebben a szakaszban korlátokat adunk a  $ZFD(T, M)$  kód minimális hosszára.

A legegyszerűbb alsó korlátot annak felismerésével kapjuk, hogy a legfeljebb  $M$  kódszóból álló összegvektoroknak különbözniük kell egymástól, vagyis számuk nem haladhatja meg az  $n$ -bités bináris vektorok számát:

$$\sum_{k=0}^M \binom{T}{k} \leq 2^n$$

Felhasználva, hogy  $\sum_{k=0}^M \binom{T}{k} \simeq T^M$  kapjuk a következő korlátot:

$$n_{ZFD}(T, M) \geq M \log T.$$

Bassalygo alábbi eredménye jellegében eltér a szokásos korlátoktól, azonban támpontot ad arra vonatkozólag, hogy milyen esetekben érdemes az időosztásnál hatékonyabb kódolást keresnünk.

### 2. tétel (Bassalygo-korlát, [8,9,1])

$$n_{ZFD}(M, T) \geq \min \left\{ \frac{(M+1)(M+2)}{2}, T \right\}$$

A tételből következik, hogy ha  $\sqrt{2}T < M < T$ , akkor  $n_{ZFD}(T, M) = T$ , vagyis ha  $M$  értéke ebben a tartományban van, akkor egyetlen ZFD( $T, M$ ) kód sem lehet jobb az időosztásnál.

**3. tétel [8,9,11,15]**

A ZFD( $T, M$ ) kód minimális hosszára jelenleg ismert legjobb alsó korlát,  $T \rightarrow \infty, M \rightarrow \infty$  és  $M \ll T$  esetén:

$$n_{ZFD}(T, M) \geq k \frac{M^2}{\log M} (1 + o(1)) \log T$$

ahol a  $k$  konstans értéke [8] szerint  $1/2$ , [11] szerint  $1/4$ , [15] alapján pedig  $1/8$ .

A és Zeisel [3] a véletlen kódválasztás technikájával adott felső korlátot a minimális kódszóhosszra.

**4. tétel [3]**

Rögzített  $M$  és  $T \rightarrow \infty$  esetén

$$n_{ZFD}(T, M) \leq K(M)M^2(1+o(1)) \log T,$$

ahol  $K(M) \leq 1.5112$ .

Amennyiben  $M \rightarrow \infty$  is teljesül,

$$\limsup_{M \rightarrow \infty} K(M) = \frac{1}{\ln 2} \approx 1.4427$$

*Bizonyítás*

Tekintsünk egy véletlenszerűen választott kódot az  $1, 2, \dots, L$  ( $L \geq M$ ) ábécé felett. A kódszavak hossza  $n/L$ , és az egyes szimbólumokat egymástól függetlenül egyenletes eloszlás szerint sorsoljuk. Ebből a kódból készítünk egy bináris  $C$  kódot úgy, hogy minden szimbólumot  $L$  hosszúságú, 1 súlyú bináris sorozattal helyettesítünk az alábbi leképezés szerint:

$$\begin{aligned} 1 &\Rightarrow 0 \dots 001 \\ 2 &\Rightarrow 0 \dots 010 \\ &\dots \\ L &\Rightarrow 1 \dots 000 \end{aligned}$$

Ekkor a  $C$  kód  $n$  hosszúságú, bináris kódszavakból áll. A  $C$  kód nem teljesíti a ZFD tulajdonságot, amennyiben ki tudunk választani a  $T$  darab kódszó közül  $M$ -et úgy, hogy minden  $L$  hosszú szegmensben legalább egynek olyan pozíció van az 1-es bitje, amely meg egyezik egy tőlük függetlenül választott  $(M+1)$ -edik kódszóéval.

$$\begin{aligned} \mathbf{P} &\leq \binom{T}{M} (T-M) \left(1 - \left(1 - \frac{1}{L}\right)^M\right)^{\frac{n}{L}} \leq \\ (C \text{ nem ZFD}) &\leq \exp\left((M+1) \ln T + \frac{n}{L} \ln \left(1 - \left(1 - \frac{1}{L}\right)^M\right)\right) \end{aligned}$$

Azt kell belátnunk, hogy ez a valószínűség 1-nél kisebb, mert ekkor létezik  $M$ -edrendű ZFD kód. Ehhez az exponenciális függvény argumentumának kell 0-nál kisebbnek lennie:

$$(M+1) \ln T + \frac{n}{L} \ln \left(1 - \left(1 - \frac{1}{L}\right)^M\right) < 0$$

Ebből a kódszóhosszat kifejezve kapjuk, hogy

$$\frac{(M+1)L \log T}{-\log \left(1 - \left(1 - \frac{1}{L}\right)^M\right)} < n_{\text{véletlen}}$$

vagyis a kódszóhosszat ennél nagyobbra választva létezik  $M$ -edrendű ZFD kód, tehát a minimális kódszóhossz ennél nem nagyobb:

$$n_{ZFD}(T, M) \leq \frac{(M+1)L \log T}{-\log \left(1 - \left(1 - \frac{1}{L}\right)^M\right)}.$$

Aszimptotikusan az alábbi felső határt kapjuk:

$$n_{ZFD}(T, M) \leq K(M)M^2(1+o(1)) \log T$$

ahol

$$K(M) = \min_{M \leq L} \frac{\ln 2}{-\frac{M+1}{L} \ln \left(1 - \left(1 - \frac{1}{L}\right)^M\right)}$$

Válasszuk meg a szegmenshosszt  $L = \lfloor \frac{M+1}{\ln 2} \rfloor$  értékre, és alkalmazzuk az  $M \leq L$  esetén fennálló

$$\left(1 - \frac{1}{L}\right)^M \geq \exp\left(-\frac{M+1}{L}\right) \text{ egyenlőtlenséget.}$$

Így kapjuk, hogy

$$K(M) \leq \frac{\ln 2}{-\alpha \ln(1 - e^{-\alpha})} \leq \frac{\ln 2}{-\ln(1 - e^{-1})} \approx 1.5112$$

ahol  $\alpha = \frac{M+1}{\lfloor \frac{M+1}{\ln 2} \rfloor}$ .

Ha  $M \rightarrow \infty$ , akkor  $\alpha \rightarrow \ln 2$ , így

$$\limsup_{M \rightarrow \infty} K(M) = \frac{\ln 2}{-\ln 2 \ln(1 - e^{-\ln 2})} = \frac{1}{\ln 2} \approx 1.4427$$

s ezzel beláttuk a tételt. ■

A minimális kódszóhosszra kapunk felső korlátot abban az esetben is, ha a 4. tételtől eltérően nem konstans súlyú kódszavakkal dolgozunk, hanem a kódszavakban szereplő 1-es bitek száma binomiális eloszlás szerinti.

**5. tétel (Dyachkov és Rykov [9])**

$1 \ll M \ll T$  és  $T \rightarrow \infty$  esetén

$$n_{ZFD}(T, M) \leq e \ln 2 M(M+1) \log T \approx 1.884M^2(1+o(1)) \log T.$$

*Bizonyítás*

Tekintsünk egy  $n$  hosszú kódszavakból álló véletlenszerűen választott bináris kódot. A kódszavak bitjeit egymástól függetlenül sorsoljuk:  $p$  valószínűséggel 1-est,  $1 - p$  valószínűséggel pedig 0-t. Így egy kódszóban lévő 1-esek száma binomiális eloszlást követ. A  $C$  kód nem ZFD tulajdonságú, amennyiben ki tudunk választani a  $T$  darab kódszó közül  $M$ -et úgy, hogy minden olyan pozícióban, ahol egy tőlük függetlenül választott  $(M+1)$ -edik kódszó 1-est tartalmaz, az  $M$  közül legalább egynek szintén van 1-ese.

$\mathbf{P}(C \text{ nem ZFD})$

$$\leq \sum_{k=0}^n \mathbf{P}(\text{minden 1-es fedett} \mid \text{1-esek száma} = k) \mathbf{P}(\text{1-esek száma} = k) \leq$$

$$\begin{aligned} &\leq \sum_{k=0}^n \binom{T}{M} (T-M) (1 - (1-p)^M)^k \binom{n}{k} p^k (1-p)^{n-k} = \\ &= \binom{T}{M} (T-M) \sum_{k=0}^n \binom{n}{k} (p(1 - (1-p)^M))^k (1-p)^{n-k} = \\ &= \binom{T}{M} (T-M) (p(1 - (1-p)^M) + 1-p)^n = \\ &= \binom{T}{M} (T-M) (1 - p(1-p)^M)^n. \end{aligned}$$

Ez a kifejezés a minimumát  $p = \frac{1}{M+1}$  esetén veszi fel, és itt az értéke

$P(C \text{ nem ZFD})$

$$\begin{aligned} &\leq \binom{T}{M} (T-M) \left(1 - \frac{1}{M+1} \left(1 - \frac{1}{M+1}\right)^M\right)^n \leq \\ &\leq \binom{T}{M} (T-M) \left(1 - \frac{e^{-1}}{M+1}\right)^n \leq \\ &\leq \binom{T}{M} (T-M) e^{-\frac{n}{M+1} e^{-1}} \leq \\ &\leq T^M e^{-\frac{n}{M+1} e^{-1}} = \\ &= e^{M \ln 2 \log T - \frac{n}{M+1} e^{-1}} < \\ &< 1. \end{aligned}$$

Mindkét oldal logaritmusát véve kapjuk

$$M \ln 2 \log T - \frac{n}{M+1} e^{-1}$$

és ebből

$$e \ln 2M(M+1) \log T < n_{\text{véletlen}},$$

vagyis a kódszóhosszat ennél nagyobbra választva létezik  $M$ -edrendű ZFD kód, tehát a minimális kódszóhossz ennél nem nagyobb:

$$n_{\text{ZFD}}(T, M) \leq e \ln 2M(M+1) \log T. \quad \blacksquare$$

### 3. Kódkonstrukciók

ZFD kódok létrehozására Kautz és Singleton [14] konstans súlyú módszert javasolt.

Egy kód maximális átlapolásának (maximum overlap) vagy keresztkorrelációjának (cross-correlation) hívjuk azon pozíciók számát, amelyeken két tetszőlegesen választott kódszó közül mindkettőben 1-es áll.

#### 1. lemma [14]

Legyen  $w_{\min}$  a  $C$  kód kódszavainak minimális súlya. Ha a kódszavak közötti maximális átlapolás  $c$ , akkor a  $C$  kód ZFD tulajdonságú legalább  $M_0$ -adrendben, ahol

$$M_0 = \left\lfloor \frac{w_{\min} - 1}{c} \right\rfloor.$$

Ha minden  $c$  elemű szimbólumegyüttes a  $C$  kód legalább két kódszavában szerepel, akkor a kód rendje pontosan  $M_0$ .

#### Bizonyítás

A  $C$  kód rendelkezik a ZFD tulajdonsággal  $M_0$ -adrendben, ugyanis minden kódszó súlya legalább  $w_{\min} \geq M_0 c + 1$ , így egyetlen kódszót sem tud tőle különböző  $M_0$  darab másik kódszó Boole-összege lefedni, hiszen mindegyikkel csak legfeljebb  $c$  pozíción lapol át.

Ha minden  $c$  elemű szimbólumegyüttes legalább két kódszóban szerepel, akkor bármely legfeljebb  $(M_0 + 1)c$  súlyú kódszóhoz található  $M_0 + 1$  másik kódszó, amelyek összege fedi azt. Így  $C$  nem lehet ZFD tulajdonságú  $M_0 + 1$ -edrendben.  $\blacksquare$

#### 6. tétel [14 és 10]

Legyen  $C_Q$  egy  $(n_Q, k)$  paraméterű és  $d_Q$  kódtávolságú GF(Q) feletti kód ( $Q$  prímszámú). Helyettesítsük a  $Q$ -ásis szimbólumokat  $Q$ -hosszú 1-súlyú bináris sorozatokkal a következő módon:

$$0 \Rightarrow 0 \dots 001$$

$$1 \Rightarrow 0 \dots 010$$

...

$$Q - 1 \Rightarrow 1 \dots 000$$

Az így kapott konkatenált  $C$  kód  $T = Q^k$  kódszót tartalmaz, melyek hossza  $n = Qn_Q$ , és legalább  $M$ -edrendben ZFD tulajdonságú, ahol

$$M \geq \left\lfloor \frac{n_Q - 1}{n_Q - d_Q} \right\rfloor.$$

#### Bizonyítás

Nyilvánvalóan  $T = |C| = |C_Q| = Q^k$ ,  $n = Qn_Q$ . A bináris kód Hamming-távolsága a  $Q$ -ásis távolság kétszerese:  $d = 2d_Q$ , és minden kódszó  $w = n_Q$  súlyú. A bináris kód maximális átlapolása

$$c = w - \frac{d}{2} = n_Q - d_Q,$$

melyből az 1. lemma miatt következik az állítás.  $\blacksquare$

Vizsgáljuk meg a Kautz–Singleton-konstrukció néhány speciális esetét!

#### Reed–Solomon kód [2, 10, 17]

Legyen  $C_Q$  egy maximális,  $n_Q = Q - 1$  hosszú Reed–Solomon-kód. A  $C$  konkatenált kódnak  $T = Q^k$  kódszava van, s mindegyiknek  $w = n_Q = Q - 1$  súlya.

Mivel a Reed–Solomon-kód maximális távolságú, így  $d_Q = n_Q - k + 1$  (lásd Györfi, Györi és Vajda [13]), amiből

$$c = n_Q - (n_Q - k + 1) = k - 1.$$

Egy maximális távolságú kódban bármely  $k$  szimbólum lehet üzenet, így minden  $c = k - 1$  elemű szimbólumegyüttes pontosan  $Q$ -szor ismétlődik a bináris kódban. Az 1. lemmából következően a ZFD tulajdonság rendje pontosan

$$M = \left\lfloor \frac{w - 1}{c} \right\rfloor = \left\lfloor \frac{Q - 2}{k - 1} \right\rfloor.$$



Felhasználva, hogy  $k = \frac{\log T}{\log Q}$  kapjuk az alábbi egyenlőtlenséget:

$$\frac{\log T}{\log Q} \leq \frac{Q-2}{M} + 1.$$

Ha adott  $T, M$ , és egy minimális hosszúságú kódot szeretnénk előállítani a Kautz–Singleton-konstrukcióval, akkor meg kell keresnünk ennek az egyenlőtlenségnek eleget tevő legkisebb (prímhatvány)  $Q$  értéket, s ekkor a kódszóhossz  $n = Qn_Q = Q(Q-1)$ .

### BCH kód [12]

Legyen  $C_Q$  egy maximális,  $n_Q = Q^r - 1$  hosszú BCH-kód valamely  $r \geq 2$ -re. Ekkor a  $C$  konkatenált kódnak  $T = |C_Q| = Q^{(k-1)r+1}$  kódszava van, melyek hossza  $n = Qn_Q = Q(Q^r - 1)$ , súlyuk pedig  $w = n_Q = Q^r - 1$ .  $C_Q$  minimális távolságára alsó becslést adhatunk [4]:

$$d_Q \geq Q^r - 1 - (k-1)Q^{r-1}.$$

Ekkor a  $C$  bináris kód ZFD tulajdonságának rendje legalább

$$\begin{aligned} M &\geq \frac{n_Q - 1}{n_Q - d_Q} \\ &\geq \frac{(Q^r - 1) - 1}{(Q^r - 1) - (Q^r - 1 - (k-1)Q^{r-1})} \\ &= \frac{Q^r - 2}{(k-1)Q^{r-1}} \\ &\simeq \frac{Q}{k-1}, \end{aligned}$$

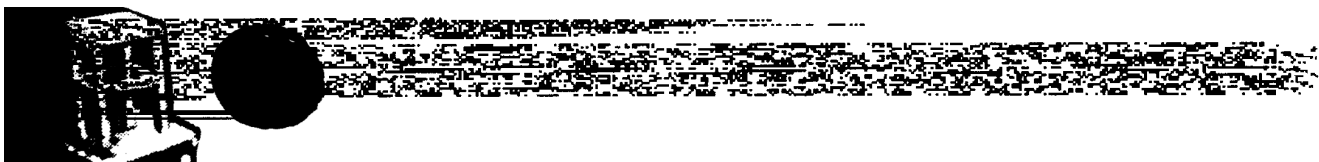
ami hozzávetőleg megegyezik a Reed–Solomon-kód esetében kapott értéknek. A BCH-kód használatának előnye, hogy a lehetséges felhasználók számára még kis  $r$  érték esetén is hatalmas számot kapunk, bár az is igaz, hogy a minimális kódszóhossz szintén nagyobb a Reed–Solomon-kódéhoz képest.

A kódkonstrukciókról részletes tanulmány olvasható Dyachkov, Macula és Rykov [7] írásában.

### Irodalom

- [1] N. Q. A., Some coding problems of multiple-access communication systems. DSc dissertation, Hungarian Academy of Sciences, 1986.
- [2] N. Q. A., L. Györfi, and J. L. Massey, Constructions of binary constantweight cyclic codes and cyclically permutable codes. *Problems of Control and Information Theory*, 38(3):940–949, 1992.
- [3] N. Q. A and T. Zeisel, Bounds on constant weight binary superimposed codes. *Problems of Control and Information Theory*, 17(4):223–230, 1988.

- [4] R. E. Blahut, *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley, 1984.
- [5] R. T. Chien and W. D. Frazer, An application of coding theory to document retrieval. *IEEE Transactions on Information Theory*, 12(2):92–96, 1966.
- [6] A. R. Cohen, J. A. Heller, and A. J. Viterbi, A new coding technique for asynchronous multiple access communication. *IEEE Transactions on Communication Technology*, 19:849–855, 1971.
- [7] A. G. Dyachkov, A. J. Macula, and V. V. Rykov, New constructions of superimposed codes. *IEEE Transactions on Information Theory*, 46(1):284–290, 2000.
- [8] A. G. Dyachkov and V. V. Rykov, Bounds on the length of disjunctive codes. *Problemy Peredachi Informatsii*, 18(3):7–13, 1982.
- [9] A. G. Dyachkov and V. V. Rykov, A survey of superimposed code theory. *Problems of Control and Information Theory*, 12(4), 1983.
- [10] P. Erdős, P. Frankl, and Z. Füredi, Families of finite sets in which no set is covered by the union of  $r$  others. *Israel Journal of Mathematics*, 51(1-2):79–89, 1985.
- [11] Z. Füredi, On  $r$ -cover-free families. *Journal of Combinatorial Theory*, 73:172–173, 1996.
- [12] L. Györfi and I. Vajda, Constructions of protocol sequences for multiple access collision channel without feedback. *IEEE Transactions on Information Theory*, 39(5):1762–1765, 1993.
- [13] Györfi L., Györfi S. és Vajda I., *Információ- és kódelmélet*. TypoTEX, Budapest, 2002.
- [14] W. H. Kautz and R. C. Singleton, Nonrandom binary superimposed codes. *IEEE Transactions on Information Theory*, 10:363–377, October 1964.
- [15] M. Ruszinkó, Note on the upper bound of the size of the  $r$ -cover-free families. *Journal of Combinatorial Theory*, 66(2):302–310, 1994.
- [16] I. Vajda, Code constructions for code division multiple access channels. *Journal on Communications*, 45:2–9, 1994.
- [17] V. A. Zinoviev, Cascade equal-weight codes and maximal packings. *Problems of Control and Information Theory*, 12(1):3–10, 1983.



# Aspektus-orientált programozás

LENGYEL LÁSZLÓ, LEVENDOVSKY TIHAMÉR

BME, Automatizálási és Alkalmazott Informatikai Tanszék  
lengyel@aut.bme.hu, tihamer@aut.bme.hu

Reviewed

**Kulcsszavak:** aspektus-orientált programozás, átszövő vonatkozások

Az aspektus-orientált programozás (AOP) egy szerencsés kiegészítése a ma leginkább elterjedt objektum-orientált paradigmának. Jelen cikkben bemutatjuk az AOP fontosabb koncepcióit, a legelterjedtebb megvalósítás, az AspectJ szemléletét alapul véve. Bevezetést adunk az átszövő vonatkozások problémájába, majd megoldásként sorra vesszük az AOP adta lehetőségeket. Röviden megemlíti a legnépszerűbb megvalósításokat (AspectJ, HyperJ, kompozíciós szűrők) is.

## Bevezetés

Napjaink domináns programozási paradigmája az objektum-orientáltság (OO). Az objektum-orientált programozás (OOP) jól használható megoldást adott az áttekinthető és ilyen módon biztonságosabb programok írásához, valamint a kód-újrafelhasználhatóság követelményére. Ez okozta széleskörű elterjedését, és viszonylagos egyeduralmát. Az objektum-orientáltság mögött az a szemlélet húzódik meg, hogy az elkészítendő program önálló entitások, úgynevezett objektumok összessége, melynek működését ezen objektumok kommunikációja valósítja meg. Ez a módszer a tapasztalatok szerint áttekinthető viszonylag komplex feladatok esetén is [1,2].

Ha egy komplex feladatot objektumokra dekomponálunk, elsősorban az önálló entitások létrehozására koncentrálunk, valamint arra, hogy az adatokat és a hozzájuk kapcsolódó műveleteket egységbe zárjuk. Így azonban figyelmen kívül kell hagynunk sokkal fontosabb logikai rendező elveket, csoportosítási szempontokat, mint például a perzisztencia, az elosztottság, a hibakövetés, amelyek így a programkódban jellemzően elszórtan jelennek meg, nehézkessé téve a rendszer megértését és így a karbantartását is. Ezeket a szétszóródott és a program különálló egységein áthúzódó, de logikailag egybe tartozó kódrészleteket *átszövő vonatkozásoknak* (crosscutting concerns) nevezzük.

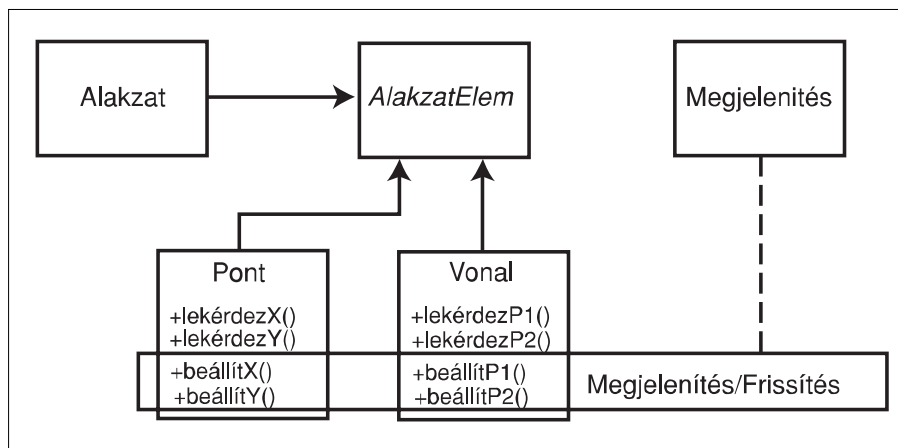
Egy példa az átszövő vonatkozásokra a program futásának nyomon követése. Elosztott rendszereknél az alkalmazás gyakran ír naplófájlt, amely az alkalmazás hibás működése esetén segíti a nyomkövetést azáltal, hogy minden függvényhívást és kivételt tartalmaz. A napló fájl írásához minden osztálynak tartalmaznia kell naplózást megvalósító pro-

gramsorokat, általában szétszórva, holott a naplózást végrehajtó kódrészletek nagyon szoros kapcsolatban állnak egymással: egy funkciót valósítanak meg.

Egy másik lehetséges példához [3] vegyük szemügyre az 1. ábrán látható egyszerű alakzatszerkesztő UML osztálydiagramját. Az *AlakzatElem* két konkrét lezármazott osztályát láthatjuk: az egyik a *Pont*, a másik, pedig a *Vonal*. Az osztályokra bontás ígéretesnek bizonyul: mindkét osztály jól definiált interfésszel rendelkezik, valamint az adatok és a rajtuk végzett műveletek egységbezárását is sikerült megvalósítanunk. De gondoljuk át a képernyőkezelő szemszögéből a dolgot, aminek értesítést kell kapnia minden elem mozgásáról. Ez megköveteli, hogy minden függvény, amely mozgást végez, a mozgás után értesítse a képernyőkezelőt.

Az ábrán a *Megjelenítés/Frissítés* téglalap azokat a függvényeket keretezi be, amelyeknek ezt a feladatot kell megvalósítaniuk. Hasonlóan a *Pont* és *Vonal* téglalapok azokat a függvényeket keretezik be, amelyek a hozzájuk tartozó vonatkozást valósítják meg. Látható, hogy a *Megjelenítés/Frissítés* téglalap nem illik az ábra semelyik másik téglalapjába, hanem átvágja, átszövi azokat.

1. ábra Átszövő vonatkozások



Az OO lehetőségeket kiegészítve, az átszövő vonatkozások problémájára egy megoldást kínál az *aspektus-orientált programozás* (Aspect Oriented Programming, AOP) [4], amely a programot alkotó kódot az alapján osztja részekre, hogy azok milyen szempontból járulnak hozzá a program működéséhez. Aspektus-orientáltan megközelítve a problémát, az egyes szempontokat *aspektusokba* tömörítve külön, egymástól viszonylag függetlenül kódoljuk, majd egy *aspektus-szövő alkalmazás* (aspect weaver) segítségével egyesítjük. Az egybeszövés az adott megközelítéstől függően történhet akár futási időben dinamikusan, akár a fordítás előtt vagy során statikusan.

Objektum-orientált programozást használva az átszövő vonatkozások megvalósítása szétszóródik az egész rendszerben. Az AOP mechanizmusait használva a *Megjelenítés/Frissítés* téglalap által határolt vonatkozás megvalósítása egyetlen aspektusban elvégezhető, emellett tervezési szinten is lehetővé válik az aspektusokban való gondolkodás, és így a modularitás megvalósítása is.

A modularitás lényege, hogy programunk logikailag egy egységet képező részeit egy fizikai egységben adhassuk meg. Általános elv, hogy a modulok belső kohéziója minél erősebb legyen, míg a modulok egymással lazán kapcsolódjanak. Az absztrakció segítségével különböző elemekből ki tudjuk emelni azok közös jégeit.

Míg a vonatkozások szétválasztása és a modularitás segítségével horizontális módon szervezhetjük a programunkat, addig az absztrakció vertikális jellegű. Egy programozási paradigmát illetve technológiát legfőbb mértékben az határoz meg, milyen típusú absztrakciót használ fel. A gyakori, ismétlődő kódrészletek, illetve kódminták az adott programozási technika absztrakciós képességeinek hiányát jelentik. A redundancia azért kellemetlen, mert egy kisebb változtatás a programtervben azt eredményezheti, hogy számos, egymással kapcsolatban nem álló modulban kell változtatást elvégezni.

## Átszövő vonatkozások

A vonatkozások (concerns) elkülönítése azt a képességet jelenti, mellyel azonosítjuk és kiemeljük a szoftver azon részeit, melyek egy adott szándékot, célt valósítanak meg. A vonatkozások elkülönítésének elsődleges célja, hogy a szoftvert kezelhető és könnyebben érthető részekre bontsuk szét. Természetes kérdés, hogy miként végezzük ezt a felbontást. Mely funkciók kerüljenek osztályokba, melyek aspektusokba?

Fontos látni, hogy az átszövés egyéni felbontásokra vonatkozik, ugyanis az átszövő vonatkozásokat nem lehet teljesen elkülöníteni egymástól. Az alapvető tervezési szabály, hogy vegyük az alapvető vonatkozásokat, mint első szintű absztrakciót, valósítsuk meg őket osztályokkal, és majd később elvégezzük a kiterjesztésüket aspektusokkal, ha szükséges lesz. Az alak-

zatszerkesztő példánál két fontos tervezési vonatkozás van: a grafikai elemek reprezentálása és a grafikai elemek mozgásának a követése.

Az 1. ábrán látható osztályok az első vonatkozást képviselik. Minden grafikai osztály magában foglalja saját belső adatstruktúráját, amelynek kiterjesztésére az aggregáció és az öröklés biztosít lehetőséget. A második vonatkozást, az elemek mozgásának a nyomon követését ugyancsak egy külön osztályként kellene ábrázolni, de az első vonatkozás megakadályozza ezt, mert az egységbezárás miatt a mozgást megvalósító függvények a grafikai osztályok részei. A rendszert a mozgás nyomon követése köré is tervezhetjük, de abban az esetben a grafikai funkcionalitás fogja átszőni a nyomon követés osztályait. Melyik megoldást érdemes alkalmaznunk?

A probléma tisztázható például a domináns felbontás segítségével. A szoftvert – a könyvekhez hasonlóan – folyó szöveggént írjuk, ahogy a könyv is fejezetekből és bekezdésekből áll össze, a szoftvernek is vannak moduljai, mint például az osztályok. A modulok, amelyek a domináns felbontást alkotják, egységes vonatkozásokat tartalmaznak és legtöbbször *önállóan futtathatók*. Egy domináns modul nem tartalmazhat olyan vonatkozást, amely számos más modult sző át. Ezek átszövő vonatkozások lesznek.

Jellegzetes átszövő vonatkozások például a szinkronizáció, a monitorozás, a pufferezés, a tranzakciókezelés vagy a környezetfüggő hibakezelés. Az átszövő vonatkozások egész magas szintűek is lehetnek, mint például az adatbiztonság vagy a szolgáltatásminőség aspektusai.

## Aspektus-orientált programozás

Az aspektus-orientált programozási paradigma a kilencvenes évek közepén született meg, ma már a programozási nyelvekkel kapcsolatos kutatások egyik nagyon fontos területe, és várhatóan a közeljövőben széleskörűen el fog terjedni.

Joggal mondhatjuk, hogy a Fortran óta minden programozási nyelv rendelkezik azzal a képességgel, hogy alprogramokba különítse el a vonatkozásokat. Az alprogramok máig jól használhatók, és mint ahogy az objektum-orientált programozás nem tud elszakadni a blokk struktúráktól és a strukturált programozástól, ehhez hasonlóan az AOP sem veti el a már meglévő technológiákat. Gyakori, hogy a vonatkozások nem valósíthatók meg egy egyszerű eljáráshívással, ugyanis egy vonatkozás összekeveredik más strukturális elemekkel, kuszassá válik. Az alprogramok másik hátránya, hogy megköveteli, hogy a hívó komponens programozók is tudatában legyenek a vonatkozásoknak, tudják, hogyan kell őket beiktatni, használni. Az AOP az alprogramok mellett olyan hívási mechanizmust kínál, amely lehetővé teszi, hogy a hívó komponens fejlesztőinek ne kelljen tudniuk a kiegészítő vonatkozásokról, vagyis az alprogram meghívásáról.

Az aspektus-orientált programozás két legfontosabb alapelve az átszövő vonatkozások elkülönítése és a modularitás [5]. Az AOP egyik felismerése, hogy a moduláris egységek határai ritkán esnek egybe a vonatkozások határvonalával. A modularizálás bizonyos vonatkozásoknál megoldható, de az átszövő vonatkozások megvalósító kód szétszórva található meg a programban, átszöve az egyes moduláris egységeket. Az AOP célkitűzése, hogy ezeket az átszövő vonatkozásokat is önálló moduláris egységekbe lehessen szervezni, a programtermék (kód és/vagy terv) bonyolultságát csökkentse, a karbantarthatóságát, olvashatóságát javítsa, újrafelhasználhatóságát megkönnyítse.

Egy aspektus a megvalósítás egy moduláris egysége, olyan viselkedést foglal magába, amely több osztályra is hatással van. Az AOP segítségével először elkészítjük az alkalmazást egy tetszőleges objektum-orientált nyelven (például C++, Java vagy C#), majd ezt követően külön foglalkozunk az átszövő vonatkozásokkal, amely az aspektusok beépítését jelenti. Végül az aspektusszöveget alkalmazva a kód és az aspektusok kombinációjaként készül el a futtatható alkalmazás, melynek eredményeként az aspektus számos függvény, modul vagy objektum megvalósításának lesz a része, és ezzel növeli mind az újrafelhasználhatóságot, mind a karbantarthatóságot.

A szövő folyamat a 2. ábrán látható. Fontos, hogy az eredeti kódnak semmit sem kell tudnia a hozzáadott aspektusról, az aspektusszövő nélkül lefordítva az eredeti alkalmazást kapjuk, a szöveget és az aspektusokat felhasználva pedig az aspektusok funkcionalitásával kiegészített alkalmazást. Ez azt jelenti, hogy az eredeti kódon semmit sem kell változtatni, mindkét esetben ugyanabban a formában használható.

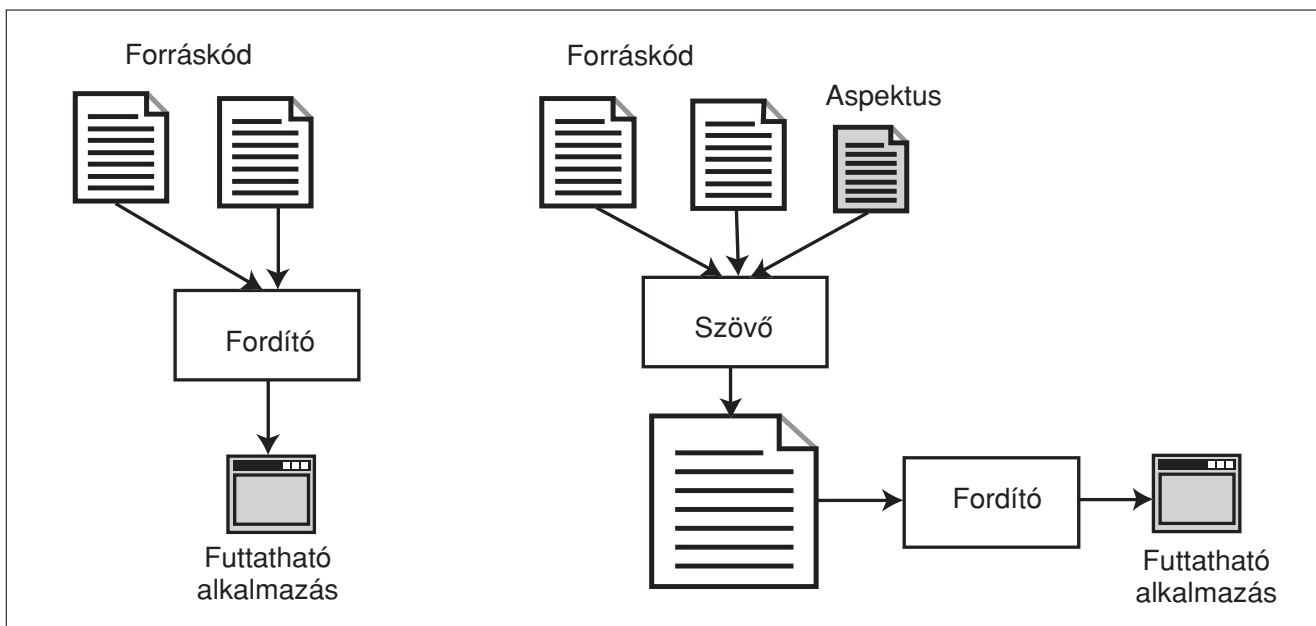
Az AOP kiegészíti, és nem helyettesíti az objektum-orientált programozást. Egy más típusú felbontást kínál, az osztályok mellett új modularizációs elemeket ve-

zet be, melynek feladata, hogy az osztályoktól külön, azokból kivonva valósítsa meg az egyes aspektusokat. Az AOP az OOP-re épül, nem tekintjük idejétmúltnak az objektumokat, függvényeket, hanem az aspektusokkal együtt tovább használjuk őket, mindig a legmegfelelőbbet.

Az AOP azáltal nyújt többletet, hogy a függvény jelöli ki a meghívás helyét és nem a hívó, amely nem is tud az egésztől. Ha a hívott programrészletnél megadjuk a hívás helyét, akkor ez automatikusan hozzászöveődik és a megvalósításban természetesen függvényhívás lesz belőle, de ezt a szövő a programozó beavatkozása nélkül, automatikusan kezeli). Ez azért hasznos, mert, ahogy az már a fentiekben szerepelt, a hívó kód magától a kiterjesztésétől függetlenül is le tud futni. Például egy operációs rendszer a memórialapozáshoz egy úgynevezett piszkos bitet (dirty bit) használ, amely azt jelzi, hogy egy memórialappon történt-e valamilyen változás, ugyanis amennyiben történt, akkor el kell menteni a megváltozott adatokat a háttértárolóra. AOP technikákkal a piszkos bit kezelése szétválasztható a memórialapozástól: a rendszer futtatható piszkos bit kezeléssel, vagy anélkül, valamint a piszkos bit kezelését végző függvények fizikailag egy helyen vannak, és nem a hívás helyétől függően a forráskódban elszórva.

Felmerülhet a kérdés: hogyan adjuk meg, hogy egy adott kódrészlet hol hívódjon meg, és mindezt úgy, hogy a hívó kódrészletben ez egyáltalán ne jelenjen meg, a megoldás a csatlakozási pont. A csatlakozási pontok a programnak azokat a jól meghatározott pontjait jelentik, amelyekben az aspektus interakciója történik a rendszer többi részével. Attól függően, hogy a csatlakozási pontok a program szövegének vagy futásának pontjait jelentik, statikus illetve dinamikus csatlakozási pontokról beszélünk. Statikus kapcsolódási pontok például a naplózás esetén minden publikus függ-

2. ábra Aspektusszövő



vény első utasítása, amely eredménye egy olyan napló fájl lesz, ahol nyomon követhető a függvényhívások sorrendje. A dinamikus kapcsolódási pontok a programfutás olyan eseményeihez kötődnek, mint metódus-hívás, hívás fogadása, futása, továbbá az attribútum lekérdezés, kivétel dobása, osztály inicializáció, objektum inicializáció.

Az aspektusok újrafelhasználhatósága az AOP egyik alapvető eredménye. Az egyszerű, kisméretű aspektusok készítése még inkább újrafelhasználhatóvá teszi az egyes kódrészleteket. A tapasztalatok felhasználásával, az aspektusok összegyűjtésével aspektuskönyvtár készíthető. Ehhez kapcsolódó kérdés, hogyan bánjunk a sok aspektussal, hogyan készítsük el őket, milyen jelölést használunk a leírásukra. Ezek és a hasonló kérdések azok, amelyeket a felhasználók és a kutatócsoportok fognak felderíteni az elkövetkező néhány évben.

Az egyik legfontosabb, még jelenleg is nyitott kérdés az aspektusok szemantikai helyessége. A komponens alapú rendszereknél mindig is kérdés volt, hogy miként biztosítsuk az egyes komponensek helyes működését. Az aspektus alapú megközelítés által kínált csatlakozási pont alapú összetétel sokkal gazdagabb mechanizmusokat kínál, mint az interfész vagy üzenet alapú kapcsolatok.

Az egyes aspektusokat mind a specifikáció, mind pedig a komponens teszt szempontjából széleskörűen kell megvizsgálni. Amennyiben egy aspektust újrafelhasználunk, nincs garancia arra, hogy minden helyen, ahol ezután újrafelhasználjuk, az elvárásoknak megfelelően működik. Meg kell találni a módját, hogy leírjuk az újrafelhasználásra szánt aspektusok speciális környezetbeli működését.

Az AOP ismertetése után röviden megemlíthetjük a három legelterjedtebb AOP megvalósítást.

## AspectJ

Az AspectJ (<http://www.aspectj.org>) a Java nyelv természetes kiterjesztése: minden Java program egyben AspectJ program is. Az AspectJ a következő új programozási konstrukciókat vezet be a Javába az AOP alapfogalmak számára [6,7]:

- *aspektus* (aspect): Új programozási egység, amely kiemeli az átszövő vonatkozásokat. Az aspektusok tartalmazzák a vágási pontok, a tanácsok és bevezetések definícióit, emellett az osztályokhoz hasonlóan lehetnek adattagjaik és metódusaik is.
- *vágási pont* (pointcut): Dinamikus csatlakozási pontok halmazát jelöli ki egy logikai kifejezés segítségével. Ezeket vágáspont-leíróknak (pointcut descriptor) hívjuk. A vágási pontok paraméterezhetők, a paramétereken keresztül a vágási pont környezetének objektumait tudjuk átadni a tanácsnak.
- *tanács* (advice): A metódushoz hasonló programozási egység. A tanácsok mindig egy adott vágási ponthoz kapcsolódnak. Törzsük tartalmazza azt a

viselkedést, melyet az adott vágási pont által leírt csatlakozási pontokban automatikusan végrehajt.

- *bevezetés* (introduction): A bevezetéssel új adattagokat és metódusokat tudunk definiálni az osztályokban. (A csatlakozási pontot itt az osztály jelenti.)

Annak érdekében, hogy futás közben egyértelműen eldönthető legyen, hogy egy belépési pontra kapcsolódó több aspektus elemei milyen sorrendben futhatnak, az aspektusok között és az aspektusokon belül jól definiált precedencia relációk állnak fenn [8].

Az AspectJ-ben [9,10] az osztályok és az aspektusok nem azonos szintű programozási egységek. Míg az osztályok önálló entitásoknak tekinthetők, addig az aspektusok csak azokkal az osztályokkal együtt értelmezhetőek, amelyeknek az átszövő kódját tartalmazzák. Úgy tekinthetjük, hogy az aspektusok az eredeti program kódjában végzett változtatásokat tartalmazzák. Az aspektusok ennek megfelelően önállóan nem fordíthatóak, újrafelhasználásuk csak forrásszinten lehetséges. A jelenlegi AspectJ megvalósításnál az egybeszövés a fordítás során történik, és szükség van az alapprogram forráskódjára is [11].

## Hiperterek – HyperJ

A hipertér-technológia (<http://www.research.ibm.com/hyperspace/index.htm>) megközelítésének alapelveül a vonatkozások többdimenziójú szétválasztása szolgál. Az elv szerint egy szoftverben egyszerre különböző típusú vonatkozások vannak, ezzel szemben a ma elterjedt nyelvek és módszerek csak egy fajta vonatkozás szerinti dekompozícióra nyújtanak lehetőséget. Ezt a jelenséget nevezzük a domináns dekompozíció egyeduralmának.

A domináns dekompozíció alapja OO nyelveknél az osztály, funkcionális nyelveknél a függvény, szabályalapúaknál a szabály.

A hiperterek lehetővé teszik vonatkozások tetszőleges dimenziójának explicit azonosítását a szoftver életciklusának tetszőleges szakaszában. Ehhez a hipertérmodell a következő fogalmakat használja:

- *hipertér*: A vonatkozások azonosítását szolgálja. A hipertér a szoftvert felépítő egységek egy halmazát jelenti. Ilyen egység lehet például OO környezetben egy osztály, egy metódus vagy egy adattag. Ezeket az egységeket a hipertér egy többdimenziójú mátrixba szervezi. A hipertér képzeletbeli koordináta-rendszerében a vonatkozások jellegük szerinti felosztását (a vonatkozások dimenzióit) jelentik, a koordináták pedig a dimenzióon belüli konkrét vonatkozásokat. A hipertérben szereplő egységek koordinátái kijelölik, hogy az adott egység mely vonatkozások leírásában játszik szerepet.
- *hipersík*: A vonatkozások egységbe zárását szolgálja. Az egy vonatkozáshoz tartozó egységek egy hipersíkon helyezkednek el. A hipersíkok megadásával foghatjuk össze a vonatkozásokat érintő egységeket

- *hipermodul*: A vonatkozások integrálására szolgál. Egy hipermodul magában foglalja az integrálandó hipersíkok egy halmazát, valamint az integrációs kapcsolatokat, melyek a hipersíkok egymáshoz való viszonyát és az integrálás módját írják le.

A HyperJ a vonatkozások többdimenziójú szétválasztásának Java nyelvű megvalósítása [12]. A HyperJ az integrációt nem forráskódon, hanem a lefordított hipersík csomagokon végzi, így lehetőség van már létező alkalmazások újrafelhasználására újra modularizálással. A csatlakozási pontok statikusak, helyüket a hipermodul specifikációja tartalmazza.

## Kompozíciós szűrők

Mivel az objektum-alapú rendszerekben a viselkedést az objektumok közötti üzenetváltás valósítja meg, az objektumok bejövő és kimenő üzeneteinek (tipikusan függvényhívásainak) manipulálásával a viselkedésbeli változtatások rendkívül széles skálája elvégezhető.

A *kompozíciós szűrők* ([http://trese.cs.utwente.nl/composition\\_filters/](http://trese.cs.utwente.nl/composition_filters/)) modelljében az üzenetek vizsgálatát és manipulációját erre a célra tervezett szűrők végzik. A modell az átszövő vonatkozásokat az objektumok moduláris és független kiegészítéseként fejezi ki. A modularitás itt abban nyilvánul meg, hogy a szűrők jól definiált interfésszel rendelkeznek, és koncepcionálisan függetlenek az objektum megvalósításától [13,14]. A szűrők függetlenek egymástól, vagyis a specifikációjukban nem szerepel hivatkozás egyéb szűrőkre.

## Összegzés

Az AOP egy koncepció, egy olyan elképzelés, amely nem kötődik egyik programozási nyelvhez sem. Valójában, egyszerű és hierarchikus felbontást alkalmazva, a már létező programozási nyelvek elemeit megtartva és felhasználva képes kiküszöbölni a (nem csak objektum-orientált) programozási nyelvek hiányosságait.

Az AOP koncepcióit már több különböző nyelvre kidolgozták: C, C++, Java, Perl, Python, Ruby, SmallTalk és C#. (A Java az a nyelv, amely iránt a kutató közösségekben a legnagyobb érdeklődés mutatkozik, a legfejlettebb AOP környezetek ezen a nyelven állnak rendelkezésre.)

A szoftverfejlesztés területén manapság egyre nagyobb szerepet kapnak a szoftver hosszú távú életciklusát érintő kérdések. Ilyen a fejleszthetőség, a karbantarthatóság, a követelmények gyakori változásához való alkalmazkodóképesség vagy az újrafelhasználhatóság. Az AOP ezeket a célokat egy, az OO-nál rugalmasabb, magasabb absztrakciós szintű modell biztosításával segíti.

A téma irodalomjegyzéke igen bőséges. Az érdeklődő olvasóknak további információk végett a CACM AOP-nek szentelt különszámát [15] ajánljuk, illetve az egyes megvalósítások web oldalait.

## Irodalom

- [1] Czarnecki, K. and Eisenecker, U.W.: Generative Programming: Methods, Tools and Applications. Addison Wesley, Boston, 2000.
- [2] Dijkstra, E.W.: A Discipline of Programming. Prentice-Hall, 1976.
- [3] Tzilla Elrad, Mehmet Aksit, Gregor Kiczales, Karl Lieberherr, Harold Ossher: Discussing Aspects of AOP, CACM Volume 44, Issue 10 (October 2001)
- [4] Aspect-Oriented Software Development. <http://www.aosd.net/>
- [5] Tzilla Elrad, Robert E. Filman and Ataf Bader: Aspect-oriented Programming, CACM Volume 44, Issue 10 (October 2001)
- [6] Gregor Kiczales, Erik Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm and William G. Griswold: An Overview of AspectJ. Jorgen Lindskov Knudsen (Ed.): Proceedings of the 15th ECOOP, Budapest 2001, pp.327–353.
- [7] Gregor Kiczales, Erik Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm, and William G. Griswold: Getting started with AspectJ, CACM Volume 44, Issue 10 (October 2001)
- [8] Kiczales, G., et al.: An overview of AspectJ. In Proceedings of the 15th European Conference on Object-Oriented Programming (ECOOP). Springer, 2001.
- [9] The AspectJ Programming Guide, <http://www.aspectj.org>
- [10] Bill Griswold, Erik Hilsdale, Jim Hugunin, Wes Isberg, Gregor Kiczales, Mik Kersten: Aspect-Oriented Programming with AspectJ, <http://www.aspectj.org>
- [11] The AspectJ Primer; <http://www.aspectj.org/doc/primer>.
- [12] Peri Tarr, Harold Ossher: Hyper/J User and Installation Manual <http://www.research.ibm.com/hyperspace> <http://www.math.klte.hu/~epakm/GOF/hires/Pictures/>
- [13] Mehmet Aksit, Bedir Tekinerdogan: Solving the modeling problems of object-oriented languages by composing multiple aspects using composition filters, 1998.
- [14] Mehmet Aksit, Lodewijk Bergmans: Software evolution problems using composition filters. ECOOP 2001, Budapest
- [15] Communications of the ACM Volume 44, Issue 10 (October 2001)

# Szolgáltatásminőségi paraméterek előrejelzésének javítása outlierek detektálásával és eltávolításával

KOVÁCS LÁSZLÓ, VASS DOROTTYA, VIDÁCS ATTILA

BME, Távközlési és Médiainformatikai Tanszék

[kovacs@alpha.tmit.bme.hu](mailto:kovacs@alpha.tmit.bme.hu), [vass@alpha.tmit.bme.hu](mailto:vass@alpha.tmit.bme.hu), [vidacs@tmit.bme.hu](mailto:vidacs@tmit.bme.hu)

**Kulcsszavak:** outlier detektáló algoritmusok, jellemzők számítása, szolgáltatásminőségi paraméterek előrejelzése

Az Interneten széles körben terjedő valós idejű alkalmazások szükségessé teszik, hogy a szolgáltatók megfelelő szolgáltatásminőségi paramétereket biztosítsanak a felhasználóknak. Ehhez szükséges a hálózati forgalom minél pontosabb előrejelzése. Ennek egy lehetséges megoldása a forgalom mérése, a mért adatok elemzése, majd ezt követően az adathalmazra megfelelő modell illesztése, végül ez alapján a kívánt paraméter meghatározása. Az előrejelzés pontosságát azonban nagymértékben rontják az adatsorban található kiugró minták (outlierek). Kifejlesztettünk egy eszközt, amely gyorsan és automatikusan képes detektálni és eltávolítani az outliereket az idősorokból. Ezenfelül javasolunk egy, az ARIMA (autoregresszív integrált mozgóátlag) modellt használó előrejelzési eljárást is az előzetes outlier detektálással és eltávolítással kiegészítve. Megmutatjuk, hogy ez az eljárás jelentősen növeli az előrejelzés hatékonyságát.

## 1. Bevezetés

A végfelhasználók számára fontos, hogy a szolgáltatók megfelelő szolgáltatásminőségi paramétereket tudjanak garantálni. A szolgáltatóknak szükségük van a mért adatokon alapuló minél pontosabb előrejelzésre, hogy meg tudják tervezni az erőforrások, az alkalmazások és a felhasználók összerendelését. Az előrejelzésnek figyelembe kell vennie, hogy a legtöbb mért adatsor tartalmaz outliereket, melyek helyi véletlen események által okozott extrém ingadozások. Ezek speciális minták (kiugró értékek), melyek nem követik az adatok többségének jellegzetes eloszlását. Az outliereknek jelentős hatásuk lehet a statisztikai becslésre az adatok analízise és a modellezése során.

Az általunk javasolt eljárás kombinálja az előzetes outlier detektálást és eltávolítást a szolgáltatásminőségi paraméterek ARIMA [11] modellen alapuló előrejelzésével. Ezt a modellezési eljárást gyakran használják különböző kutatásokban a hálózati forgalom és a szolgáltatásminőségi paraméterek leírására, előrejelzésére.

Hasonló megközelítéssel találkozhatunk közgazdasági területen [1], amely alapján pénzügyi idősorok előrejelzésére egy ingyenes szoftver is született [2]. A szoftver közgazdasági jellegéből adódóan csak kis adathalmazok kezelésére alkalmas, a távközlési hálózatokban végzett mérések során azonban gyakran nagy adathalmazok keletkeznek. Ezért kifejlesztettünk egy új szoftvert, amely képes nagy adathalmazok kezelésére is. Az általunk kifejlesztett alkalmazás gyorsan távolítja el az idősorokból az outliereket, és az előrejelzés számára megfelelőbb adatokkal helyettesíti azokat. A kifejlesztett algoritmus az úgynevezett L.O.C.I. [3] algoritmuson alapul. A program képes az algoritmus paramétereinek automatikus meghatározására is.

A működés helyességét mesterséges és valós adatsorokon is ellenőriztük, és az eredmények azt mutatják,

hogy az outlierek detektálása és eltávolítása gyors és hatékony.

A cikk felépítése a következő: a második fejezet az outlierek detektálásáról szól. Az általunk használt algoritmust a harmadik fejezet tárgyalja. A negyedik fejezet a kifejlesztett eszközt mutatja be, és vizsgálja a működés helyességét is. Az ötödik fejezetben az előzetes outlier detekció és eltávolítás hatását vizsgáljuk az előrejelzés jóságára. Végül a hatodik fejezet összegzi a fentieket és a továbbfejlesztési lehetőségeket.

## 2. Outlierek detektálása idősorokban

D. Hawkins szerint [10]: „Az outlier az a megfigyelés, amely annyira eltér a többi megfigyeléstől, hogy azt a gyanút kelti, hogy valamilyen más mechanizmus hozta létre.” Az outlierek az idősorban található, az előrejelzés szempontjából értelmetlen adatok, melyek különböző véletlen hibák hatására keletkeznek. Az outlierek jelentősen befolyásolhatják az idősorra illesztett modell paramétereit.

A fellelhető outlier detektáló algoritmusok az alábbi öt kategóriába sorolhatók:

- **Eloszlás alapú:** Ezek az algoritmusok néhány alapvető eloszlásmodellt tartalmaznak (például normális eloszlás, Poisson eloszlás stb.) és azokat a mintákat jelölik meg outlierként, amelyek az adott eloszlásmodellhez nem illeszkednek [4]. Ezen eljárások legnagyobb hátránya, hogy a mért adatok eloszlása nem mindig ismert. Gyakran sok és hosszadalmas vizsgálatot kell végrehajtanunk ahhoz, hogy megállapítsuk, melyik modell illeszkedik legjobban az adathalmazunkra. (Ezen túlmenően még az sem bizonyos, hogy egyáltalán létezik-e olyan modell, amelyre a mért adatok illeszkednek.)

- **Mélység alapú:** Ezek a geometrián alapuló algoritmusok az adatokat különböző héjakba sorolják, és a

legkülső héjon elhelyezkedő adatokat jelölik meg outlierként [5]. Az algoritmus sebessége a rétegszámmal fordítottan arányos.

- **Csoportosításon alapuló:** Ezen algoritmusok a bemenetként kapott adatokat valamilyen szempont szerint csoportosítják, osztályokba sorolják [6]. Outlierek tekintik azokat az elemeket, amelyek egymagukban alkotnak egyetlen csoportot. Mivel nem az outlierek meghatározása a fő feladatuk, így ezt a problémát nem képesek hatékonyan megoldani.

- **Távolság alapú:** Ezt az algoritmust először Knorr és Ng írta le [7,8]. Ennek alapján azt mondjuk, hogy egy  $P$  adathalmazban lévő  $x$  érték outlier, ha a  $P$ -ben lévő adatok legalább  $b\%$ -a adott távolságnál ( $r$ ) messzebb helyezkedik el a vizsgált  $x$  értéktől. Ez egy egyszerű, két paramétertől ( $r, b$ ) függő eljárás. Problémát az okozhat, ha az adatok jellemzői az adathalmaz különböző részein jelentősen eltérnek egymástól.

- **Sűrűség alapú:** Ezen algoritmust elsőként Breunig adta meg [9]. Ez az eljárás minden értékhez egy helyi outlier tényezőt (Local Outlier Factor, LOF) rendel hozzá, amelyet a szomszédok sűrűsége határoz meg. (Minél magasabb a LOF értéke, az adott pontnak annál kevesebb szomszédja van.) A magas LOF értékkel rendelkező adatokat outlierként jelöli meg. Ezen megoldás hátránya, hogy nagyon érzékeny a szomszédságot meghatározó paraméterekre.

### 3. Az algoritmus

Az általunk kidolgozott algoritmus az úgynevezett L.O.C.I. eljárás [3] alapul. A távolság alapú és a sűrűség alapú algoritmusok előnyeit kombináltuk. Az aktuális mintának csak egy kis szomszédságát vizsgáljuk, így nem okoz problémát, ha az adatok jellemzői jelentősen eltérnek egymástól az adathalmaz különböző részein. Ezen a szomszédságon belül az ebben található pontok statisztikai jellemzői alapján döntjük el, hogy a vizsgált pont outlier-e. Az algoritmusunk két különböző szomszédsággal dolgozik.

A mintavételezési szomszédság vagy más néven  $r$ -szomszédság (1. ábra) egy  $2r$  széles ( $r$  sugarú) intervallum. Az ábrán az aktuálisan vizsgált pontot vastag  $x$ -el jelöltük. A szaggatott vonal a vizsgált ponthoz tartozó mintavételezési tartomány határa. Ezen tartomány minden pontjára definiálunk egy kisebb szomszédságot, ez a számlálási szomszédság. Ezt néhány pontra pontozott vonallal jelöltük az ábrán. A számlálási szomszédság két paraméterrel jellemezhető:  $\alpha, \beta$ . E tartományban vizsgáljuk a szomszédok sűrűségét. Ha az aktuálisan vizsgált pont szomszédsűrűsége jelentősen eltér a mintavételezési tartományban található pontok átlagos szomszédsűrűségétől, akkor a pontot outlierként jelöljük meg. A szomszédsűrűséget alapvetően a számlálási szomszédságban található pontok száma határozza meg. Eddig az algoritmusunknak három fő paramétere van. Az  $r$  paraméter a mintavételezési szomszédság sugara, az  $\alpha$  és  $\beta$  paraméterek pedig a

számlálási szomszédság definiálásához szükségesek. Ezekon kívül még egy paraméterre van szükségünk ( $k$ ), ami a detektálás „szigorúságát” határozza meg. Az 1. táblázat a  $k$  paraméter definiálásához szükséges alapdefiníciókat tartalmazza.

Minden  $p_i, r, \alpha$  és  $\beta$  értékre definiáljuk a szórási tényezőt (DF) az alábbiak szerint:

$$DF(p_i, r, \alpha, \beta) = \frac{\hat{n}(p_i, r, \alpha, \beta) - n(p_i, \alpha, \beta)}{\hat{n}(p_i, r, \alpha, \beta)}$$

Megjegyezzük, hogy  $p_i$   $r$ -szomszédsága mindig tartalmazza  $p_i$ -t. Így a fenti tört nevezője mindig nagyobb nullánál, azaz a fenti kifejezés mindig értelmezett.

Definiáljuk  $\sigma_{DF}(p_i, r, \alpha, \beta)$ -et  $n(p, \alpha, \beta)$  normalizált szórásként a  $p \in N(p_i, r)$ -n:

$$\sigma_{DF}(p_i, r, \alpha, \beta) = \frac{\sigma_{\hat{n}}(p_i, r, \alpha, \beta)}{\hat{n}(p_i, r, \alpha, \beta)}$$

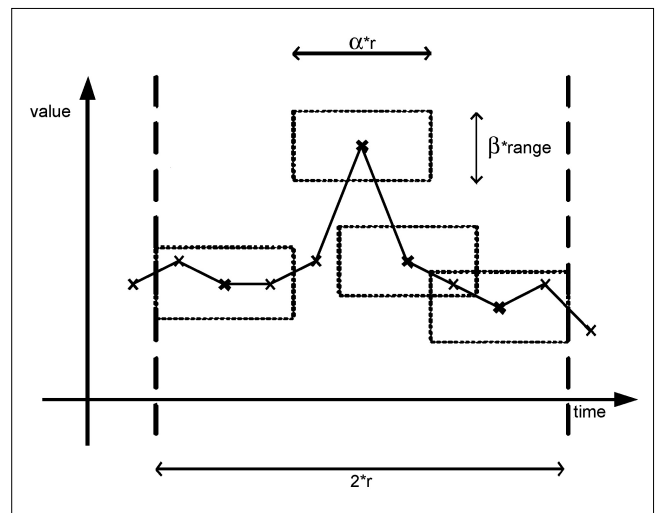
A pontot outlierként jelöljük meg, ha:

$$DF(p_i, r, \alpha, \beta) \geq k \cdot \sigma_{DF}(p_i, r, \alpha, \beta)$$

Tehát, ahogy már fent említettük, a  $k$  paraméter a szűrés „szigorúságát” határozza meg. A  $k$  paraméter értékét három körülinek választva és azt feltételezve, hogy a minták eloszlása normális, annak az esélye, hogy egy minta (amelyik nem outlier) kívül esik a háromszoros szóráson (és így outliernek tekintjük) kevesebb mint 1%. Automatikus paraméterbeállítások esetén  $k$  értéke 2.8. Ha az  $r$  paraméter értéke túl kicsi, nincs elég adatunk a szórást pontos megállapításához. Ha ez az érték túl nagy, az az eltérő helyi jellegekből fakadó hibás detekcióhoz vezethet. Alabeállítások esetén  $r$  értéke a minták számának 10%-a, de maximum 50.

A következő fejezetekben ismertetett eredmények során az  $\alpha$  és  $\beta$  paraméterek értéke megegyezett. Ha ezen paraméterértékek túl kicsik, előfordulhat, hogy helyes adatoknak sem lesz szomszédjuk, míg ha túl nagyok, majdnem minden minta beletartozik a számlálási szomszédságba, így az outlier nem esik kívül a  $k$ -szoros szóráson.

1. ábra Szomszédságok





<b>Jelölések és definíciók</b>	$P, p_i$	Az adathalmaz: $P = (p_1, \dots, p_i, \dots, p_N)$
	$N(p_i, r)$	$p_i$ r-szomszédjainak halmaza: $N(p_i, r) = \{p_x \in P \mid (i-r) \leq x \leq (i+r)\}$
	$n(p_i, r)$	r-szomszédok száma: $n(p_i, r) \equiv  N(p_i, r) $
	$n(p, \alpha, \beta)$	A p pont számlálási szomszédságába tartozó elemek száma
	$\hat{n}(p_i, r, \alpha, \beta)$	$n(p, \alpha, \beta)$ átlaga $p_i$ r-szomszédjain: $\hat{n}(p_i, r, \alpha, \beta) = \frac{\sum_{p \in N(p_i, r)} n(p, \alpha, \beta)}{n(p_i, r)}$
	$\sigma_{DF}(p_i, r, \alpha, \beta)$	$n(p, \alpha, \beta)$ szórása $p_i$ r-szomszédjainak halmazán: $\sigma_{\hat{n}}(p_i, r, \alpha, \beta) = \sqrt{\frac{\sum_{p \in N(p_i, r)} (n(p, \alpha, \beta) - \hat{n}(p_i, r, \alpha, \beta))^2}{n(p_i, r)}}$
	$DF(p_i, r, \alpha, \beta)$	Szórási tényező a pi pontra. Részletesen lásd lentebb.
	$\sigma_{\hat{n}}(p_i, r, \alpha, \beta)$	Normalizált szórás (hogy össze lehessen hasonlítani DF-el). Részletesen lásd lentebb.

1. táblázat

Alapbeállítások estén az a és b értéke 0.1 és 0.05 közé esik, az r paraméter értékétől függően. Az automatikus paraméterbeállítások még nem optimálisak, de már jó eredmények érhetőek el velük (lásd alább). Gyorsan változó idősorok esetén problémák jelentkezhetnek, ennek vizsgálata további terveink között szerepel.

#### 4. Outlier detektáló és eltávolító eszköz (ODRT Tool)

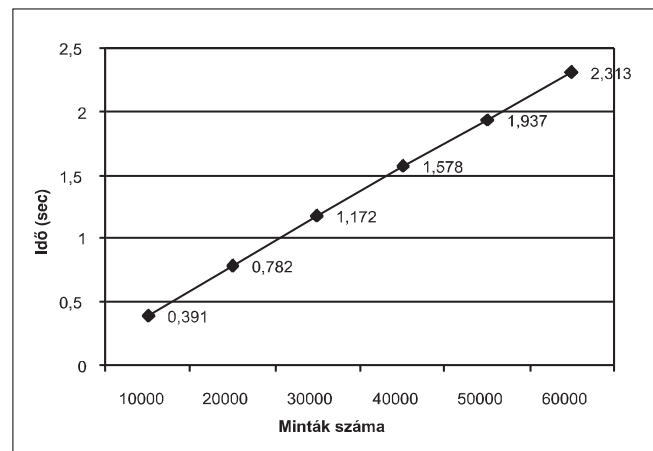
Kifejlesztettünk egy új szoftvert (ODRT, Outlier Detection and Remove Tool), amely gyorsan és hatékonyan képes kezelni nagy adathalmazokat is. Az alkalmazás jelenlegi megvalósítása Microsoft Windows alatt fut. A program automatikusan meg tudja határozni a paramétereket, így felhasználói beavatkozás nélkül is képes adathalmazokat kezelni. Az algoritmus helyes működésének ellenőrzésére számos mérést végeztünk.

A tesztek során az automatikus paraméterbeállításokat használtuk. Számos különböző paraméterű ARLMA folyamatot használtunk kiindulásként, majd ezekhez az idősorokhoz adtunk hozzá additív outliereket. Így meg tudtuk vizsgálni, hogy különböző esetekben mennyire hatékony az algoritmus, azaz a hozzáadott outlierek hány százalékát ismeri fel. Az ellenőrzés során vizsgáltuk az algoritmus sebességét (2. ábra), az outlierek számának hatását a detektálásra, az outlierek eloszlásának hatását a detektálásra, és néhány speciális esetet is (például: negatív outlierok, szinteltolódásos outlierok stb.).

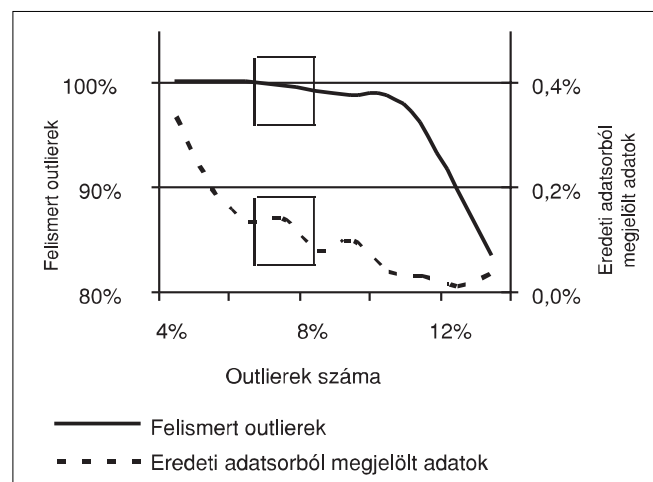
Az algoritmus sebességét automatikus paraméterbeállítások esetén vizsgáltuk. Ha a minták száma több mint 500, a paraméterek fix értékek, így a függvény lineáris. (A mérés során használt hardver: Intel P4 2,4Ghz processzor, ABIT BD7II alaplap, 256 MB RAM).

Amikor az outlierok számának a detektálás jóságára gyakorolt hatását vizsgáltuk, a hozzáadott outlierok száma a minták számának 4-13%-a volt. 200 mérést végeztünk, az eredményt a 3. ábra mutatja.

2. ábra Az algoritmus sebessége



3. ábra Az outlierok számának hatása



A folytonos vonal, és a baloldali ordináta tengely a felismert outliereket jelöli, míg a jobboldali tengely és a szaggatott vonal az eredeti adatsorban outlierként megjelölt értékeket. Az eredmények alapján megállapíthatjuk, hogy az 5-10%-os intervallumban az algoritmus a hozzáadott outlierok közel 100%-át felismeri, míg az eredeti adatsorból kevesebb mint 0.2%-ot jelöl meg outlierként.

### 5. Szolgáltatásminőségi (QoS) paraméterek előrejelzése

A széles körben terjedő valós idejű alkalmazások szükségessé teszik, hogy a szolgáltatók megfelelő szolgáltatás-minőségi paramétereket biztosítsanak a felhasználóknak. Az Internet-szolgáltatók számára fontos, hogy a szolgáltatásminőségi paramétereket valós méréseken alapulva és minél pontosabban tudják előrejelezni, hogy tervezni tudják az erőforrások és az alkalmazások, továbbá a felhasználók összerendelését.

#### 5.1. Előrejelzés az ARIMA modell alkalmazásával

Az ARIMA modell segítséget nyújt olyan problémák megoldásában, mint az idősorok előrejelzése és spektrális analízise. Box és Jenkins 1976-ban átfogó leírását adta az ARIMA modellnek [11], és napjainkban is számos szakirodalom tárgyalja (például [13,14,12]). Azt mondjuk, hogy a  $p$ -edrendű  $x_t$  folyamat autoregresszív, ha felírható a következő alakban:

$$x_t = c_1x_{t-1} + c_2x_{t-2} + \dots + c_px_{t-p} + \varepsilon_t$$

ahol  $\varepsilon_t$  egy  $\sigma$  szórású fehér zaj,  $c_1, \dots, c_p$  konstans.

Ekkor az idősor értékeit a saját korábbi értékeivel, és a véletlen ingadozást reprezentáló  $\varepsilon_t$  diszkrét fehér zajjal fejezzük ki.

A  $q$ -adrendű  $x_t$  folyamat mozgó átlag folyamat, ha felírható következő alakban:

$$x_t = d_1\varepsilon_{t-1} + d_2\varepsilon_{t-2} + \dots + d_q\varepsilon_{t-q} + \varepsilon_t$$

ahol  $d_1, \dots, d_q$  konstans.

A kombinált ARMA-modell autoregresszív és mozgóátlagolás tagokat is tartalmaznak, tehát a következő alakban írható fel:

$$x_t = c_1x_{t-1} + c_2x_{t-2} \dots c_px_{t-p} + d_1\varepsilon_{t-1} + d_2\varepsilon_{t-2} \dots d_q\varepsilon_{t-q} + \varepsilon_t$$

Az idősor gyakran magában hordozza a környezet változásának hatását. Ebben az esetben a megfigyelt értékek közti különbség lesz stacionárius, tehát az eddig leírt modellekkel jellemezhető.

Az ARIMA folyamatok az ARMA folyamatoknál általánosabb modellt írnak le. Ha egy ARIMA( $p,d,q$ ) folya-

matról beszélünk, akkor az azt jelenti, hogy az idősort  $d$ -szer differenciálva egy ARMA( $p,q$ ) folyamatot kapunk. (Az integrált modellek esetében tapasztalataink szerint a gyakorlatban általában elegendő egyszer differenciálni a sort ahhoz, hogy stacionárius modell legyen illeszthető rá.) Az autoregresszív paraméterértékeket a Yule–Walker egyenletrendszer [15] alapján számítottuk ki. A modellt az S-PLUS 2000 matematikai szoftver segítségével határoztuk meg [16].

#### 5.2. QoS paraméterek előrejelzése az outlierok előzetes eltávolításával

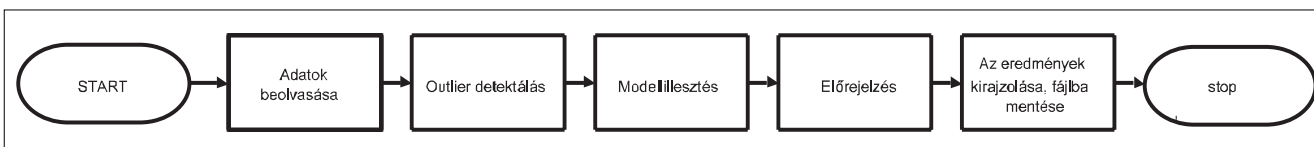
Az előrejelzésnek figyelembe kell vennie, hogy a legtöbb valós adatsor tartalmaz outliereket, amelyek a helyi véletlenszerű események hatására bekövetkező extrém ingadozások. Az outlierok detektálásával kiküszöbölhetjük ezen események hatását az előrejelzésre. A két eljárás, azaz az előrejelzés és az outlier detektálás együttműködését az 2. ábra szemlélteti.

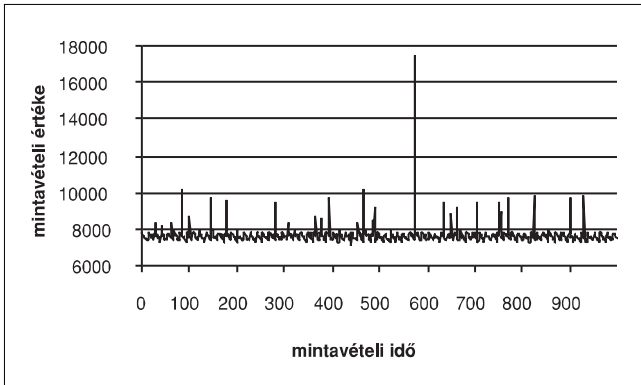
A következőkben az outlierok előzetes eltávolításának hatására az előrejelzésben bekövetkezett javulást tanulmányozzuk. Vizsgálatainkat valós, mért adatsorokon végeztük. Az adatok egy része a Salzburg Research-től származik. A mérések során a Salzburg Research hálózata és a szolgáltató (Telekom Austria) közötti végpont-végpont összeköttetések monitorozták az aggregált forgalom késleltetését. A késleltetés-mérés a GPS óraszinkronizáción alapult. A küldő és a fogadó fél közötti forgalom két útvonalválasztón haladt keresztül. Ez a hálózati kialakítás a mai szolgáltatók és felhasználók között tipikusnak tekinthető [18]. A kutatóintézet több százezer adatból álló idősort bocsátott a rendelkezésünkre, ebből használtunk fel részleteket a teljesítményelemzéshez. Az adatok másik része saját méréseken alapul, melyet a tanszék két gépe között végeztünk. Az előrejelzés és a modellillesztés során automatikus paraméterbeállításokat használtunk.

A teljesítményelemzés során 1000 adatból álló idősorokat vizsgáltunk. Az első 95% alapján jeleztük előre a következő 50 adatot. Az egyik előrejelzés során az outliereket előzetesen eltávolítottuk, a másik esetben ezt nem tettük meg. A modell rendjének meghatározásakor a rendre viszonylag magas (25-35) értékeket kaptunk. A mérések kiértékeléséhez az eredeti adatsort, az outlierok eltávolítása után és az előzetes outlier detekció nélkül kapott eredményeket egyetlen grafikonon tüntettük fel.

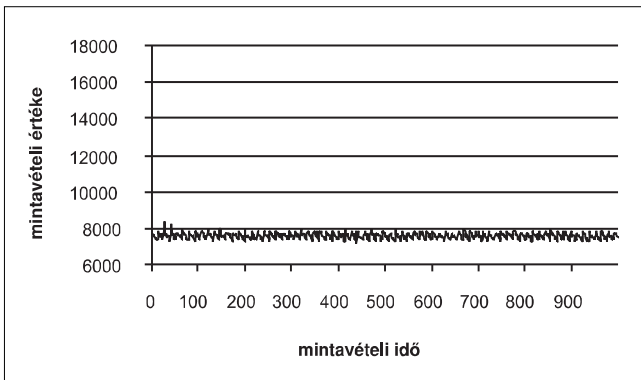
Először egy olyan esetet mutatunk be, amikor a vizsgált szakasz kihasználtsága valószínűleg alacsony, ezért a késleltetés kis tartományon belül ingadozik, de az adatsorban néha találhatók outlierok. Az eredeti és a szűrt idősort szemlélteti a 3/a. és 3/b. ábra.

2. ábra A két eljárás együttműködése





3/a. ábra Az eredeti adatsor

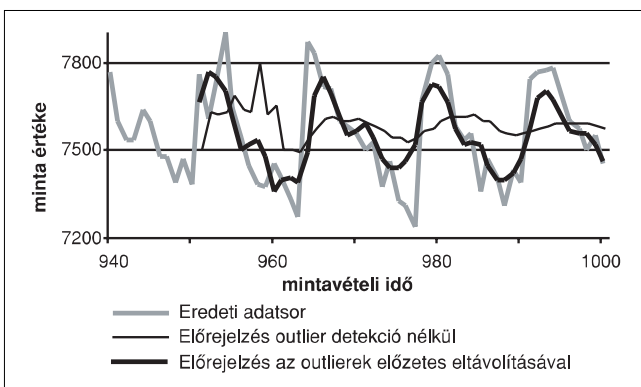


3/b. ábra A szűrt adatsor

A 4. ábrán láthatjuk az eredeti adatsort (vastag szürke vonal), az outlier detektálás nélküli előrejelzést (vékony fekete vonal) és az outlierek előzetes eltávolításával kapott előrejelzést (vastag fekete vonal). Jól látható, hogy a becült adatok az előzetes outlier detekció alkalmazásával még 50 adat után is jól követik az eredeti értékeket. Ha nem távolítjuk el az outliereket, a jóslás nem képes követni az eredeti adatsor ingadozását, ellaposodik, egy idő után konstans értéket vesz fel.

A második esetben a késleltetés értékei nagyobb intervallumban ingadoznak. Az 5. ábrán láthatjuk az eredeti adatsort (vastag szürke vonal), az outlier detektálás nélküli előrejelzést (vékony fekete vonal) és az outlierek előzetes eltávolításával kapott előrejelzést (vastag fekete vonal). Az ábra azt mutatja, hogy előze-

4. ábra Az outlier detekció hatása (1)



5. ábra Az outlier detekció hatása (2)

tes outlier detekció esetén az előrejelzés jól követi az eredeti adatsor főbb mozgásait, míg outlier eltávolítás nélkül a predikció ellaposodik, képtelen követni az adatsor változásait.

## 6. Összegzés és további teendők

A felhasználói igények kielégítéséhez valamint a hálózat kihasználtságának növeléséhez szükséges a hálózati forgalom és a szolgáltatásminőségi paraméterek minél pontosabb előrejelzése. Az előrejelzés pontosságát azonban nagymértékben rontják a mért adatsorokban található kiugró értékek, az outlierok. Egy olyan új eszközt mutattunk be, amely képes detektálni és eltávolítani az outliereket nagy méretű adatsorokból is gyorsan és hatékonyan. A modellezéshez és előrejelzéshez egy ARIMA folyamatokon alapuló módszert adtunk. Az eredmények alapján látható, hogy az előzetes outlier detekció hatására jelentősen javul az előrejelzés, ezért ez a módszer sokat segíthet a garantált szolgáltatásminőség biztosításában.

Miután eltávolítottuk az outliereket, a helyettesítési értéket a szomszédos adatok átlaga alapján határoztuk meg. Valószínűleg tovább javítana az eredményeken, ha a helyettesítési értéket valamilyen módon a megelőző értékek alapján előrejeleznénk (például lineáris predikcióval). Ennek vizsgálata további terveink között szerepel.

## Irodalom

- [1] V. Gómez and A. Maravall, „Programs TRAMO (Time series Regression with Arima noise, Missing observations, and Outliers) and SEATS (Signal Extraction in Arima Time Series). Instructions for the User”, Working Paper 9628, Servicio de Estudios, Banco de España, 1996.
- [2] A. Maravall and G. Caporello, „A tool for Quality control of time series data, Program TERROR”, Bank of Spain, Proc., Challenges to Central Bank Statistical Activities Conf., Irving Fisher Committee (ISI) and Bank for International Settlements, Basel, August 2002

- [3] S. Papadimitriou, H. Kitawaga, P. B. Gibbons, C. Faloutsos, „Fast Outlier Detection Using the Local Correlation Integral”,  
[http://www.intel-research.net/Publications/Pittsburgh/081620021325\\_99.pdf](http://www.intel-research.net/Publications/Pittsburgh/081620021325_99.pdf)
- [4] P. J. Rousseeuw and A. M. Leroy, „Robust Regression and Outlier Detection”, John Wiley and Sons, 1987.
- [5] T. Johnson, I. Kwok, and R. T. Ng, „Fast computation of 2-dimensional depth contours”
- [6] A. K. Jain, M. N. Murty, and P. J. Flynn, „Data clustering: A review”, ACM Computing Surveys, 31(3):264–323, 1999.
- [7] E. M. Knorr and R. T. Ng, „Algorithms for mining distance-based outliers in large datasets”, 1998;  
<http://www.courses.cs.uiuc.edu/~cs497jh/papers/knorrngvldb98.pdf>
- [8] E. M. Knorr, R. T. Ng, and V. Tucakov, „Distance-based outliers: Algorithms and applications”,  
[http://www.cs.ubc.ca/nest/dbsl/public/vldb\\_journal\\_feb2000.ps](http://www.cs.ubc.ca/nest/dbsl/public/vldb_journal_feb2000.ps)
- [9] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, „Lof: Identifying density-based local outliers”, 2000;  
<http://www.cs.ualberta.ca/~joerg/papers/LOF-final.pdf>
- [10] D. Hawkins, „Identification of outliers”, Chapman & Hall, London 1980.
- [11] G. Box, G. Jenkins, G. Reinsel, „Time Series Analysis: Forecasting and Control”, revised ed., Prentice Hall, 3rd Edition, 1994.
- [12] P. J. Bockwell, R. A. Davis, „Introduction to Time Series and Forecasting”, Springer Verlag, 2002.
- [13] Michelberger Pál, Szeidl László, Várlaki Péter, „Alkalmazott folyamatstatisztika és idősor analízis”, Typorex kiadó, 2001.
- [14] Vágó Zsuzsanna, „Idősorok sztochasztikus modelljei”, BKE oktatási segédlet, 1995.
- [15] P.J. Bockwell, R. A. Davis: Introduction to Time Series and Forecasting, Springer Verlag, 2002.
- [16] S-PLUS 2000, „Guide to Statistic”, Data Analysis Product Division Math Soft, 1999.
- [17] W. N. Venables, B. D. Ripley, „Modern Applied Statistics with S-PLUS”, Springer Verlag, 1998.
- [18] I. Milouceva, E. Müller and A. Anzalonli, „A practical approach to forecast QoS considering outliers”, Proc., Inter-Domain Performance and Simulation Workshop, pp.163–172, Salzburg, Austria, February 2003.

## Hírek

Az egyre növekvő **rádiófrekvenciás azonosítás (RFID)** számos új megoldást nyújtott a piac számára. A Sun, a SIS Technologies és az SSA Global termékeit és szolgáltatásait egyesítő Sun RFID-kezdményezéseit az EPCglobal konferencián is bemutatta.

A **Sun Java System RFID szoftver** (amely jelenleg Solaris operációs rendszeren érhető el, de a tervek szerint 2004 novemberében Linuxon is megjelenik) korszerűsíti az RFID-hardverek – a címkék és az olvasók – kezelését. A szoftver a Sun RFID rendszerstratégiájának fontos elemét jelenti, de ide tartoznak még a globális kliensszolgáltatások, a világ RFID tesztközpontjai, a Sun hardverek és tárolóeszközök, valamint az RFID-terület megoldási partnerei. A Sun Java RFID szoftverrel kapcsolatban további információk a [www.sun.com/rfid](http://www.sun.com/rfid) oldalon olvashatók.

A rádiófrekvenciás termékazonosítás (RFID) és az egyéb távérzékelő technológiák (SBS) kétségkívül versenyelőnyt és üzleti hasznot jelenthetnek, ha a vállalat információs infrastruktúrájának és üzleti folyamatainak szerves részévé válnak. Világszerte már számos gyártó döntött az RFID rendszerek bevezetése mellett. Sok vállalat előírta, hogy beszállítóik 2005-ig RFID jelzőcímkékkel lássák el raklapos szállítmányukat. Az IDC véleménye szerint a vállalatoknak nem szabad csupán azért bevezetni az új technológiát, mert a legnagyobb vevők ezt megkövetelik tőlük, hanem először meg kell érteniük az RFID-ban rejlő technológiai és hatékonyságnövelő képességeket, és ezek függvényében kell beépíteniük az új eszközt üzleti stratégiájukba.

A Meta Group piackutató cég szerint 2008-ra az iparcikkek 30%-át már RFID azonosítóval látják el, és ez az arány 2013-ra 80%-ra nőhet. Az amerikai Yankee Group is bízik a technológia hosszú távú sikerében: a vállalat szerint 2008-ra az RFID piaci értéke elérheti a 2,35 milliárd dollárt.

# Latin és bűvös négyzetek a játékos alkalmazásoktól a biztonságig

DÉNES TAMÁS *matematikus*

*tdenest@freemail.hu*

**Kulcsszavak:** négyzetek szerkesztése, ortogonális rendszer, összegezési szabályok

Nem egyedülálló a matematika történetében, hogy egyes fejezetei a szórakozás, a játék területén fogantak és hosszabb-rövidebb fejlődés után a matematika új fejezeteivé váltak. Ezt az utat járta be a kombinatorika egy alig 300 éves fejezete, a latin négyzetek elmélete. Különössége mégsem abban áll, hogy fejlődésének és főleg alkalmazásainak jelentős része a XX. század gyümölcse, hanem abban, hogy a klasszikus numerikus gondolkodást felcserélte a struktúrák belső összefüggéseinek elemzésével és igen szemléletes ábrázolásával. A cikk a latin négyzetek szerzteágazó, klasszikus és egészen modern alkalmazásainak vázlatos bemutatását tűzte ki célul, a szórakoztató matematikától, a XXI. század információs társadalmában kulcs jelentőségű adatátvitelen keresztül, a kriptográfiáig.

## Alapfogalmak és definíciók történeti illusztrációkba ágyazva

Mivel a latin négyzetek elmélete egyelőre nem képezi matematika oktatásunk törzsanyagát, így a cikk megértéséhez az olvasónak néhány alapfogalom megismérésére lesz szüksége. Ezeket az alapvető fogalmakat és összefüggéseket, valamint a keletkezésük történetét ismertetem a következőkben.

Egy  $n$ -ed rendű latin négyzeten egy olyan  $n \times n$  méretű négyzetes mátrixot értünk, amelynek soraiban és oszlopaiban az  $a_1, a_2, \dots, a_n$  elemek mindegyike egyszer és csak egyszer szerepel. Általában az  $a_1, a_2, \dots, a_n$  elemek az  $1, 2, \dots, n$  természetes számok. Az 1/a. és 1/b. ábra egy-egy 4-ed rendű latin négyzetre mutat példát.

1/a. ábra	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>2</td><td>1</td><td>4</td><td>3</td></tr><tr><td>3</td><td>4</td><td>1</td><td>2</td></tr><tr><td>4</td><td>3</td><td>2</td><td>1</td></tr></table>	1	2	3	4	2	1	4	3	3	4	1	2	4	3	2	1	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>2</td><td>3</td><td>4</td><td>1</td></tr><tr><td>3</td><td>4</td><td>1</td><td>2</td></tr><tr><td>4</td><td>1</td><td>2</td><td>1</td></tr></table>	1	2	3	4	2	3	4	1	3	4	1	2	4	1	2	1	1/b. ábra
1	2	3	4																																
2	1	4	3																																
3	4	1	2																																
4	3	2	1																																
1	2	3	4																																
2	3	4	1																																
3	4	1	2																																
4	1	2	1																																

A definícióból világosan kiderül, hogy jelen esetben nem az  $a_1, a_2, \dots, a_n$  elemek számértéke számít, csupán csak különbözőségük, valamint a mátrixban elfoglalt helyük (struktúrájuk).

Egy latin négyzetet ciklikusnak nevezünk, ha egymás alatti soraiban az elemek sorrendje azonos, csak egy hellyel jobbra (vagy balra) vannak az elemek eltolva (lásd 1/b. ábra). Egy  $n$ -ed rendű latin négyzet egy tranzverzálisán értjük  $n$  darab olyan elemét, amelyek mindegyike különböző sorában, illetve oszlopában helyezkedik el és nincs köztük két azonos. Az 1/a. ábrán látható latin négyzetben a satírozott négy elem például egy tranzverzális alkot. Két  $n$ -ed rendű latin négyzetet akkor nevezünk ortogonálisnak, ha egymásra helyezve őket, az egymás felett levő elemekből alkotott

párok mind különbözőek. Példaképpen bemutatjuk az 1/a. ábrán szereplő latin négyzet egy ortogonális párját (2. ábra), majd a két latin négyzet egymásra helyezésével nyert számpárokat. (A 3. ábra segítségével könnyen meggyőződhetünk arról, hogy a 16 számpár mind különböző.)

2. ábra	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>4</td><td>3</td><td>2</td><td>1</td></tr><tr><td>2</td><td>1</td><td>4</td><td>3</td></tr><tr><td>3</td><td>4</td><td>1</td><td>2</td></tr></table>	1	2	3	4	4	3	2	1	2	1	4	3	3	4	1	2	<table border="1"><tr><td>1,1</td><td>2,2</td><td>3,3</td><td>4,4</td></tr><tr><td>2,4</td><td>1,3</td><td>4,2</td><td>3,1</td></tr><tr><td>3,2</td><td>4,1</td><td>1,4</td><td>2,3</td></tr><tr><td>4,3</td><td>3,4</td><td>2,1</td><td>1,2</td></tr></table>	1,1	2,2	3,3	4,4	2,4	1,3	4,2	3,1	3,2	4,1	1,4	2,3	4,3	3,4	2,1	1,2	3. ábra
1	2	3	4																																
4	3	2	1																																
2	1	4	3																																
3	4	1	2																																
1,1	2,2	3,3	4,4																																
2,4	1,3	4,2	3,1																																
3,2	4,1	1,4	2,3																																
4,3	3,4	2,1	1,2																																

Az ortogonális latin négyzet párok létezése, mint látni fogjuk, szoros kapcsolatban van a tranzverzálisokkal. Erre vonatkozó alapvető eredmény Dulmage-Mendelshon tétele:

Két  $n$ -ed rendű latin négyzet ortogonalitásának szükséges és elégséges feltétele, hogy a diszjunkt diagonálisai száma pontosan  $n$  legyen.

$L_1, L_2, \dots, L_k$   $n$ -ed rendű latin négyzetek egy ortogonális rendszert alkotnak, ha bármely két különböző latin négyzetet véve a  $k$  darab közül, azok ortogonális párt képeznek. Bebizonyítható, hogy  $n \times n$ -es latin négyzetekből legfeljebb  $n-1$  olyan létezhet, amelyek közül bármely kettő ortogonális, ha viszont ezek mind léteznek, akkor az ortogonális latin négyzetek teljes rendszeréről beszélünk. A XX. század elején kiderült, hogy számos súlyos kombinatorikai probléma mélyén az ilyen rendszerek létezésének kérdése rejlik.

Egy  $L$  latin négyzetet akkor nevezünk vízszintesen teljesnek, ha bármely a latin négyzetben szereplő  $a, b$  ( $a \neq b$ ) elempárra van olyan sora  $L$ -nek, amelyben az  $a$  elemet  $b$  követi. (Ha a leírt tulajdonság oszlop irányban teljesül, akkor az  $L$  latin négyzetet függőlegesen teljesnek nevezzük.) Egy latin négyzetet, amely vízszintesen és függőlegesen is teljes, teljes latin négyzetnek nevezzük.

A 4. ábrán látható latin négyzet teljes (erről győződhet meg az olvasó, ha a kívánt tulajdonságot megvizsgálja az összes lehetséges (1,2), (1,3), (1,4), (2,3), (2,4), (3,4) elempárookra).

4	1	3	2
3	4	2	1
1	2	4	3
2	3	1	4

4. ábra

### Kártyalapok és a 36 tisztt problémája

Leonhard Euler (1707–1783) XVIII. századi matematikus a latin négyzetek névadója, mivel ő alkalmazott a négyzetes mátrixbeli elemek jelölésére latin betűket, az addig szokásos számok helyett. Ez az algebrai struktúrák területén hasonló jelentőséggel bírt, mint F. Viéte (1540–1603) kétszáz évvel korábbi tette, az algebrai egyenletek szimbólumainak bevezetésével. L. Eulert szokták említeni, mint aki bevezette az ortogonális latin négyzet párok fogalmát is. Azonban már Eulert megelőzően is ismerték a neki tulajdonított két fogalmat. A történelmi hűség kedvéért felhívom az olvasó figyelmét Claude-Gaspar Bachet de Méziriac-ra (1581–1638) és M. Ozanam-ra (1640–1712), akik a játékkártyával kapcsolatosan már Euler előtt is eljutottak a latin négyzetek, illetve az ezekből alkotott ortogonális párok fogalmához [1,9,10].

Az 5. ábrán látható negyed rendű ortogonális latin négyzet pár [1]-ből való, amely tulajdonképpen a következő feladatot oldja meg: *hogyan lehet a francia kártya négy színű (kőr, treff, káró, pikk) négy figurájából (ász, király, dáma, bubi) 16 lapot úgy kiválasztani és egy 4x4 méretű mátrixban elrendezni, hogy minden szín minden figurával előforduljon és minden sorban, illetve oszlopban minden szín és minden figura pontosan egyszer forduljon elő.*

Ász kőr	Király treff	Dáma káró	Bubi pikk
Bubi káró	Dáma pikk	Király kőr	Ász treff
Király pikk	Ász káró	Bubi treff	Dáma kőr
Dáma treff	Bubi kőr	Ász pikk	Király káró

5. ábra

Később 1776-ban majd 1779-ben Euler a Szent Pétervári Akadémián tartott előadásában már megmutatta, hogy ha  $n$  4-gyel osztható természetes szám, akkor van  $n$ -ed rendű latin négyzetekből álló ortogonális pár.

Ugyancsak ekkor vetette fel az azóta 36 tisztt problémájaként ismert feladatot: *Válasszunk ki 36 tiszttet úgy, hogy közöttük hat különböző rendfokozatú szerepeljen és a tisztek hat különböző csapattestből kerülje-*

*nek kiválasztásra, minden egyes csapattestből hat különböző rendfokozatú tiszt szerepeljen a 36 között. Fel lehet-e a fentiek szerint kiválasztott tiszteket úgy állítani egy 6x6-os alakzatba, hogy minden egyes sorban illetve oszlopban minden rendfokozat illetve csapattest pontosan egyszer szerepeljen. A kérdés röviden úgy is feltehető, hogy létezik-e két olyan hatod rendű latin négyzet, amely egymásra ortogonális?*

A feladat kísértetiesen hasonlít az előzőekben a francia és magyar kártyákkal megoldottakhoz, mégis Euler azt sejtette, hogy a 36 tiszt problémájának nincs megoldása. Sőt ennél általánosabban azt is sejtette, hogy általában ha  $n=4k+2$  alakú, akkor nincsen ortogonális  $n$ -ed rendű latin négyzet pár. Ez utóbbi sejtést évszázadokon át Euler sejtésnek hívták és több mint 200 évig foglalkoztatta a matematikusokat a sejtés bizonyítása, vagy cáfolása.

### Az Euler sejtés megcáfolása és néhány megoldatlan probléma

Már Euler tudta, hogy sejtése az  $n=2$  esetre igaz (a bizonyítás igen egyszerű, hiszen az 1, 2 számokból mindössze két különböző 2x2-es latin négyzet készíthető). Az  $n=6$  esetre, azaz a 36 tiszt problémájára azonban a bizonyítással 1900-ig kellett várni, míg azt G. Tarry éppen a XX. század fordulóján bebizonyította. Euler általános sejtése azonban nem bizonyult igaznak. Csaknem kétszáz évvel a sejtés megfogalmazása után, 1959-ben R. C. Bose, S. S. Shrikhande és E. T. Parker bebizonyították, hogy az Euler sejtés  $n \geq 10$  esetén nem igaz, azaz minden  $n=4k+2$  alakú számra, ha  $k$  érteke legalább 2, léteznek ortogonális latin négyzetek.

A 6. ábra szemlélteti azt a híres tizedrendű ortogonális latin négyzet párt, amelyet 1959-ben hoztak nyilvánosságra, megdöntve az Euler sejtést az  $n=10$  esetre.

6. ábra

0	4	1	7	2	9	8	3	6	5
8	1	5	2	7	3	9	4	0	6
9	8	2	6	3	7	4	5	1	0
5	9	8	3	0	4	7	6	2	1
7	6	9	8	4	1	5	0	3	2
6	7	0	9	8	5	2	1	4	3
3	0	7	1	9	8	6	2	5	4
1	2	3	4	5	6	0	7	8	9
2	3	4	5	6	0	1	8	9	7
4	5	6	0	1	2	3	9	7	8

0	7	8	6	9	3	5	4	1	2
6	1	7	8	0	9	4	5	2	3
5	0	2	7	8	1	9	6	3	4
9	6	1	3	7	8	2	0	4	5
3	9	0	2	4	7	8	1	5	6
8	4	9	1	3	5	7	2	6	0
7	8	5	9	2	4	6	3	0	1
4	5	6	0	1	2	3	7	8	9
1	2	3	4	5	6	0	9	7	8
2	3	4	5	6	0	1	8	9	7

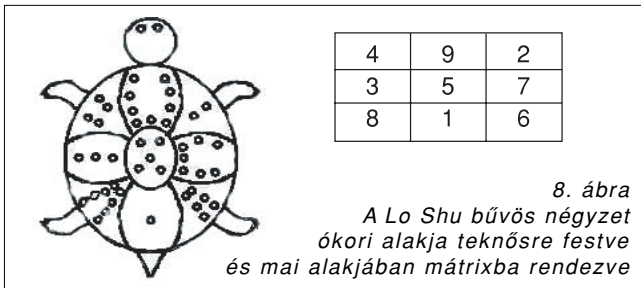
Nem sikerült azonban teljes  $n$ -ed rendű latin négyzetből álló ortogonális rendszert találni, ha  $n \neq p^r$  ( $n$  nem prímszám hatvány). Azonban R. H. Bruck és H. J. Ryser bebizonyították a következő tételt:  $n-1$  darab  $n$ -ed rendű latin négyzetből álló ortogonális rendszer nem létezik, ha  $n \equiv 1, 2 \pmod{4}$  (azaz  $n$  négygyel osztva 1, vagy 2 maradékot ad), hacsak nem  $n=a^2+b^2$  ( $n$  két négyzetszám összegeként állítható elő). Itt érdemes megjegyezni, hogy *Fermat híres karácsonyi tétele*, melyet 1640. karácsonyán fogalmazott meg, így szól: *Minden  $n=4k+1$  alakú prímszám felírható két egész szám négyzetének összegeként.* E két tétel összevetéséből tehát az következik, hogy ha  $n$  egy  $4k+1$  alakú prímszám, akkor létezik  $n-1$  darab (azaz  $4k$  darab) latin négyzetből álló (azaz teljes) ortogonális rendszer.

### A latin és bűvös négyzetek kapcsolata

$n$ -ed rendű *bűvös négyzetnek* nevezünk egy olyan négyzetes mátrixbeli elrendezést, amelyben  $n^2$  egész szám szerepel (általában, de nem szükségszerűen  $0, 1, \dots, n^2$ ) és a négyzetes mátrix minden sorában, oszlopában, illetve két fő átlójában az elemek összege azonos.

#### Lo Shu bűvös négyzet

Talán nem véletlen, hogy a bűvös négyzet, akár csak a számmissztika, már jóval korábban, az ókori Kínában felbukkant. A Lo Shu négyzet az ókori Kínából származó bűvös négyzet, melyet egy óriásteknős páncéljára festettek és a Feng Sui fontos részét képezi (7. ábra).

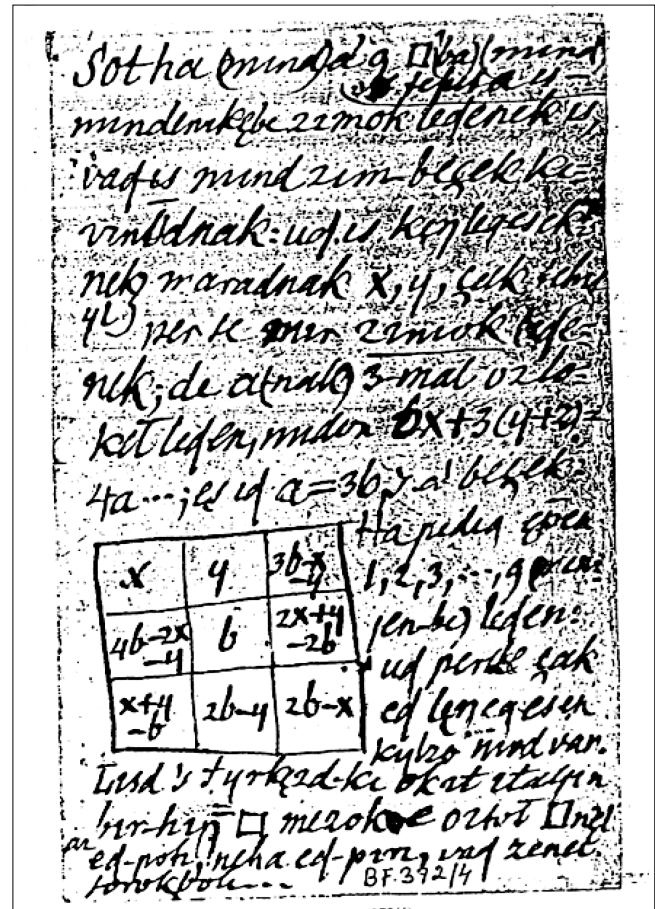


8. ábra  
A Lo Shu bűvös négyzet ókori alakja teknősrre festve és mai alakjában mátrixba rendezve

#### Bolyai János bűvös négyzete

Több évtizedes kutatás után 1999-ben jelent meg Kiss Elemér marosvásárhelyi matematika professzor kötete [8], melyben Bolyai Jánosról egy egészen új képet tárt elénk. Bolyai János kéziratot hagyatékának szisztematikus áttanulmányozása arra a meglepő eredményre vezetett, hogy Bolyai közismert geometriáján kívül, nagyrészt a matematika egészen más területeivel foglalkozott. Ezen eddig ismeretlen eredményei közül való, a 8. ábrán látható kéziratröredék, amely éppen a  $3 \times 3$ -as bűvös négyzetek általános leírásával foglalkozik. Mivel Kiss Elemér professzor alapos kutatásai ellenére sem találta meg ezen kézirat többi oldalát, e helyen szeretném rekonstruálni annak néhány lehetséges összefüggését.

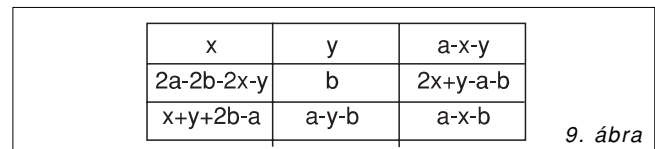
Mint látni fogjuk, a végeredmény meglehetősen összefüggést tár fel a fentiekben bemutatott Lo Shu bűvös négyzettel.



8. ábra  
Bolyai János  $3 \times 3$ -as bűvös négyzetekről szóló kéziratröredéke

Az ábra kéziratában Bolyai  $a$ -val jelölte a  $3 \times 3$ -as bűvös négyzet sor, oszlop és átló összegeit, a középső cella elemét  $b$ -vel, az első két cella elemeit pedig  $x$ -szel és  $y$ -nal jelölte.

Így a 9. ábrán látható kitöltést kapjuk:



9. ábra

Írjuk fel az ábra mátrixának mellékátlójára adódó egyenletet:

$$a-x-y+b+x+y+2b-a = a, \tag{1}$$

amelyből azt kapjuk, hogy  $a=3b$ ,

ami éppen Bolyai kéziratának közepén található összefüggés. Ha pedig a cellákba a bűvös négyzet képzési szabályai szerint az  $1, 2, 3, \dots, 9$  számtani sorozatot írjuk, akkor fennáll:

$$\frac{(1+9)9}{2} = 3a \Rightarrow a=15 \Rightarrow b=5 \tag{2}$$

Ekkor a 9. ábrába helyettesítve kapjuk:

10. ábra	x	y	15-x-y
	20-2x-y	5	2x+y-10
	x+y-5	10-y	10-x

(2)-ből tudjuk, hogy minden sor, oszlop és átló elemeinek az összege 15. Mivel az 5-ös szám közepén van, így minden irányban a két mellette elhelyezkedő elem a 10-nek egy-egy partíciója. Ezek: 1+9, 2+8, 3+7, 4+6.

Mivel a cellákban csupa különböző szám áll, így a vízszintes, a függőleges és átlós irányú szomszédok éppen a 4 db partíciót alkotják. A négyből kettő szabadon választható, a másik kettő ezekből már adódik. Az összes megoldások száma tehát:

$$\binom{4}{2} = 6 \tag{3}$$

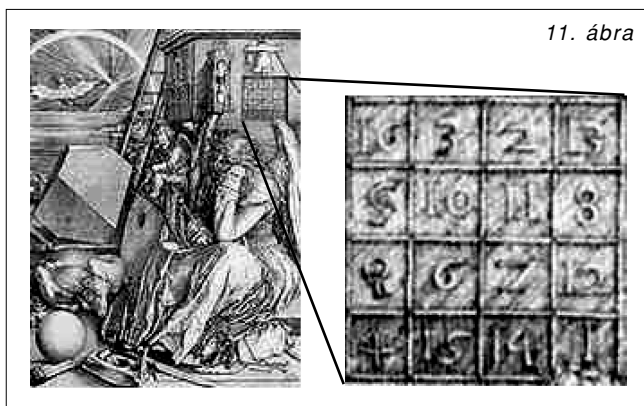
A megoldások a 10. ábrán szereplő mátrix elemeire felírt összefüggésekből levezethetők.

Bár nem tudjuk, hogy Bolyai a jegyzetei többi oldalán mit írhatott, de a 10. ábra rajza melletti megjegyzése pontos, miszerint ha ezt a számtani sorozatot alkalmazzuk a bűvös négyzet kitöltésére, akkor csupán egyetlen megoldás van. A hat mátrix a középső elem körüli elforgatásokkal egymásba átvihető.

A Lo Shu bűvös négyzet pontosan megegyezik a Bolyai-féle hatodik megoldással. Vajon Bolyai János tudott erről?

### Albrecht Dürer bűvös négyzete

Egy-egy bűvös négyzet összeállítása a matematika történetének hajnalán még nehéz feladatnak számított, ezért nem csodálkozhatunk, hogy azoknak valamilyen mágikus erőt tulajdonítottak. Az utóbbi századokban sokat foglalkoztak a bűvös négyzetek matematikai tulajdonságaival és kutatták készítésük módszereit.



Albrecht Dürernek, a reneszánsz kor nagy német festőjének a figyelmét is egy „többszörösen” bűvös négyzet ragadta meg, ezt láthatjuk Melankolia című rézmetszetének háttérében (11. ábra).

Ez a bűvös négyzet már Dürer korában több mint ezeréves múltra tekinthetett vissza, valószínűleg Indiából került át Európába. A művész a sorokat kissé átren-

dezte, hogy az alsó sor közepére a 15 és a 14 számok kerüljenek, jelezve műve készítésének évét, 1514-et.

Dürer bűvös négyzete teljesen helytálló: mezőin az 1-től 16-ig terjedő egész számok helyezkednek el, sorokban, oszlopaiban és átlóiban a számok összege mindenütt 34, másrészt az említett dátumot adó két számon kívül több egyéb különlegessége is van. Alsó és felső sorában a számok négyzeteinek összege is egyenlő, és ugyanez áll a két szélső oszlop számaira is. A négyzetet függőleges és vízszintes középvonala négy darab 2x2-es négyzetre vágja szét, ezek mindegyikében ugyancsak 34 a számok összege, de ugyanannyi a közepén elhelyezkedő 2x2-es négyzetben is. A négyzet négy csúcsánál levő számok összege is 34, s ugyanez áll azoknak a 3x3-as négyzeteknek a sarokszámaira is, amelyeket az eredetiből egy szélső sor és oszlop elhagyásával nyerünk.

### Bűvös négyzet konstrukciók latin négyzetekből

A játékos elme bravúros teljesítménye a sakktablára írt lóugrásos bűvös négyzet. Ezen ha a huszár elindul az 1-es számot tartalmazó mezőről, bejárhatja a sakktablát úgy, hogy mindegyik ugrása a következő számot tartalmazó mezőre vezet, ráadásul a 64. ugrás visszajuttathatja a kiindulási helyére. Bemutatunk egy ilyen sakktablát (12. ábra), melynek minden sorában és oszlopában 260 a számok összege (az átlókra ez itt nem teljesül).

50	11	24	63	14	37	26	35
23	62	51	12	25	34	15	38
10	49	64	21	40	13	36	27
61	22	9	52	33	28	39	16
48	7	60	1	20	41	54	29
59	4	45	8	53	32	17	42
6	47	2	57	44	19	30	55
3	58	5	46	31	56	43	18

12. ábra Lóugrásos bűvös négyzet

A bűvös négyzetek megalkotásának ezen kívül még számos módját dolgozták ki. A játékos alkalmazásokon jóval túlmutató jelentősége van azonban a bűvös négyzetek és a latin négyzetek kapcsolatának. A 13/a. ábrán szereplő két latin négyzetről [5]-ben mutatta meg a szerző, hogy összegük egy bűvös négyzet, amit a 13/b. ábrán mutatunk be.

E speciális eset általánosítható a következő módon:  
 Ha  $n=2k+1$  alakú, akkor létezik két  $n$ -ed rendű latin négyzet (jelöljük ezeket  $L(a_{ij})$  és  $L(b_{ij})$ -vel), melyek összege az  $M(c_{ij})$   $n \times n$ -es mátrix, melynek minden sorában, oszlopában és két főátlójában az elemek összege  $\frac{(n^2+1)n}{2}$  és az  $M(c_{ij})$  mátrix elemei az  $1, 2, \dots, n^2$  számtani sorozat elemei.



5	3	1	4	2
4	2	5	3	1
3	1	4	2	5
2	5	3	1	4
1	4	2	5	3

5	15	0	10	20
0	10	20	5	15
20	5	15	0	10
15	0	10	20	5
10	20	5	15	0

13/a. ábra

10	18	1	14	22
4	12	25	8	16
23	6	19	2	15
17	5	13	21	9
11	24	7	20	3

13/b. ábra

Bizonyításként bemutatjuk a két latin négyzet szerkesztési elvét.

Mivel  $L(a_{ij})$  egy latin négyzet, így definíció szerint minden sorában és oszlopában az  $1, 2, \dots, n$  egész számok egy-egy permutációját helyezük el, így a sorokban, illetve oszlopokban lévő számok összege mindig

$$\frac{(1+n)n}{2} \quad (4)$$

Az világos, hogy az előállítandó bűvös négyzet összes elemeinek összege

$$\frac{(1+n^2)n^2}{2} \quad (5)$$

(5)-ből következik, hogy a bűvös négyzet sor, illetve oszlop összegei:

$$\frac{(1+n^2)n^2}{2n} = \frac{(1+n^2)n}{2} \quad (6)$$

Azaz (6)-ból (4)-et kivonva kapjuk meg az  $L(b_{ij})$  latin négyzet sor, illetve oszlop összegeit:

$$\frac{(1+n^2)n}{2} - \frac{(1+n)n}{2} = \frac{n^2(n-1)}{2} \quad (7)$$

Amennyiben feltételezzük, hogy  $L(b_{ij})$  sorai és oszlopai is egy  $b_1, b_2, \dots, b_n$  számtani sorozat elemeit tartalmazzák, úgy (7) alapján felírhatjuk a sorozat összegére vonatkozóan:

$$\frac{(b_1 + b_n)n}{2} = \frac{n^2(n-1)}{2} \quad (8)$$

Legyen  $b_1=0$ , ekkor  $b_n=b_1+(n-1)d=(n-1)d$ , ahol  $d$  jelöli a számtani sorozat differenciáját. Ezt (8)-tel összevetve kapjuk:

$$\frac{(n-1)dn}{2} = \frac{n^2(n-1)}{2} \Rightarrow d = n \quad (9)$$

Az  $L(b_{ij})$  latin négyzet elemei tehát  $0, n, 2n, \dots, n(n-1)$  számok lesznek, ahogy ezt a 14. ábra első latin négyzeténél láthatjuk is. Ezeket a számokat a bűvös négy-

zet szabályait kielégítő módon helyezük el az  $L(b_{ij})$  mátrixban.

Ehhez elsőként válasszuk ki az  $L(a_{ij})$  latin négyzet egy tranzverzálisát. Az azonos tranzverzálishoz tartozó elemek azonos szürkeárnyalattal szerepelnek a fehértől a feketéig), ez a definíció szerint az  $L(a_{ij})$  mátrix minden sorából és oszlopából pontosan egy elemet tartalmaz, amelyek az  $1, 2, 3, \dots, n$  számokból állnak. Ha tehát az  $L(b_{ij})$  mátrixban ennek a tranzverzálisnak megfelelő helyek mindegyikére a 0 értéket írjuk, akkor az  $M(c_{ij})$  mátrixban a  $c_{ij}=a_{ij}+b_{ij}$  elemek értékei, rendre az  $1, 2, 3, \dots, n$  számok lesznek. Most keressünk az  $L(a_{ij})$  latin négyzetben egy másik tranzverzalist és az ennek megfelelő helyekre írjuk az  $L(b_{ij})$  mátrixban az  $n$  számot. Ekkor az  $M(c_{ij})$  mátrixban a  $c_{ij}=a_{ij}+b_{ij}$  elemek rendre az  $n+1, n+2, n+3, \dots, 2n$  értékeket veszik fel. Ezt az eljárást folytatva az  $L(a_{ij})$  latin négyzet  $n$  darab tranzverzálisával, az eredmény  $M(c_{ij})$  mátrix pontosan a kívánt  $1, 2, 3, \dots, n^2$  számsorozat értékeit fogja tartalmazni és az  $L(b_{ij})$  mátrix is egy latin négyzet lesz. Ezzel az állításunkban megfogalmazott összes feltételt teljesítettük, amit a lenti ábrákon lépésről-lépésre követhetünk.

A konstrukcióból egyértelműen adódik, hogy  $L(a_{ij})$  és  $L(b_{ij})$  ortogonális párok. Itt érzékelhetjük, hogy az ortogonális párok és a tranzverzálisok között milyen szoros kapcsolat van. Továbbá feltételeztük  $n$  darab tranzverzális létezését. Ennek általános érvényű bizonyítása még várat magára, de megmutatjuk, hogy a fenti bűvös négyzetek konstrukciójára vonatkozó algoritmusunk nem megalapozatlan, igaz ugyanis az alábbi tétel:

*Ha egy  $n$ -ed rendű latin négyzet ciklikus, akkor létezik benne  $n$  darab tranzverzális, amelyek éppen a főátlóra vonatkozó tört diagonálisok elemei (a tranzverzálisok azonos szürke árnyalattal vannak satírozva az alábbi, 15. ábrán).*

1	2	3	4	5
2	3	4	5	1
3	4	5	1	2
4	5	1	2	3
5	1	2	3	4

15. ábra

### Latin négyzet játék

Végül a latin négyzetekkel való megbarátkozáshoz segítségül bemutatok egy latin négyzet társasjátékot. A játékot két játékos játsza,  $A$  és  $B$  (mindig  $A$  teszi meg az első lépést). A játékot egy  $n \times n$  méretű üres táblán játsszák. Kezdetkor  $A$  egy  $1$  és  $n$  közötti számot a táb-

L(a <sub>ij</sub> )					L(b <sub>ij</sub> )					L(a <sub>ij</sub> )+L(b <sub>ij</sub> )				
5	3	1	4	2	10	15	0	5	20	15	18	1	9	22
4	2	5	3	1	0	5	0	10	15	4	7	25	13	16
3	1	4	2	5	20	10	15	0	5	23	11	19	2	10
2	5	3	1	4	15	0	5	0	10	17	5	8	21	14
1	4	2	5	3	5	20	10	15	0	6	24	12	20	3

14. ábra

la tetszés szerinti helyére ír, majd  $B$  úgy ír a tábla még üres mezőire egy szintén  $1$  és  $n$  közötti számot, hogy az a latin négyzet tulajdonságának ne mondjon ellent (az ilyen lépéseket legálisnak nevezzük). Aki utoljára tud legális lépést írni a táblára, az nyer.

Egy  $4 \times 4$ -es táblán lehetséges játékot mutatunk be az utolsó, 16. ábrán. A fehér mezőkre írt számok  $A$ , a szürke mezőkre írtak  $B$  lépéseit mutatják.

1	2	3	
2	3	1	
3	1	2	4
4			1

16. ábra  
A latin négyzet játék  
egy lejátszása

Mivel  $A$  kezdett, így hat lépés után láthatjuk, hogy  $B$  nyert, mivel  $A$ -nak már nincs helyes lépése. Harary és Leary [6]-ban azt bizonyították be, hogy ha  $n$  páros, akkor  $B$  nyer, ha  $n$  páratlan, akkor  $A$  nyer.

### Irodalom

- [1] Claude-Gasper Bachet:  
Problèmes plaisant et detectables, 1612.  
[2] Berger György: Bűvös négyzetek  
Dacia Könyvkiadó, Kolozsvár, 1986.

- [3] J. Chernick:  
Solution of the general magic square.  
Amer. Math. Monthly 4(1938) 172–175.  
[4] J.Dénes, A.D. Keedwell:  
Latin squares and their applications.  
Academic Press, New York, Akadémiai Kiadó, Bp.,  
English Universities Press, London, 1974.  
[5] A. H. Frost:  
The construction of Nasik squares of any order.  
Proc. London Math. Soc. 27 (1985-96) 487–518.  
[6] F. Harary, T. Leary:  
Latin square schiefment games.  
J. Recreational Math.16 (1983/84) 241–246.  
[7] D.King:  
Magic square puzzles.  
Frederich Müller Limited, London, 1984.  
[8] Kiss Elemér:  
Matematikai kincsek Bolyai kéziratos hagyatékából  
Akadémiai Kiadó, Typotex Kiadó, Budapest, 1999.  
[9] M. Ozanam:  
Recreations mathématique et physiques.  
Tome 1-4. Paris, Claude Joubert, 1723.  
[10] Poignard:  
Traité des Quarrées sublimes contenant  
d Methodes Generales, toutes Nouvelles e faciles,  
pour faire les sept Quarres planetaires et tout  
autre a l infine, Brüssel, 1794.

## Hírek

A „**The Oracle Grid Index**” kezdeményezés keretében rendszeresen értékeli a számítóhálózatos technológiák terjedését, és a bevezetésük iránti hajlandóságot. Az Oracle Grid Index olyan nulla és tíz közötti mutatószám, amely a számítóhálós megoldásokkal kapcsolatos, az európai vállalatok körében végzett felméréseken alapul.

A 2004 őszére vonatkozó európai Oracle Grid Index értéke 3,1. Ez a szám önmagában csak nagyvonalakban mutatja a számítóhálózatok terjedését az európai informatikában, az index mögött álló mutatók és adatok azonban néhány érdekes tényre és statisztikára világítanak rá. A kutatások emellett rávilágítottak a grid computing jellegű technológiák bevezetésére irányuló egyes döntések hátterére. A számítóhálózatokat nagymértékben támogató válaszadók több mint negyven százaléka úgy nyilatkozott, hogy informatikai struktúrájuk átfogó terhelése és kihasználtsága meghaladja az átlagot, ami azt sugallja, hogy a grid computing korai bevezetését ösztönzi az informatikai kapacitások kiegyensúlyozására irányuló törekvés. A válaszadók többsége (51 százalék) emellett kijelentette, hogy a számítóhálózatok fő előnye az informatikai beruházások és üzemeltetési költségek szintjének átfogó csökkentése.

A **Sun Microsystems** egyik munkatársát, Dr. Robert Drost vezető kutatót a **világ 100 legjobb fejlesztője** közé választották. Dr. Dorst az elismerést a nyomtatott áramkörtől a lapkák közötti „proximity” kommunikációs, vagyis valós, molekuláris fizikai kapcsolat nélküli adatátvitel technológia kidolgozására irányuló kutatásaiért kapta.

A kutatási eredmények a számítógépek teljesítményének megváltoztatását ígrik. Ezzel az újítással lehetővé válna következő generációs szuperszámítógépek megépítése olyan nagy adatfeldolgozási igényű alkalmazások teljesítményének jelentős javítására, mint a távoli galaxisok feltérképezése, a fehérjék térbeli szerkezetének szimulációja, az orvosi kezelések eredményének megjósolása. A „proximity” kommunikáció folyamán egy lapkapár egymással szemben helyezkedik el, mikronokra egymástól, de nem érintkezve egymással. Ezáltal az egyik lapkán található adó áramkörök és a másikon lévő fogadó áramkörök között az adatcsere lapkán belüli sebességgel történhet.

# Optikai szűrők hatása optikailag átlátszó WDM hálózatok méretére

SZÖDÉNYI ÁKOS

BME, Távközlési és Médiainformatikai Tanszék, Nagysebességű Hálózatok Laboratórium  
sa239@hszk.bme.hu

**Kulcsszavak:** hullámhossz-nyalábolás, átlátszó optikai csatoló multiplexer, optikai szűrők, bithiba-arány

Az optikailag átlátszó hálózatok mérete (csomópontszáma) korlátolva van, hisz az ilyen „szigeteken” áthaladó optikai jelek információtartalma nem romolhat bizonyos bithiba-arány alá, ugyanakkor a jelregenerálás túl lassú és költséges. Három különböző optikai szűrőt alkalmazó optikai ki-becsatoló multiplexert fűztem fel tíz-tíz csomópontot alkotó láncba és az ezeken áthaladó egyforma jelek bithiba-arány romlását hasonlítottam össze szimulációs program segítségével.

Az optikai technológiák nagyon gyorsan fejlődnek hogy kielégítsék a nagy sávzélesség-igényű alkalmazásokat. Így a szolgáltatók rá vannak kényszerítve, hogy a jövőben ezen kihívást jelentő megoldásokat alkalmazzák. Átlátszó „szigetek” – melyekben az optikai jel opto-elektromos átalakítás nélkül terjedhet – telepítése egyszerűbben és költségkímélőbb módon megvalósítható, mint meglévő rendszerek elektromos funkcióit egyre inkább optikai szinten megvalósítani.

Ebben a cikkben egy városi hálózat átlátszó „szigetének” méretére teszek becslést az alkalmazott ki-becsatoló multiplexerek függvényében szimulációs program segítségével. Három féle ki-becsatoló multiplexert vizsgáltam tízet-tíz láncba fűzve. Az optikai jel minőségének meghatározására bithiba-arány becslést használtam, ami meghatározza az átlátszó „sziget” méretét.

Legjobb tudásom szerint eddig ez az első becslés egy átlátszó hálózat méretére az alkalmazott optikai eszközök és a célként kitűzött BER érték függvényében.

## A három optikai szűrő-típus

Multiplexer és demultiplexer egyaránt keskenysávú szűrőket alkalmaz, sorba kapcsolva vagy egyéb módon kombinálva, hogy elérjék a kívánt célt. Különleges technikák, melyeket az ilyen szűrésekhez használnak, állhatnak vékonyfilm szűrőkből, fiber Bragg vagy bulk grating-ekből, elvékonyodó szálból, tiszta kristálysűrőkből vagy integrált optikai szűrőkből (Arrayed Waveguide Grating, AWG). Abban a projektben, aminek keretében ez a munka folyt, bulk grating és AWG került beszerzésre, FBG csak szimulálásra.

### Diffrakciós rács (mux)

A bulk-optic diffrakciós rács [1] visszaveri a fényt egy bizonyos szögben, mely szög arányos a hullámhosszal és így erősítő és gyengítő interferencia jön létre. A beeső fény minden egyes hullámhossz egy adott szög alatt verődik vissza az optikai rácsvonalakról.

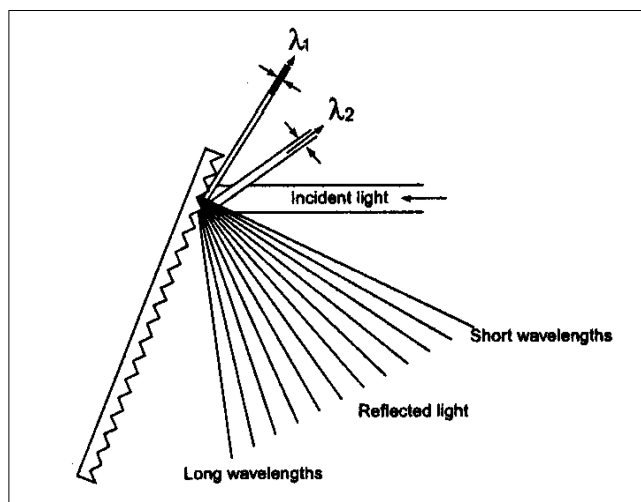
Ezek a visszavert hullámok fázisukban pont egy hullámhossz-osztásnyira fognak különbözni. Ennél a szögnél minden sugár eloszlása összegződni fog, így ez lesz az a szög, ahol az áteresztés maximális lesz erre a hullámhosszra nézve (1. ábra).

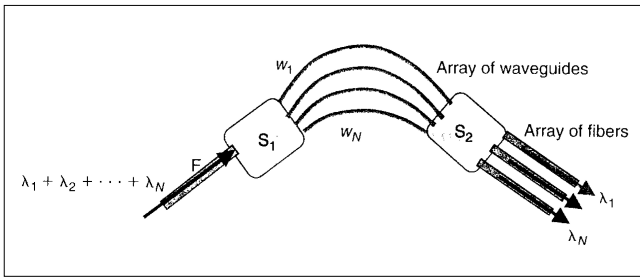
Diffrakciós ráccsal megvalósított mux és demux tervezésének lényege a bemenet és a kimenet pozícionálása a megfelelő hullámhosszak kiválasztásához. Jóllehet elég bonyolult gyártani őket, és elég drágák is, az ilyen eszközök beiktatási csillapítása lényegében független a hullámhossz-csatornák számától, ami ezt a technológiát a legígéretesebbé teszi sok hullámhossz csatornás rendszerekben. Mindenesetre kritikus hangsúlyt kell fektetni a polarizációra.

### Arrayed Waveguide Grating (tömbös hullámvezető rács)

AWG-knek [2,3] több nevük is van: phased-array grating (PHASAR) és waveguide grating router-ként (WGR) is ismertek. Az előbb tárgyalt bulk diffrakciós ráccsal szemben AWG-k óriási előnye, hogy hullámhossz függetlenek (véges hullámhossz tartományon), és ezért periodikusak. Egy AWG feladata hasonló a multiplexer/demultiplexerhez, mivel szét tud választani,

1. ábra Bulk diffrakciós rács





2. ábra Arrayed Waveguide Grating (AWG)

és össze tud nyalábolni különböző hullámhosszakot, melyek egy egymódusú szálabban terjednek. Csak a megvalósítási módszer különbözik.

AWG működése interferometrián alapszik. A felépítése a 2. ábrán látható. A bejövő fény a több különböző hullámhosszú fény sugarat csatol az első rendezőbe ( $S_1$ ), mely egy hullámvezetőkből álló tömbhöz van optikailag csatolva. Ahány hullámhossz van a bejövő szálabban, annyi hullámvezető van kivezetve  $S_1$ -ből. Ezen hullámvezetők hossza különbözik, tehát az optikai úthossz sem egyezik, így hullámhossz-függő szögeltolás jön létre ezen hullámvezető tömb végén a második rendezőben ( $S_2$ ).

Itt a különböző optikai úthosszat megtett fénynyalábok egymással csatolásba kerülve úgy interferálódnak, hogy minden hullámhossz intenzitás-eloszlása éppen egyetlen kimenetnél lesz maximális. Feltéve, hogy minden kilépő hullámfrontnak meg kell egyezzen a szöge (modulo  $2\pi$ ), két szomszédos optikai útnak meg kell, hogy egyezzen a beeső hullámfronttól a kilépő hullámfrontig tartó optikai úthossz-különbsége. Ez a különbség egyenlő egy egész számmal, mely meg van szorozva a hullámhosszal.

AWG-eket optikai útvonalválasztókként is használnak ha nem csak egy bemenő porttal rendelkeznek. Vegyünk például egy  $2 \times 2$ -es AWG-t két bemeneti és két kimeneti porttal. Tegyük fel, hogy mindkét porton kettő egyező hullámhossz érkezik: az A porton:  $\lambda_{1A}$  és  $\lambda_{2A}$  illetve a B porton:  $\lambda_{1B}$  és  $\lambda_{2B}$  ahol  $\lambda_{1A} = \lambda_{1B}$  és  $\lambda_{2A} = \lambda_{2B}$  természetesen más információ tartalommal. Ilyenkor az AWG kicseréli a hullámhosszakot úgy, hogy az egyik kimeneten lesz:  $\lambda_{1A}$  és  $\lambda_{2B}$  a másikon pedig:  $\lambda_{1B}$  és  $\lambda_{2A}$ . Eközben a két azonos hullámhossz nem keveredik! Ezek az útvonalválasztók ígéretes jelöltek a jövő átlátzó hálózataiban. Mostanában például felhasználták egy speciális hálózati topológia szabadalmi igényhez is, mely az Egyesült Államok szabadalmi hivatalánál van jelenleg, a Berliini Egyetem adta be, és Ringostar a neve.

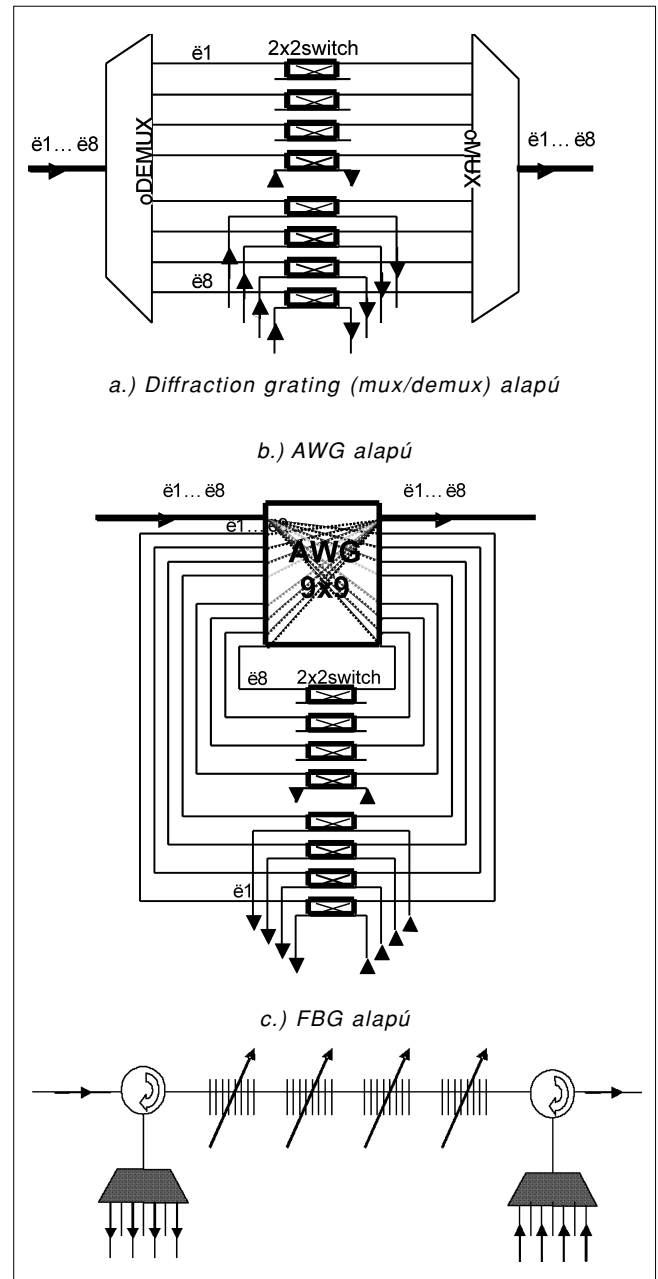
**Fiber Bragg Grating (FBG)**

Ezek az eszközök olyan szűrők, amelyek az összes hullámhossz csatornát átengedik, kivéve egyet, amelyre tervezték őket, ezt visszaverik. FBG-k lehetnek hangolhatóak vagy nem hangolhatóak. A nevével ellentétben ezek nem rácsok. Azért hívják őket mégis rácsnak, mert ahogy a fény áthalad rajtuk, úgy érzi, mintha egy rácson haladna keresztül, és ezért úgy is viselkedik.

Hogy ez az eszköz visszaverjen egy adott hullámhossz sávot, periodikus törésmutató változást hoznak létre a szálabban. Ez az, amire a fény úgy reagál, mintha egy rácson menne át. Ennek létrehozására két technológia is rendelkezésre áll.

Az egyik a gyakrabban alkalmazott UV technológia. A Germánium adalékolt magot megvilágítják egy mintán keresztül, ami interferenciát hoz létre és egyben meg is változtatja periodikusan a szálab törésmutatóját. Ez a minta szoros kapcsolatban van a visszaverésre kiválasztott hullámhosszal. Minél hosszabb egy FBG, annál keskenyebb, meredekebb a visszavert hullámhossz-sáv. Ennek az az ára, hogy az eszköz hosszával nő a beiktatási csillapítása. Különböző FBG-eket egymás után téve egynél több hullámhossz-csatornát is kiválaszthatunk. Optikai cirkulátorral való összekapcsolá-

3. ábra OADM-ek felépítése



a.) Diffraction grating (mux/demux) alapú

b.) AWG alapú

c.) FBG alapú

sa lehetővé teszi, hogy WDM szálból egy hullámhosszat kicsatolhassunk. Egy másik alkalmazása ennek az összekapcsolásnak kromatikus diszperzió kompenzáció. Ezeket a speciális FBG-et chirped FBG-knek hívjuk.

**Az optikai csomópontok felépítése**

A kísérleti hálózat egy ASON/GMPLS optikai hálózat, melynek átviteli síkja (transport plane) három újrakonfigurálható optikai ki-becsatoló multiplexerből (OADM) áll, vezérlő síkja (control plane) és menedzsment síkja (management plane) pedig lehetővé teszi az intelligens optikai csatorna kiszolgálást.

Egy speciális OADM elrendezés három különböző megvalósítását vizsgálom. Az első bulk grating optikai multiplexert és demultiplexert alkalmaz 2x2-es optikai kapcsolókkal együtt, a második egy AWG-t használ ugyancsak 2x2-es optikai csatlakozókkal, a harmadik pedig négy darab FBG-t (3. ábra). Nyolc darab ITU csatorna van kialakítva [193,0THz(=1553,33nm)-tól 193,7 THz(=1547,715nm)-ig] 100GHz (0,78nm) csatornák közti távolsággal, jóllehet csak maximum 4 csatornát lehet egy-egy csomópontban (OADM) ki-be csatolni. A kísérlet célja továbbá az, hogy felfedezzük milyen mértékben lehet majd ezt a kísérleti hálózatot a jövőben kibővíteni, és/vagy más átlátszó hálózattal összekapcsolni.

**Eredmények és értékelés**

A szimuláció [4] során ugyan azt az optikai jelet engedjük keresztül tíz mux csomóponton, tíz AWG csomóponton és tíz FBG optikai csomóponton. Mindegyik csomópont előtt egy EDFA elő-erősítőt helyeztünk el, a csomópontok között 35km-es fényvezető szál van. A szimulációs modellekhez, eszközökhöz a paramétereiket a megvásárolt eszközök tesztelési adatlapján lévő a legrosszabb esetben mért értékeket használtuk fel és adtuk be a szimulációs programba.

Legfontosabb összehasonlítás a három technológia között a mi esetünkben a beiktatási csillapítás volt; a mux esetében 4dB, az AWG esetében 10 dB, FBG

esetében 2.8dB hiszen minden egyéb adat megegyezett. Az optikai jelek – mivel nem kerültek kicsatolásra egyetlen hullámhosszon sem, egyetlen csomópontban sem – kétszeres csillapítást szenvedtek, hiszen először demultiplexálni, majd multiplexálni kellett őket. (FBG csomópont esetében ez nem áll fenn.)

A bithiba-arányt a program következő képlet alapján becsüli:

$$BER = \frac{1}{2} \operatorname{erfc} \left( \frac{Q}{\sqrt{2}} \right)$$

ahol

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{+\infty} e^{-t^2} dt$$

és Q a jósági tényező (quality factor) [4].

A 'Q' jósági tényező a digitális jelek szemábra-aperatúrájának (nyitottságának) mértéke, egy olyan kiértékelési metódus, amely normális zajeloszlást feltételez [5] és a következő képlet alapján adható meg:

$$Q = \frac{|\mu_1 - \mu_0|}{\sigma_1 - \sigma_0} \quad [6]$$

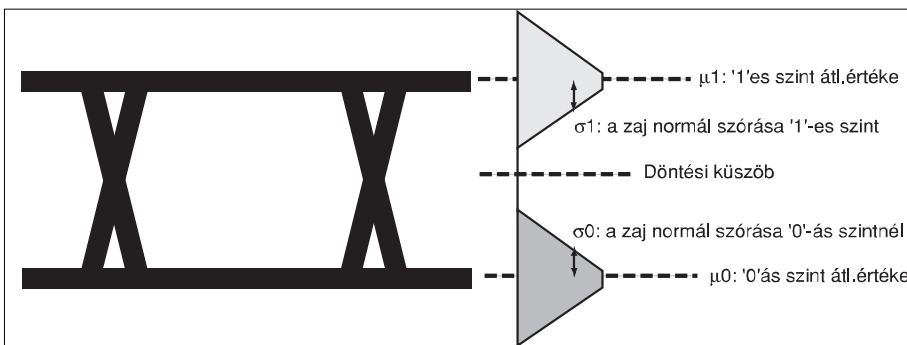
ahol  $\mu_1$ : a logikai '1'-es szint,  $\mu_0$ : a logikai '0' szint átlagos értéke, és  $\sigma_1$  illetve  $\sigma_0$  pedig ugyanígy az '1'-s és a '0'-ás szint normál szórása, ahogyan az a 4. ábrán látszik. A vevődióda döntési küszöb feszültség változtatásával a rendszer érzékenységét vizsgálva tesz becslést a szimulációs program a Q értékre. Ez az úgynevezett „Variable Threshold Method” amely részletesen ismertetve van az ITU-T G.976 (1997) ajánlásban, illetve a [7]-es észak-amerikai szabványban.

A következő oldalon, az 5. ábrán láthatóak a becsült BER adatok annak függvényében, hogy a jel hány csomóponton haladt már át. Maximum 10 csomópont-ra végeztük a szimulációt. Az ábra határt szab a keresztelhető átlátszó optikai csomópontok számának a becsült BER értékek függvényében mindhárom csomópont-fajtára.

Miután a fény keresztülhaladt tíz csomóponton, az optikai jel minősége (BER) lecsökken  $10^{-4}$  re bulk-grating mux csomópontok esetében,  $10^{-6}$ -ra AWG és  $10^{-9}$ -re FBG nyalábolók esetében. (Az első három csomópont kimenetén a bithiba-arány nem volt mérhető.) Ha az optikai szolgáltatások osztályokba lennének sorolva négy kategória szerint, az 1. táblázat alapján, akkor – tíz csomópontból álló hálózat esetén – az

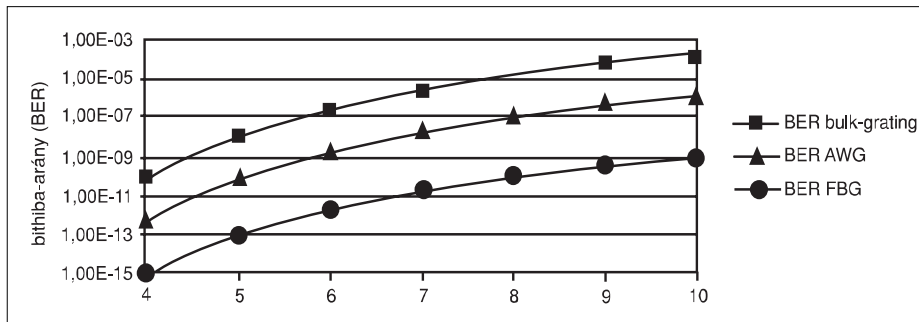
5. ábra alapján, az összes szolgáltatási osztály elérhető lenne, mindhárom csomópont-hálózat esetében a becsült BER értékek alapján még a Premium is, amely a legszigorúbb a négy közül.

4. ábra  
A Q jósági tényező kiszámításának szemléltetése (balra egy egyszerűsített szemábra látható)

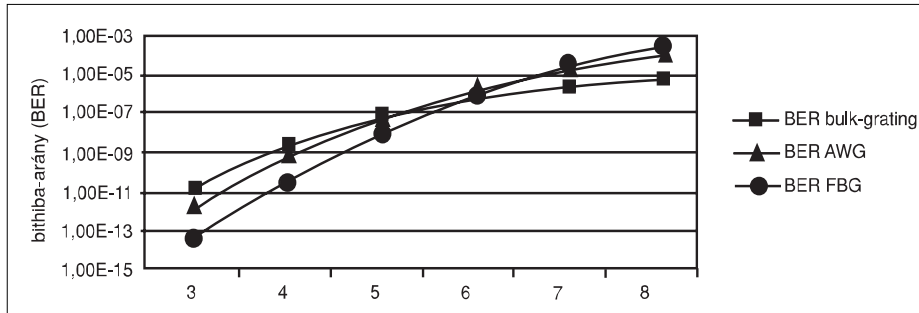


	Premium	Gold	Silver	Bronze
Out-of-service	Degraded	Degraded	Fault	Fault
	BER=10 <sup>-4</sup>	BER=10 <sup>-4</sup>	LOS	LOS

1. táblázat  
Szolgáltatáson kívüli kritériumok különböző szolgáltatási osztályokra [8]



5. ábra Bithiba-arány a csomópontszám függvényében



6. ábra Csomópontszám BER függvényében ha a csatornatávolság 50GHz

Jóllehet más kritérium, mint például az összeköttetés létesítési idő, visszaállítási idő mind beleszámít az optikai szolgáltatások minőségének biztosításába.

Másrészről, ha BER határérték egy szigorúbb mondjuk  $10^{-9}$  értékre lenne kitűzve, akkor a szituáció teljesen más erre a három technológiára nézve, hiszen az átlátszó sziget bulk-grating mux esetben négy, AWG esetben öt csomópontból állna, FBG esetben pedig akár tízből is állhat.

Ha csupán 50GHz csatornák közötti távolságot definiálunk, akkor a bithiba-arány becslés egy kicsit rosszabb eredményt ad, különösen FBG esetben. Ez utóbbi annak köszönhető, hogy a közelebb kerülő csatornákon nagyobb az áthallás, mely kevésbé kiküszöbölhető, ha a mux és AWG esettel ellentétben FBG esetben nincs minden csomópontnál optikai szűrés (nem kibe csatolási esetben). Ezen 50GHz csatorna távolság esetén kapott BER értékek az 6. ábrán láthatók.

## Összefoglalás

Teljesen-optikai eszközök és rendszerek továbbfejlesztésének egy iránya lehet az, hogy teljesen-optikai és opto-elektromos technológiák együtt, egymás mellett léteznek. Ekkor átlátszó-optikai „szigetek” jöhetnek létre és ezeket 2R/3R regenerátorok kapcsolhatják össze. Egy ASON/GMPLS hálózat szemszögéből [9] ezek az „optikailag átlátszó szigetek” különböző tartományoknak (domain) tekinthetők a kontroll és menedzsment síkon megkönnyítve ezen feladatok elosztását és elvégzését.

Ezen szigetek megfelelő méretezése elsődleges fontosságú lenne a kívánt szolgáltatási szint specifikáció (SLS, Service Level Specification) követelményei-

nek biztosításában (beleértve a BER méréseket is). Ebben a cikkben bemutatásra került hogy az optikai csomópont felépítése befolyással van az optikailag átlátszó hálózat méretére. Továbbá ebben a tárgyalt egyedi hálózat esetében FBG-vel megvalósított OADM megoldások sokkal előnyösebbek akkor, ha az átlátszó sziget méretét szeretnénk megnövelni. Megéri tehát limitálni a méretet nem csupán technikai megfontolásokból, hanem szabályozási kényszerek, együttműködési képesség, költség és nem utolsósorban hálózat-vezérlési és menedzsment szemszögéből.

## Köszönetnyilvánítás

A szerző köszönetet mond a Katalán Telekommunikációs Kutató Központ (CTTC) Optikai Laboratóriumának (Barcelona), ahol a munkát az EMPIRICO projekt keretében végezte.

## Irodalom

- [1] Ghislain Lévesque, Vance Oliver, „Guide to WDM Technology and Testing” 2000 EXFO Electro-Optical Engineering Inc., Quebec City, Canada, ISBN 1-55342-001-2
- [2] K. A. McGreer, „Arrayed Waveguide Gratings for Wavelength Routing”, IEEE Communications Magazine, Dec.1998, pp.62–68.
- [3] S. V. Kartalopoulos, „Introduction to DWDM Technology” USA, IEEE Press, ISBN 0-7803-5399-4
- [4] VPI photonics™
- [5] ANRITSU CORPORATION, Application Notes: Q Factor Measurement/Eye Diagram Measurement, SDH/SONET Pattern Editing [http://www.eu.anritsu.com/files/MP1632\\_1763\\_1764\\_EF1100.pdf](http://www.eu.anritsu.com/files/MP1632_1763_1764_EF1100.pdf)
- [6] MSZ EN 61280-2-8 Fibre optic communication subsystem test procedures – Digital systems, Part 2-8: Determination of low BER using Q-factor measurements (IEC 61280-2-8:2003), Magyar Nemzeti Szabvány
- [7] TIA/EIA-526-9: OFSTP-9: Accelerated Measurement Procedure for Determining BER and Q-factor in Optical Transmission Systems Using the Variable Threshold Method
- [8] W. Fawaz et al., IEEE Communications Magazine Volume 42 (2004), pp.36–43.
- [9] G.8080/Y.1304 (11/2001.)

# Intelligens rendszerek egységes tervezése

KOVÁCS DÁNIEL LÁSZLÓ

BME, Villamosmérnöki és Informatikai Kar, Méréstechnika és Információs Rendszerek Tanszék  
dkovacs@mit.bme.hu

Reviewed

**Kulcsszavak:** ágens, evolúció, játék elmélet, racionalitás, tervekészítés

Jelen cikk amellel érvel, hogy az ágens-, játék- és evolúciós-elméletek egyesítéséhez, avagy az – intelligens rendszerek egységes tervezését támogató – átfogó rendszer-specifikációs elv kidolgozásához a „racionalitás” fogalma nyújt elsősorban segítséget. Elsőként tehát áttekintjük a fontosabb racionalitás-fogalmakat. Ezt követően rátérünk az ágens-elméletre. Ezen belül foglalkozunk az ágensek architektúrájával, környezetével, a megoldandó problémákkal, végül az ágensek racionalitásával, avagy a korlátosan optimális ágensekkel. Ezek után példát adunk egyszerűbb játékokra, majd ezek vizsgálatán keresztül bemutatjuk a fontosabb játék-elméleti racionalitás fogalmakat. Rámutatunk a játék- és ágens-elmélet összekapcsolási lehetőségeire, majd bevezetjük a korlátosan optimális játékosok fogalmát. Végül összefoglaljuk a két elmélet összekapcsolásának következményeit. Ezt az evolúciós-elmélet rövid áttekintése követi, miután rátérünk a genetikai algoritmusokra. Az alapfogalmak tisztázását követően rámutatunk az elmélet ágens- és játék-elmélettel való összekapcsolási lehetőségeire. Végül áttekintjük a három elmélet összekapcsolásának következményeit.

## 1. Bevezetés

Az intelligens rendszerek fontos szerepet játszanak mindennapjainkban, kezdve az orvosi diagnosztikától, a katonai tervezésen át egészen az intelligens háztartásokig. Céljuk többek közt az emberi munkavégzés segítése kezelhetetlen bonyolultságú feladatok esetén (például nagykomplexitású hálózatok menedzselése), esetleg az emberi munkaerő kiváltása nehezen hozzáférhető környezetekben (mélyűr, tengerfenék, katasztrófa helyszínek stb). Az intelligens porszívótól, a marsi felderítő járműveken át egészen az Interneten baráncoló szoftver-ágensekig lassan már mindenhol megtalálhatók. Legfontosabb közös jellemzőjük az adaptív feladatmegoldó készség.

Az adaptív feladatmegoldó rendszerek tervezésekor a legfőbb nehézséget egy átfogó rendszerspecifikációs elv hiánya jelenti. Több elmélet is foglalkozik az ilyen jellegű rendszerek tervezésével, ám mindegyik másképp közelít a problémához. A mesterséges intelligencia területén oly népszerű ágens-elmélet a rendszer „intelligenciájára”, a játék-elmélet a rendszer „racionalitására”, az evolúciós-elmélet pedig a rendszer „optimalitására” helyezi a hangsúlyt. Bár az említett fogalmak első hallásra igen különbözőnek tűnhetnek, mégis közös nevezőre hozhatók, lehetőséget adva az elméletek egyesítésére, egy átfogó rendszer-specifikációs elv kidolgozására.

Egy ilyen elv előnye, hogy lehetővé teszi az intelligens rendszerek tervezésekor felmerülő különböző szempontok egységes, ám szempontokhoz illeszkedő tárgyalását. Például a rendszer benső felépítésének kialakításához az ágens-elméletet; a rendszer csoporton belüli működésének kialakításához a játék-elméletet; míg rendszerek csoportos viselkedésének kialakításá-

hoz az evolúciós-elméletet hívhatnánk segítségül, biztosítva a különböző megközelítések közti átjárhatóságot. Ily módon a megtervezett rendszer viselkedése, kommunikációs és együttműködési készsége, hasznosága, robusztussága, vagy más egyéb jellemzői egységes elvek alapján, akár többféle szemszögből is tervezhetők volnának.

## 2. A racionalitásról általában

Az „intelligens” jelző sok félreértésre ad okot, hiszen rengeteg értelmezést kínál, sőt, talán még igazán pontos definíciója sincs. Amennyiben nem emberekre, hanem mesterséges rendszerekre alkalmazzuk, úgy általában a rendszer önálló, adaptív feladat-felismerő és megoldó képességét, hatékony erőforrás kihasználását és célratörő működését értjük alatta. Tehát ilyen értelemben egy rendszert akkor nevezünk „intelligensnek”, ha ésszerű (racionális) a működése; ha emberi intelligenciát igénylő feladatok megoldására is alkalmazható. Az „intelligens” jelző tehát valójában a rendszerek racionalitását takarja.

A filozófusok és a közgazdászok már jóval a mesterséges intelligencia területének létrejötte előtt elkezdtek keresni a racionális viselkedés kielégítő meghatározását. Erre azért volt szükség, mert a racionalitás fogalma az etika, az induktív tanulás, a következtetés, a döntéshozás és a közgazdasági modellek meghatározó tényezője. Több különböző racionalitás-fogalom is napvilágot látott:

### • Tökéletes racionalitás:

A filozófia és a közgazdaságtan klasszikus racionalitás-fogalma. Arisztotelész etikájából fejlődött ki, míg végül a döntés-elméletben [1] nyert formális megfogal-

mazást. Azt a rendszert nevezzük tökéletesen racionálisnak, mely minden pillanatban olyan döntést hoz tapasztalatai alapján, amivel maximálja várható nyereségét. Mivel a megfelelő döntés meghozatala számítási-igényes feladat, és ehhez időre van szükség, ezért nem-triviális esetekben nemigen beszélhetünk tökéletesen racionális rendszerekről. A tökéletes racionalitás tehát a gyakorlatban *nem megvalósítható* követelmény.

• **Kalkulatív racionalitás:**

A mesterséges intelligencia által vizsgált racionalitás-fogalom. Egy kalkulatíve racionális rendszer olyan döntést hoz, amely számításai megkezdésekor tökéletesen racionális lett volna. Sajnos a gyakorlatban ez a követelmény *sem használható*, mivel egyrészt akár kívárhatalan sokáig is eltarthat, amíg a rendszer meghozza döntését, másrészt (különösen dinamikus, folytonosan változó környezetekben) a hozott döntés jóságát már semmi sem garantálja. Ezért is fordul elő a gyakorlatban, hogy az intelligens rendszerek tervezői elrugaszkodnak az elvektől, és ad-hoc módon alakítják ki rendszereiket.

• **Metaszintű racionalitás:**

A kalkulatív racionalitás által felvetett problémákra adott válasz. Egy metaszinten racionális rendszer a döntések meghozásáért felelős objektum-szintű számítások felett optimalizál [2]. Tehát kiválasztja az optimális „számítás+döntés” kombinációt, ahol a döntés a számítás eredménye. Sajnos azonban a teljes metaszintű racionalitás igen ritka, hiszen egyrészt már maguk a metaszintű számítások is időt igényelnek, másrészt a metaszintű döntési probléma gyakorta nehezebb, mint az eredeti, objektum-szintű probléma. Ennek ellenére a gyakorlatban hasznosnak bizonyult a metaszintű racionalitás néhány egyszerűbb közelítése. Mindazonáltal ez is csak azt mutatja, hogy az intelligens rendszerek egységes tervezésének mindmáig nincsen elfogadott, átfogó rendszer-specifikációs elve. A gyakorlatban tehát a metaszintű racionalitás *sem elvárható* követelmény.

• **Korlátos racionalitás:**

A közgazdaságtan válasza a tökéletes racionalitás problémáira. Amíg tehát a tökéletes racionalitás a rendszer (például gazdasági entitás) erőforrásaitól, döntéshozási képességeitől és a döntéshozásra rendelkezésre álló idejétől függetlenül definiálta a racionalitást, addig a korlátosan racionális [3] rendszerek e tényezők, pontosabban döntéshozási korlátaik figyelembevételével maximálják várható nyereségük. Magyarán lehetőségeikhez mérten a lehető legjobb eredményt produkálják. A korlátos racionalitás tehát a racionalitás egy *használható* definíciója.

Tehát átértékelődik a racionalitás tárgya. Amíg a tökéletes racionalitás még szigorúan csak a döntések racionalitására fókuszál, addig a metaszintű, illetve a korlátos racionalitás már a rendszer működésére, a döntések meghozásáért felelős mechanizmusokra (is) vonatkozik. Észrevehető a racionalitás definíciójának abszolúttól a relatív irányában történő elmozdulása.

Amíg tehát a tökéletes racionalitás gyakorlatilag figyelmen kívül hagyja a rendszer képességeit, addig a kalkulatív racionalitás már közvetve ezekről tesz megállapítást; a metaszintű racionalitás már a rendszer funkcionális felépítését is taglalja; míg végül a korlátos racionalitás már a rendszer egészének viszonylatában definiálja a racionalitást.

Ennek a szemléletbeli változásnak több oka is van. Elmozdulás történt a közgazdaságtanban a tökéletes-től a korlátos racionalitás felé; a játék-elméletben a cselekvések kiválasztásától a – cselekvéseket előállító – programok kiválasztása felé; az evolúciós-elméletben a megoldások evolválásától a megoldásokat előállító programok evolválása [4] felé; a filozófiában pedig a cselekedet-utilitarizmus (vagy tett-utilitarizmus) felől a szabály-utilitarizmus, avagy a cselekvések meghozásáért felelős általános mechanizmusok racionalitása felé.

Beláthatjuk, hogy e folyamat gyökerei igen egyszerűen visszavezethetők a racionális rendszerek tervezhetőségére, hiszen így már nem támasztunk velük szemben olyan kritériumokat (például nem várunk el olyan döntéseket), melyeket a rendszer egyetlen döntéshozási mechanizmusa se tud teljesíteni. Végére is a rendszer által véghezvitt döntésekről és számításokról a rendszer tervezője csak közvetve gondoskodik, viszont az őket előállító döntéshozási mechanizmusokról (programokról) már közvetlenül. Ezek a megfontolások vezettek végül a mesterséges intelligencia legújabb racionalitás-fogalmához – a *korlátos optimalitás*-hoz [5]. A következőkben ezzel foglalkozunk.

### 3. Ágens-elméleti racionalitás

A mesterséges intelligenciában az előbb tárgyalt rendszereket (melyek maguk képesek eldönteni, hogy mit kell tenniük ahhoz, hogy teljesítsék a rájuk szabott feladatot) *ágens*nek nevezzük. Az „ágens” jelentését illetően azonban mégsem alakult ki általános konszenzus. Szakterületről szakterületre változik az ágensekkel kapcsolatos jellemzők fontossága. Bizonyos alkalmazásokban például elengedhetetlen, hogy az ágens képes legyen *tanulni*, míg nem másokban szükségtelen. Talán csak az *autonómia*, mint központi megfontolás, tekinthető az ágensek egyetlen, általánosan elfogadott, közös jellemzőjének. Mivel azonban valamiféle definícióra mindenképp szükségünk lesz a későbbiekben, kezdetnek két neves szerző – egymásnak összhangban lévő – javaslatát vesszük alapul:

„Egy ágens bármi lehet, amit úgy tekinthetünk, mint ami az érzékelői segítségével érzékeli a környezetét, és beavatkozó szervei segítségével megváltoztatja azt.” [6]

„Ágensnek nevezzük azt a rendszert, mely adott környezetbe helyezve képes autonóm cselekvésre céljai elérése érdekében.” [7]

Az 1. ábra alapján tehát az ágensek tekinthetők olyan függvényeknek, melyek érzetek sorozatából képeznek cselekvésekké. Ennek formális definíciója lehet a következő:



**Definíció 1.**

*Ágens-függvény:*

$$f_{\text{Ágens}}: \text{Érzet\_történetek} \rightarrow \text{Cselekvések}$$

Az ágens-függvény specifikálja tehát, hogy egy-egy érzettörténet (azaz érzetek egy-egy sorozatának) hatására mit cselekszik az ágens. Ily módon lehetőség nyílik a *tökéletesen* és a *kalkulatíve racionális ágensek* közti különbségtételre.

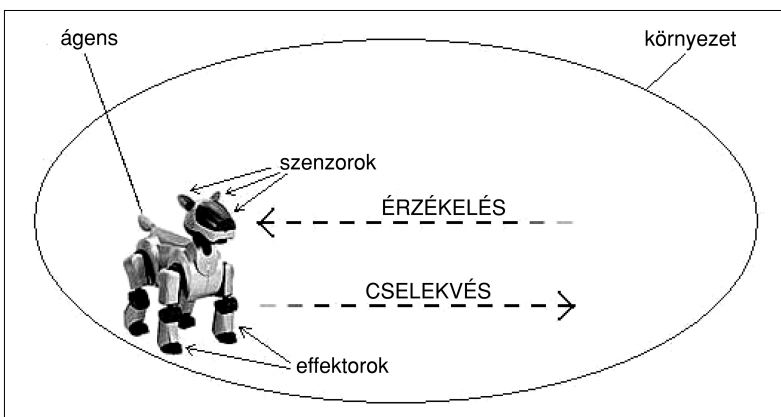
Az ágenseket beágyazó környezetet (amely akár más ágenseket is magába foglalhat), röviden *agens-környezetnek* nevezzük. Az ágens-környezetek többféle osztályozása lehetséges. Jelen cikk szemszögéből a következő szempontok érdekesek:

• **Hozzáférhető, vagy nem hozzáférhető:** A környezet hozzáférhető, ha az ágens érzékelői segítségével hozzáférhet a környezet teljes állapotához. Egy környezet ténylegesen hozzáférhető, ha az ágens érzékelése lefedi a környezet mindazon aspektusait, melyek szükségesek lehetnek egy-egy cselekvés kiválasztásához. Minél inkább hozzáférhető egy környezet, annál egyszerűbb felépítésű ágensek lesznek elegendőek hozzá. Nem hozzáférhető környezetre lehet példa a fizikai valóság, vagy akár az Internet stb.

• **Determinisztikus, vagy nem determinisztikus:** A környezet determinisztikus, ha tetszőleges állapotában az ágens által végrehajtott tetszőleges cselekvés egyértelműen meghatározza a cselekvés nyomán előálló következő környezeti állapotot. Egy determinisztikus, ámde nem hozzáférhető környezetet az ágens akár nem-determinisztikusnak is érzékelhet, így aztán gyakran jobb a környezetet a determinizmus szemszögéből vizsgálni.

• **Epizódszerű, vagy nem epizódszerű:** A környezet epizódszerű, ha benne az ágens tapasztalata „epizódokra” bontható. Ilyen környezetben az ágens használja a diszkrét számú „epizódon” vett hasznától függ, ahol az egyes „epizódokon” vett hasznok függetlenek egymástól. Az epizódszerű környezetek egyszerűbbek, mint a nem epizódszerűek, mivel az ágensnek csak az adott „epizódon” belül kell előre tekintenie, hiszen az egyes „epizódokban” végrehajtott cselekvések nem befolyásolják a következő „epizód” alakulását.

1. ábra Környezetébe ágyazott ágens



• **Statikus, vagy dinamikus:** A környezet statikus, ha csak az ágens cselekvéseinek nyomán módosul az állapota, ám egyébként változatlan. A dinamikus környezetet az ágens hatáskörén kívül eső jelenségek, folyamatok, vagy más egyéb történések is befolyásolhatják, változtathatják. Szemi-dinamikus környezetről beszélünk, ha az idő előrehaladtával a környezet nem változik, de az ágens által elért haszon mértéke igen.

• **Diszkrét, vagy folytonos:** A környezet diszkrét, ha az ágens különböző lehetséges cselekvéseinek és észleléseinek halmaza véges, egyébként folytonos.

Most pedig térjünk rá az ágens-környezet formális megfogalmazására. Először is tekintsük a környezetet olyannak, amely minden pillanatban valamilyen állapotban van. A környezet lehetséges állapotainak halmaza legyen véges.

Ahhoz, hogy kezelni tudjuk azt az esetet, amikor a környezet nem teljesen hozzáférhető az ágens számára, vezessük be az  $f_{\text{Érzékelés}}$  függvényt. Ekkor tehát az ágens egy adott pillanatban – korlátos érzékelőinek köszönhetően – nem a teljes környezeti állapothoz, hanem annak csak egy  $f_{\text{Érzékelés}}$  (Állapot) leképződéséhez fér hozzá. Továbbá, az ágens cselekedeteinek környezetre gyakorolt hatását modellezzük egy  $f_{\text{Átmenet}}$  függvénnyel, amely az ágens adott cselekvése, és a környezet aktuális állapota alapján megszabja a környezet következő állapotát. Összefoglalva, az ágens környezetét az alábbiak szerint definiálhatjuk:

**Definíció 2.**

*Az ágens környezete*

$$\text{Környezet} = (\text{Állapotok}, f_{\text{Átmenet}}, f_{\text{Érzékelés}})$$

Látható, hogy az ágens-függvény és a környezet egyértelműen meghatározza a környezet állapotainak – ágens által kialakított – sorozatát: az állapot-történetet. Jelölje *Hatás* ( $f_{\text{Ágens}}$ , *Környezet*) azt az állapot-történetet, amit az  $f_{\text{Ágens}}$  függvény generál a *Környezet*-ben. Vegyük észre, hogy a fentiek alapján a környezetet diszkrétnek és determinisztikusnak definiáltuk. A környezet folytonos és nem-determinisztikus kiterjesztése ennél valamivel komplikáltabb.

Ezek után térjünk rá az ágensek felépítésére. Tekintsük az ágens egy olyan, benső állapottal rendelkező rendszernek, mely egy architektúra és egy program együttese. Az architektúra felelős a környezet és a program közti kapcsolat biztosításáért, továbbá a program futtatásáért.

Minden egyes *Architektúra*-val kapcsolatban definiáljunk egy *Nyelv* *Architektúra* *véges programozási nyelvet*, mely gyakorlatilag az architektúra által futtatható összes  $\text{Program} \in \text{Nyelv}_{\text{Architektúra}}$  *agens-program* halmaza. Az ágens-programok az ágens aktuális *Benső\_Állapota* és észlelése alapján állítják elő az ágens cselekvéseit, illetve újabb benső állapotát:

**Definíció 3.**

Az ágens architektúrája egy fix interpreter (értelmező-program) az ágens-programok számára, amely futtatja a programot, frissíti az ágens benső állapotát és generálja cselekvéseit:

$$\text{Architektúra}(\text{Program}, \text{Benső\_Állapot}, \text{Észlelés}) = (\text{Benső\_Állapot}, \text{Cselekvés})$$

Most pedig már összefüggésbe hozhatjuk az ágens-programokat és az ágens-függvényeket azáltal, hogy definiáljuk az adott architektúrán futtatott ágens-program által implementált  $\text{Ágens}(\text{Program}, \text{Architektúra})$  függvényt. Ez éppen azon  $f_{\text{Ágens}}$  függvény, melyre teljesül, hogy tetszőleges környezet esetén:

$$\text{Hatás}(f_{\text{Ágens}}, \text{Környezet}) = \text{Hatás}(\text{Ágens}(\text{Program}, \text{Architektúra}), \text{Környezet})$$

Látható, hogy amíg minden architektúrának és programnak megfeleltethető egy-egy ágens-függvény, addig – adott architektúra esetén – már nem minden ágens-függvény implementálható feltétlen. Érdekes tehát bevezetnünk az adott architektúrán megvalósítható függvények halmazát:

$$\text{Megvalósítható}(\text{Architektúra}) = \{f \mid \exists \text{Program} \in \text{Nyelv}_{\text{Architektúra}}, f = \text{Ágens}(\text{Program}, \text{Architektúra})\}$$

Most már tehát rendelkezésünkre áll az ágensek és környezetük teljes definíciója, ám az ágensek által megoldandó problémáról még nem ejtettünk szót. Ez azért szükséges, mert csakis a megoldandó probléma fényében áll módunkban nyilatkozni az ágensek racionalitásáról.

Problémának nevezzük az ágens-környezet kiinduló állapotának, cél-állapotainak és lehetséges cselekvéseinek együttesét. Ekkor a probléma megoldása a cselekvések egy olyan sorozata, amelyek végrehajtása a kiinduló állapotból a cél-állapotok valamelyikébe vezet. Az aktuális cselekvés kiválasztása ezért legtöbbször bizonyos szintű „előrelátást” kíván az ágens részéről, aminek során különböző cselekvés-sorozatokat mérlegel, hogy kiválaszthassa közülük a céljainak legmegfelelőbbet. Ezt a folyamatot nevezzük *tervkészítésnek*. *Tervnek* nevezzük ekkor a lépések egy halmazát és a rajtuk értelmezett *kényszerek* és *relációk* összességét. Tehát a terv nem feltétlen azonos annak végrehajtott változatával, hiszen, egyrészt a terv lépései nem feltétlen azonosak a cselekvésekkel; másrészt, amíg a terv tartalmazhat feltételes elemeket is, addig a *végrehajtott terv* már csupán egy cselekvéssorozat, mely szerencsés esetben a környezet kiinduló állapotából a cél-állapotok valamelyikébe vezet.

A cél-állapotok megadása lehet explicit, vagy implicit: Explicit akkor, ha a cél-állapotok halmaza egyértelműen adott, és implicit akkor, ha nem. Az utóbbi esetben többnyire egy *cél-függvény* segítségével dönthetjük el, hogy az adott állapot része-e a cél-állapotok halmazának. A problémákat a következőképp osztályozhatjuk:

- **Egyállapotú problémák** azok, amelyek olyan hozzáférhető (kvázi determinisztikus) környezetet írnak le, ahol az egyes cselekvések kimenetele az ágens számára teljes egészében ismert. Ilyen problémákkal kapcsolatos az *egyállapotú tervkészítés*.

- **Többállapotú problémák** azok, amelyek olyan, nem hozzáférhető (kvázi nem-determinisztikus) környezetet írnak le, ahol az egyes cselekvések lehetséges kimenetelei az ágens számára teljes egészében ismertek. Ilyen problémákkal kapcsolatos a *többállapotú tervkészítés*.

- **Eshetőségi problémák** azok, amelyek olyan, nem hozzáférhető (kvázi nem-determinisztikus) környezetet írnak le, ahol az egyes cselekvések lehetséges kimenetelei az ágens számára nem egészen ismertek. Ilyen problémákkal kapcsolatos az *eshetőségi tervkészítés*.

- **Felderíthetőségi problémák** azok, amelyek olyan, nem hozzáférhető (kvázi nem-determinisztikus) környezetet írnak le, ahol az egyes cselekvések lehetséges kimenetelei az ágens számára (kezdetben) egyáltalán nem ismertek. Ilyen problémákkal kapcsolatos a *felderíthetőségi tervkészítés*.

Definiáljuk tehát az ágensek által megoldandó problémát implicite: adjuk meg az ágensek adott környezetben vett hasznát, s e haszon maximalizálását tekintjük a probléma megoldásának. Vezessünk be egy valós, lehetséges állapot-történetek felett értelmezett  $U$  haszon-függvényt, aminek maximalizálása megfelel a haszon-függvény és a környezet együttese, egyszóval a *probléma-környezet* által reprezentált probléma megoldásának. Emlékezzünk vissza, hogy az  $f_{\text{Ágens}}$  függvény által adott *Környezet*-ben generált állapot-történetet  $\text{Hatás}(f_{\text{Ágens}}, \text{Környezet})$  jelölte. Ennek alapján tehát  $f_{\text{Ágens}}$  hasznossága a *Környezet*-ben:

$$V(f_{\text{Ágens}}, \text{Környezet}) = U(\text{Hatás}(f_{\text{Ágens}}, \text{Környezet}))$$

Hasonlóan, adott *Környezet*-ben, adott *Architektúrán* futtatott *Program* hasznossága az általa implementált  $\text{Ágens}(\text{Program}, \text{Architektúra})$  függvény hasznossága:

$$V(\text{Program}, \text{Architektúra}, \text{Környezet}) = V(\text{Ágens}(\text{Program}, \text{Architektúra}), \text{Környezet})$$

Az ágens (avagy az intelligens rendszer) tervezőjének most már csak az a feladata, hogy – a probléma ismeretében – valamilyen elvnek, elvárásnak, követelménynek megfelelően megtervezze ágensét. Ez az elvárás lehet a tökéletes racionalitás. Ekkor

**Definíció 4.**

Adott probléma-környezetben tökéletesen racionális ágens  $f_{\text{opt}}$  ágens-függvényére teljesül, hogy

$$f_{\text{opt}} = \arg \max_f V(f, \text{Környezet})$$

A kérdés már csak az; vajon nem-triviális környezetek esetén is implementálható-e ez a tökéletesen racionális ágens-függvényű Adott *Architektúra* esetén könnyen elképzelhető, hogy

$$f_{\text{opt}} \notin \text{Megvalósítható}(\text{Architektúra}).$$

Ekkor tehát az ágens-függvény nem implementálható. De létezhet-e olyan valóságos *Architektúra*, amelyen  $f_{opt}$  implementálható?

Ha létezne ilyen architektúra, akkor a rajta futó tökéletesen racionális ágens-program, annak érdekében, hogy a következő pillanatban beérkező érzet hatására azon nyomban az optimális cselekvést tudja produkálni, vagy már az érzet beérkezése előtt ki kellene, hogy számítsa az optimális cselekvést, vagy az érzet beérkezésekor nyomban. Az utóbbi eset kizárt, hiszen a számítás a valóságban időt igényel. Az előbbi esetben pedig az ágens megtervezésekor úgy kellene kialakítani programját, hogy már előre tudja a jövőben bekövetkező eseményeket (érzeteket). Ez azonban akauzális (a jövő ismeretét igénylő), nem-triviális környezetek esetén nem elvárható tervezési szempont. Viszont a korlátos optimalitás már reális elvárásokat támaszt az ágens tervezőjével szemben:

**Definíció 5.**

*Adott probléma-környezetben, adott Architektúra-val rendelkező korlátosan optimális ágens Program-jára teljesül, hogy*

$$Program = \arg \max_{Prg \in Nye \setminus Architektúra} V (Prg, Architektúra, Környezet)$$

A korlátos optimalitás tehát már nem az ágens-függvényekre, vagy számításokra, esetleg cselekvésekre tesz megkötést, hanem adott ágens-architektúra mellett az ágens-programokra, s így definíciójából következően *megvalósítható* elvárás az ágenssel (és tervezőjével) szemben. Ha tehát a korlátos optimalitást választjuk a racionalitás mércéjéül, akkor lényegében azt várjuk el, hogy rendszerünk, amennyiben racionális, úgy *képességeihez mérten* a lehető legjobban viselkedjen.

**4. Játék-elméleti racionalitás**

Az előző fejezetben – a racionalitás fogalmának tárgyalásakor – mindvégig csak egy szereplőre (például ágensre, rendszerre) szorítkoztunk, s csak közvetve (pl. az ágens környezetének kimondatlan részeként) érintettünk másokat. A játék-elmélet viszont már közvetlenül is foglalkozik a több szereplő közt létrejövő stratégiai kölcsönhatásokkal. Magyarán a játék-elmélet az ésszerű (racionális) viselkedés elmélete olyan szituációkban, melyekben minden szereplőnek a többi szereplő részéről várható ellenlépések fényében kell meghoznia döntését.

A játék-elmélet a vizsgált szereplőket nem ágenseknek, vagy rendszereknek, hanem játékosoknak tekinti. Számos kiváló matematikus és közgazdász kezdeti munkái után a játék-elmélet, mint rendszeres tudományos elmélet 1944-ben indult útjára [1]. Eredeti célkitűzése a racionális visel-

kedés meghatározása – a való életből vett – gazdasági, politikai és társadalmi helyzetekben.

**4.1. Játék-elméleti alapfogalmak**

A legegyszerűbb – többszereplős döntési problémának megfelelő – játék az, amelyben két játékos két-két stratégiával rendelkezik, s ezek közt kell egyszerre, egymástól függetlenül választania. Egyik játékos számára se ismert, hogy a másik játékos éppen milyen stratégiát választ, viszont mindkettejük haszna (nyeresége, profitja, kifizetése) függ a másik választásától is. Az ilyen kétszereplős játékokat célszerű bimátrix alakban [8] ábrázolni. A 2. ábra néhány bimátrix játékra mutat példát.

Kezdetnek vizsgáljuk az 1)-es játékot. Ha az 1-es játékos például az  $s_1$  stratégiát játssza, míg a 2-es játékos az  $s_2$ -t, tömören fogalmazva, ha az  $(s_1, s_2)$  stratégia-kombinációt játsszák, akkor az 1-es játékos haszna  $a$ , míg a 2-es játékos haszna  $d$  (lásd a 2. ábrán a második sor harmadik oszlopa), avagy a játékosok haszna  $(a, d)$ . Látható, hogy az 1-es játékos  $s_1$  stratégiája a másik játékos tetszőleges stratégia-választása esetén jobb eredményt ad, mint az  $s_2$  stratégia, hiszen  $c > d$  és  $a > b$ . Ekkor azt mondjuk, hogy az 1-es játékos esetén az  $s_1$  stratégia *dominálja* az  $s_2$  stratégiát. Hasonló a helyzet a 2-es játékos esetén is. Az olyan stratégiákat, amelyek minden más stratégiát dominálnak egy adott játékos esetén, a játékos *domináns stratégiájának* nevezzük. Mivel az 1)-es játékban mindkét játékosnak csupán két-két stratégiája van, ezért mindkettőjük domináns stratégiája  $s_1$ .

Ésszerűnek tűnhetne azt mondani, hogy a játékosok mindig a domináns stratégiájukat válasszák, függetlenül attól, hogy a többiek mit döntenek, hiszen másként csak rosszabbul járnának, akármit is döntenek a többiek. Sajnos azonban a legtöbb játékban nincs domináns stratégia (lásd például 2. ábra, 3)-as játék). Mivel tehát a domináns-stratégia választás általában nem elvárható, ezért a játék-elméletben egy ennél valamivel gyengébb elvárást, az egyensúlyra való törekvést szokás racionálisnak tekinteni.

A játék-elmélet központi egyensúly-fogalma a Nash-egyensúly [9]. Tömören fogalmazva: *Nash-egyensúlynak* nevezzük azt a stratégia-kombinációt, amely esetén egyik játékosnak se érné meg egyedülálló módon stratégiát váltania.

2. ábra  
Az 1-es és a 2-es játékosok az  $s_1$  és  $s_2$  stratégiák közül választhatnak, melyeknek függvényében hasznuk  $a, b, c$ , vagy  $d$  lehet, ahol most  $a > b > c > d$ .

	2	$s_1$	$s_2$
1	$c$	$d$	$b$
$s_1$	$c$	$a$	$c$
$s_2$	$d$	$a$	$a$

	2	$s_1$	$s_2$
1	$c$	$b$	$a$
$s_1$	$c$	$c$	$c$
$s_2$	$a$	$c$	$c$

	2	$s_1$	$s_2$
1	$b$	$a$	$b$
$s_1$	$a$	$b$	$a$
$s_2$	$b$	$a$	$a$

A Nash-egyensúlyban tehát minden játékos stratégiája legjobb válasz a többiek stratégiájára, s így senkinek se érné meg egyedülálló módon eltérnie az egyensúly által előírt stratégiától. A Nash-egyensúly tehát bizonyos értelemben „önbeteljesítő”. Viszont több probléma is felmerül vele kapcsolatban.

Egyrészt a Nash-egyensúly *nem kooperatív* megfontolás, hiszen, ha több játékos is stratégiát válthatna egyszerre (például közös megegyezés, kommunikáció, bizalom alapján, koalíciókba szerveződve), pontosabban, ha a játékosok valamely, egynél több főből álló csoportja együttműködésben (kooperálva) alakíthatná ki döntéseit, úgy értelmetlenné válna Nash-egyensúlyi stratégiát játszani, hiszen semmit se garantálna.

Ráadásul a Nash-egyensúlyi stratégia még nem kooperatív esetben se jelent feltétlen garanciát, hiszen bizonyos játékosok (akár csak egyedülálló, irracionális módon) eltérhetnek Nash-egyensúlyi stratégiájuktól, s így nem csak ők járnak rosszabbul, hanem esetleg azok is, akik racionálisan a számukra előírt Nash-egyensúlyi stratégiát követték. Az ilyen eseteket *hiteltelen fenyegetésnek* nevezik, hiszen az a játékos, amelyik eltér egyensúlyi stratégiájától, saját magának is árt, ami ugyebár neki sem lehet ésszerű érdeke. A hiteltelen fenyegetéseket viszont kiküszöböli a Nash-egyensúly egy finomítása: az *aljáték-tökéletes egyensúly* [10].

A 2. ábra 1)-es játékban jól láthatóan csak egyetlen Nash-egyensúly van: az  $(s_1, s_1)$  stratégia-kombináció. Ez az egyetlen olyan stratégia-kombináció tehát, amely esetén egyik játékosnak se érné meg külön-külön más stratégiát választania. Viszont, ha a játékosok az  $(s_2, s_2)$  stratégia-kombinációt játszanák, hasznuk egyaránt magasabb lenne, hiszen  $b > c$ . Ebből következik, hogy a Nash-egyensúly nem feltétlen „optimális”. Az  $(s_2, s_2)$  stratégia-kombináció az 1)-es játék *Pareto-optimuma*, mivel nincs más olyan stratégia-kombináció, amely esetén a játékosok valamely részhalma jobban jár, míg a többiek nem járnak rosszabbul. Tehát a Nash-egyensúly bizonyos esetekben (például Pareto-értelemben) *szub-optimálisnak* adódik.

A 2. ábra 2)-es játékban szemmel láthatóan több, pontosan három különböző Nash-egyensúly is adódik: az  $(s_2, s_1)$ ,  $(s_1, s_2)$  és a szub-optimális  $(s_2, s_2)$  stratégia-kombináció. A Nash-egyensúly tehát *nem feltétlen egyértelmű*. Felmerül a kérdés, hogy egy „racionális” játékos melyiket válassza, illetve, hogy vajon a többi játékos is azt választja-e majd? Az így adódó *egyensúlyválasztási problémának* egész irodalma van (például [11]). A játék valamely adott egyensúlyválasztási elv mellett adódó Nash-egyensúlyát a *játék fókusz-pontjának* nevezzük.

A 2. ábra 3)-as játékának pedig *nincs Nash-egyensúly*, hiszen bármely stratégia-kombináció esetén létezik olyan játékos, amelynek megérné más stratégiát választania. Ezt a problémát viszont feloldja a *kevert stratégiák*, s így a *kevert Nash-egyensúly* fogalmának bevezetése. Az eddigiekben tehát mindvégig olyan *tiszta stratégiákról* beszéltünk, melyeket a játékosok 1-valószínűséggel követtek. Ha viszont megengedjük, hogy

a játékosok egynél kisebb valószínűséggel is választhassák stratégiájukat, úgy a kevert stratégiák fogalmához jutunk.

Látható, hogy a 2. ábra 3)-as játékának, bár tiszta Nash-egyensúlya nincs, van viszont (egyértelmű) kevert Nash-egyensúlya. Ha mindkét játékos 1/2 valószínűséggel választja mind az  $s_1$ , mind az  $s_2$  stratégiát, azaz, ha mindketten az  $(1/2, 1/2)$  kevert stratégiát, avagy *lutrit* játsszák, akkor egyiküknek se áll érdekében ezen változtatni. A kevert stratégiák hasznát a játékosok hasznának várható értékeként számítjuk. Egy-egy kevert stratégia-kombináció játszásakor azonban csak valamely tiszta stratégia-kombináció kerül lejátszásra.

A Nash-egyensúly létezésére vonatkozó tétel [9] – nem túl szigorú megkötéseihez köszönhetően – szinte minden játékban garantál legalább egy kevert Nash-egyensúlyt. Így tehát a kevert stratégiák bevezetésével megoldódik az egyensúly hiányának problémája. Sajnos azonban a kevert Nash-egyensúllyal kapcsolatban is fennáll a tiszta Nash-egyensúly kapcsán említett legtöbb probléma:

(1) Kooperatív játékosok esetén nem alkalmas racionalitás definíció. Természetesen a kooperatív játék-elmélet jónéhány alternatív egyensúly-definíciót javasolt (például [12]), ám ezek közös jellemzője, hogy csak igen szigorú, speciális feltételek mellett garantálható a létezésük.

(2) Mivel a kevert Nash-egyensúlyból is lehet több, ezért továbbra is fennáll az egyensúlyválasztás problémája.

(3) A kevert Nash-egyensúly is lehet szub-optimális.

#### 4.2. Ágens- és játék-elmélet összekapcsolása

Jól látható, hogy mindeddig csak olyan racionalitás fogalmakat érintettünk, melyek a játékosok stratégiáira, avagy stratégia-kombinációkra vonatkoztak. Ennek oka egész egyszerűen az, hogy a játék-elméletben ez a szemlélet egyeduralgó. Viszont, ha megfontoljuk az ágens-elméleti korlátos optimalitás megközelítését, mely szerint nem döntésekre, hanem döntéshozási mechanizmusokra vonatkoztatjuk a racionalitást, akkor kiküszöbölhetővé válik az előbb felsorolt problémák jelentős része.

Tegyük tehát megkötést a játékosokra, pontosabban a játékosok stratégia-választási mechanizmusára, ne pedig az általuk választott stratégiákra. Érzékeltetésképp gondoljunk csak bele abba, hogy miképpen alakul a 2. ábra 1)-es játéka akkor, ha az egyes játékosok úgy választják meg stratégiájukat, hogy közben azt az elvet követik, mely szerint „azt teszik másnak, amit maguknak is kívánnak”.

Mivel a racionális játékosok végeredményben hasznuk növelésére törekszenek, s az 1)-es játékban az  $(s_2, s_2)$  stratégia-kombináció adja a legmagasabb hasznát mindkét játékos számára, ezért a fenti stratégia-választási elvet követve, mindketten az  $s_2$  stratégiát választanák. Ha tehát a játékosok döntéshozási mechanizmusa a fenti elv szerint működik, akkor a 2. ábra 1)-

es játékában optimálisan cselekszenek. Természetesen ez az elv nem minden játék esetén garantál optimális kimenetelt. Viszont a példa érzékelteti, hogy a játékosok stratégia-választási mechanizmusára tett megkötések képesek lehetnek feloldani a stratégiákra tett racionalitási megkötések szub-optimalitását. Ez tehát jelentős érv mellett, hogy – hasonlóan az ágens-elméleti korlátos optimalitás fogalmához – a játékosok stratégia-választási mechanizmusára vonatkoztassuk a racionalitást, ne pedig a választott stratégiákra.

Ha meggondoljuk, akkor már a többszörös Nash-egyensúlyok okán felmerülő egyensúlyválasztási elv kijelölése is egy implicit megkötés a játékosok stratégia-választási mechanizmusára. Viszont esetünkben nem implicit, hanem explicit megkötésekre van szükség a játékosokkal kapcsolatban ahhoz, hogy – hasonlóan a 3. fejezetben tárgyaltakhoz – elvárható racionalitási követelményeket támaszthassunk velük szemben.

Egyesítsük tehát az ágens és játék-elméletet, s vezessük be az ágensek kapcsán tárgyalt főbb fogalmakat: tekintsük a játékosokat ágensnek. A játékosok stratégiáit terveknek. Az egyes játékosok hasznát tekintjük problémát definiáló haszon-függvénynek (így mindegyik játékos más-más problémával állhat szemben). Minden játékosnak legyen egy-egy architektúrája, mely adott program-nyelv programjait futtatja. Az ágens-programoknak a játékosok úgynevezett *Típus*-át feleltessük meg, s így az ágensek program-nyelvét, azaz a lehetséges ágens-programok halmazát a játékosok lehetséges típus-halmazának tekintsük (*Típusok*).

A játékosok típusainak bevezetésére eredetileg azért volt szükség, hogy a nem teljes információjú játékokat vissza lehessen vezetni teljes, de nem tökéletes információjú játékokra [13].

Ez lényegében kétfajta megkülönböztetést jelent. *Teljes információjú játékok* esetén a játékosok teljes körű információval rendelkeznek a játék – normál (vagy extenzív) formájával meghatározott – alapvető matematikai szerkezetéről. *Nem teljes információjú játékok* esetén ez nem teljesül. Tehát itt aszerint osztályozzuk a játékokat, hogy a játékosok milyen mértékben tájékozottak a játék olyan jellemzőiről, amelyek már a *lejátszás előtt* adottak (játékosok száma, lehetséges stratégiái, haszon-függvényei stb.) *Tökéletes információjú játékok* alatt pedig olyan játékokat értünk, amelyekben a játékosok teljes körű információval rendelkeznek a *lejátszás során* addig bekövetkezett lépésekkel kapcsolatban. *Nem tökéletes információjú játékok* esetén ez nem igaz. Tehát ebben az esetben aszerint osztályozunk, hogy a játékosok milyen mértékben tájékozottak a *lejátszás során* bekövetkezett eseményekkel kapcsolatban.

A következőkben tegyük fel, hogy a játékos (vagy tervezője) előtt nem pontosan ismert az aktuális játék, viszont minden játékos pontosan tudja, hogy melyik típus képviseli a játékban. Álljon fenn továbbá a közös apriori becslések (common priors assumption) feltevése is [14], miszerint minden játékos ismeri a lehetséges típus-kombinációk apriori valószínűségeit, s ennek ismeretével kölcsönösen is tisztában vannak. Namármost tegyük fel, hogy minden játékos pontosan tisztában van a játék matematika szerkezetével, s ismerik egymás architektúráját.

retével kölcsönösen is tisztában vannak. Namármost tegyük fel, hogy minden játékos pontosan tisztában van a játék matematika szerkezetével, s ismerik egymás architektúráját.

Ekkor már csak az a kérdés, hogy a többi játékosnak mi a típusa – ez jelenti az egyedüli bizonytalanságot. Tehát mindennemű – játékkal kapcsolatos – bizonytalanságot visszavezettünk egy olyan nem tökéletes információjú játékra, melyben csupán a többi játékos típusa nem ismert az egyes játékosok előtt. Ha tehát egy adott játékos haszna egy *Játék*-ban  $V$  (*Típus*, *Játék*), ahol a *Játék* része a többi játékos típusa és az adott játékos *Típus* képviseli, akkor

#### Definíció 6.

*Adott játékban korlátosan optimális játékos Típusára teljesül, hogy*

$$Típus = \arg \max_{Típ \in Típusok} V(Típ, Játék)$$

Tehát egy adott architektúrájú, korlátosan optimális játékosnak olyan a programja, amely adott játék (s így a többi játékos adott típusa) esetén maximalja a játékos várható hasznát. Tehát immár a játék-elméletben is rendelkezésünkre áll egy, a mesterséges intelligencia racionalitás-definíciójával analóg, gyakorlatban is használható racionalitás definíció. Ezen felül, mivel a játékosokat ágenseknek, stratégiáikat pedig terveknek tekintettük, immár lehetőségünk nyílik a játékelméleten belül is felhasználni a tervekészítés elméletének gazdag eredményeit. A játék-elmélet számára tehát előnyös az ágens-elmélettel való összekapcsolás.

Az ágens-elmélet szemszögéből se elhanyagolható a nyereség, amit a játék-elmélettel való összekapcsolás eredményez. Mivel a játék-elméleten belül a típus-reprezentáció módot ad tetszőleges nem teljes információjú játék teljes, ám nem tökéletes információjú játékká alakítására, ezért lényegében a fenti megfontolások a (játékos)ágensek összes bizonytalanságát képesek visszavezetni az előzőleg bekövetkezett események hiányos ismeretére. Ezáltal formálisan is kezelhetővé válik az eshetőségi és felderíthetőségi tervekészítés (lásd 3.), ami mind a mai napig megoldatlan feladat. Ráadásul a multi-ágens rendszerekben fellépő társadalmi jelenségek is kiforrott, következetes formális reprezentációt nyerhetnek a játék-elmélet bevezetésével.

A játék-elmélet főbb fogalmai az ágens-közösségekben belül is leírhatóvá válnak (például egyensúly, kooperáció), hiszen a 3. fejezet nem tett megkötést arra vonatkozólag, hogy az ágens környezete mit tartalmaz. Így módon akár más ágenseket is tartalmazhat, melyeket immár játék-elméleti eszközökkel modellezhetünk. Az elméletek összekapcsolása tehát nagyban megnöveli leíró-erejük, lehetővé téve a körültekintő, választható tervezési szempontok érvényesítését. Egységes módon válik tervezhető az egyéni és a csoportos működés, hiszen amíg az ágens-elméleti eszköztár az egyén működésére koncentrál, addig a játék-elmélet az egyén csoportviszonylatban vett működésére.

## 5. Evolúciós-elméleti racionalitás

Az előző fejezeteket a korlátos optimalitás tetőzte be, ám arról egyikben sem ejtettünk szót, hogy a korlátosan optimális megoldás (program, stratégia stb.) megtalálása miképpen lehetséges. Az optimális megoldás keresésére számos módszer ismeretes, melyek közt a természetes evolúció számítási modellje rangos helyet foglal el.

Az evolúciós számítási modellek három fajtáját szokták megkülönböztetni aszerint, hogy miként interpretálják a darwini posztulátumot, miszerint a közös erőforrásokért versengő populáció azon egyedei kerülnek előnyösebb helyzetbe, melyek a versenyben előnyt jelentő (racionális) tulajdonságokat hordoznak. A legmagasabb szintű megközelítés *evolúciós programozás* [15,16] néven vált ismertté, és a *fajok* közti versengést modellezi. Köztes szinten mozognak az *evolúciós stratégiák* [17], melyek az *egyed* szintjén vizsgálják a természetes kiválasztódás darwini folyamatát. A legelemibb megközelítésnek a *genetikus algoritmusok* [18] családjába tartozó módszerek tekinthetők, melyek a *gének* közti versengés szintjén követik nyomon az evolúció folyamatát. Ebben a fejezetben ez utóbbi módszerekkel foglalkozunk, illetve ezeknek az ágens- és játékelmélettel való kapcsolatukkal.

### 5.1. Genetikus Algoritmusok

A genetikus algoritmusok – és változatai – *globálisan konvergens*, sztochasztikus keresőeljárások [19]. Ez azt jelenti, hogy – a véletlenszerűség felhasználásával – végzett (nem randomizált) keresés során bizonyítottan tartanak a keresési tér – jósági mérce által definiált – *globális optimumához*.

Az evolált *egyedeket* klasszikus esetben bináris bit-füzérek (kromoszómák, avagy génláncok) reprezentálják, melyek pozícióinak (alléljainak) bit-értéke felel meg a géneknek. Az algoritmus lényege, hogy egyedek egy kezdeti, véletlenszerűen előállított *populációját* – úgynevezett *genetikus operátorok* felhasználásával – addig-addig módosítja, mígnem a populáció megfelelő jellemzői (például legjobb egyed, egyedek átlagos jósága) eleget nem tesznek egy választott *leállási kritériumnak*. Az egyedek jóságát megbecsülő jósági mércét szokás *fitness függvénynek* is nevezni. A fitness függvény a populáció egyedeihez rendel egy-egy valós szám-értéket, mely az adott egyed globális (a populáció viszonylatában vett) *jóságát* szimbolizálja. Alapesetben a következő genetikus operátorokat különböztetjük meg:

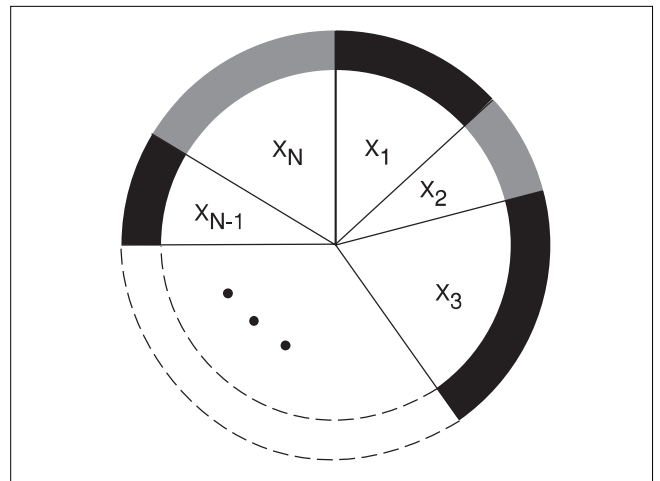
- A *mutáció* adott valószínűséggel, taláalomra módosítja a génlánc (pl. bitfűzér) alléljain található gének (pl. bitek) értékét (pl. invertálja a bitet).
- A *keresztelés* valószínűséggel kiválaszt egy-egy génláncot, majd megcseréli egy véletlenszerűen választott pozíció után következő részeit.
- A *szelekció* a populáció egyedeinek jóság szerint történő, véletlenszerű kiválasztása és átörökítése a következő generációba.

A fenti operátorok felhasználásával az algoritmus menete a következő: kezdetben előállítjuk egyedek egy véletlenszerű populációját. Az egyedek hossza, a populáció mérete, és az operátorok valószínűsége adott. Ezt követően minden egyedhez rendelünk egy-egy jósági értéket, mely alapján kiválasztunk közülük egy – a szelekció típusától függő – mennyiséget. A kiválasztott egyedeket párokba rendezzük. A párokat (szülőket) kereszteljük, majd mutáljuk. Így kapjuk a *következő generáció* egyedeit (a gyerekeket). Ezt ismételjük generációról generációra mindaddig, míg nem teljesül a leállási kritérium.

A fentebb említett három alapvető genetikus operátor megvalósítására több lehetőség is kínálkozik. Ezek közül a fontosabbak: A szelekció *rulett kerék* elven működik, ha az egyedeket jósági értékükkel arányos valószínűséggel választja ki a következő generációba. Tegyük fel, hogy a populáció  $n$  egyedből áll. Legyen a populáció valamely  $x_i$  egyedének jósága  $V(x_i)$  minden  $i=1,2,\dots,n$ -re. Ekkor (képzeletben) osszunk fel egy egy-ségnyi kerületű „rulett-kereket”, avagy kört  $n$  darab

$$V(x_i) / \sum_{j=1}^N V(x_j)$$

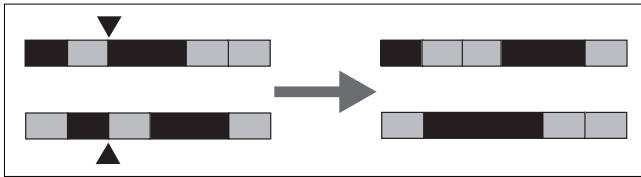
ívhosszú körcikkre, majd „perdítsük meg a kereket”, azaz állítsunk elő egy  $[0, 1)$  tartományba eső, valós véletlen-számot. Ahol a „golyó megáll”, azaz amely egyednek megfelelő körcikk íve által meghatározott  $[0, 1)$ -beli zárt rész-intervallumba esik az előállított véletlen-szám, azon egyed kerül kiválasztásra. A 3. ábra szemlélteti a rulett-kerék elven alapuló szelekciót.



3. ábra Rulett-kerék elven alapuló szelekció

A *mérkőzés* alapú szelekció is rulett-kerék elven működik:  $K$ -szor „megpörgetjük a rulett-kereket”, majd az így kiválasztott  $K$  egyed legjobbjával térünk vissza. Az új generáció előállításához ezt ismételjük  $N$ -szer. A *felső százalék* elve szerint a populáció egyedeinek legjobb  $N$  százalékából választjuk ki véletlenszerűen a következő generáció egyedeit. A *valahány legjobb* kiválasztás lényege, hogy az egyedeket jóság szerint sorba rendezzük és az első valahányat választjuk ki közülük.

A keresztezésnek is több módja lehetséges. Létezik *egy-pontú* és *több-pontú keresztezés*. A több-pontú keresztezés az egy-pontú keresztezés értelemszerű kiterjesztése. Egy-pontú keresztezésre mutat példát a 4. ábra.



4. ábra Egy-pontú keresztezés

A mutációs operátor az egyed valamely véletlenszerűen választott alléljának génjét változtatja meg taláalomra, így – definit megfogalmazásának köszönhetően – nincsenek változatai. Más struktúrájú gének (például nem bitfüzerek, hanem programfák [4]) esetén nyilván más módon kell implementálni, ám elve változatlan marad.

A fentebb említett alap-operátorokon túl természetesen még sok egyéb operátor lehetséges, melyek közül talán még az *elitizmust* említenénk. Az elitizmus a szelekció egy változata, mely – akár a szelekcióval együttműködésben – egy adott generáció valahány legjobb egyedét örökíti át változtatás nélkül a következő generációba. Újabban bizonyítást nyert, hogy a genetikus algoritmusok általános keresési terekben is globálisan optimálisak, ha használják az *elitizmust*, továbbá a populáció minden egyes egyede *elérhető* bármely más egyedből kereszteződés és mutáció révén [20].

## 5.2. Játék, ágens és evolúciós-elmélet összekapcsolása

A játék- és ágens-elmélet összekapcsolásának lehetőségeiről már szót ejtettünk (lásd 4.2). Az evolúciós- és játék-elmélet összekapcsolása is megtörtént már. Az *evolúciós játék-elméletben* [21] jellemzően játékosokból alkotott populációk dinamikáját vizsgálják. Tehát a populáció egyedeit tekintik játékosnak, és a fitness függvény a játékosok haszon-függvénye.

A populációs dinamikák leírására legtöbbször a diszkrét dinamikus rendszerek (DDR) módszertanát használják. Az evolúciós játékelmélet központi racionalitás-fogalmának tekinthető *evolúciósan stabil stratégiákat* (ESS) [22] is ily módon származtatják, mint a populációs dinamika bizonyos „fix-pontjait”.

Tömören fogalmazva, ha a populáció egyedei evolúciósan stabil stratégiát játszanak, úgy a populáció nem változik, s ha bárki is eltér a számára előírt ESS stratégiától, csak rosszabbul járhat (mind haszon, mind pedig túlélési esélyek tekintetében). Az ESS stratégiák tehát legtöbbször megegyeznek a Nash-egyensúlyi stratégiákkal, aminek következtében gyengeségeiket is öröklék (lásd 4.1). Ezért tehát, hasonlóan a játék-elmülethez, más racionalitás definíciót érdemes keresnünk.

Ahhoz, hogy *használható* racionalitás-definícióhoz jussunk az evolúciós-elméleten belül, megoldást jelent-

het az ágens-elmélettel való összekapcsolás [23]. Nem elég tehát csupán annak a kijelentése, hogy „az racionalis, ami a túléléshez kell”, hanem arra is választ kell tudnunk adni, hogy „mi alapján dől el egy egyed, illetve egyedek egy csoportjának a túlélése – mi határozza meg jószágukat”.

A két elmélet összekapcsolása során tehát az ágenseket a populáció egyedeinek, az – ágensek környezetében fennálló problémát reprezentáló – haszon-függvényt pedig az egyedek jószágát meghatározó fitness függvénynek célszerű tekintenünk. Ekkor lényegében az ágensek probléma-környezete (lásd 3. fejezet) egy evolúciós optimalizálási problémának felel meg. Hasonlóan, bármely evolúciós optimalizálási probléma triviálisan átfogalmazható ágensek egy megfelelő probléma-környezetévé.

Az ágens- és evolúciós elmélet összekapcsolása az ágens-elmélet szempontjából előnyös, hisz lehetőséget kínál az „emergent”-jellegű, és más egyéb populációs jelenségek evolúciós vizsgálatára. Az evolúciós-elmélet szemszögéből nézve is előnyösnek mondható a két elmélet összekapcsolása, hisz lehetőség nyílik egyrészt az egyedek ágens-elméleti vizsgálatára, mely során az egyedek felépítésébe és működésébe is bepillantást nyerhetünk, másrészt használható módon definiálható az egyedek racionalitása (például korlátosan optimális egyed fogalma).

Összességében az evolúciós-, játék- és ágens-elmélet összekapcsolása lehetőséget teremt evolúciós eszközökkel optimalizált, ágens-elméleti léptékekkel tárgyalható bonyolultságú komplex-rendszerek játék-elméleti eszközökkel történő vizsgálatára. Végeredményben a három elmélet összekapcsolásával egy – az eddigiekhez képest – átfogó racionalitási elv megfogalmazása válik lehetségessé, mely alapjául szolgálhat az intelligens rendszerek egységes elven történő tervezésének és elemzésének.

## 6. Összefoglalás

Három főbb témakört érintettünk a cikk során: az ágens-, játék- és evolúciós-elméleteket. Mindhárom esetben a racionalitás fogalmát használtuk az egységes tárgyalás alapjául, hogy rámutassunk az elméletek összekapcsolási lehetőségeire. A három elmélet összekapcsolási lehetőségeinek felvázolásával egy átfogó rendszer-specifikációs elvet körvonalaztunk, amely egységes elméleti alapot teremthet az intelligens rendszerek egységes tervezéséhez és elemzéséhez.

Az elméletek összekapcsolása céljából a mesterséges intelligencia „újkeletű” racionalitás fogalmát, a korlátos optimalitást használtuk. Rendre összevetettük a hagyományos játék-elméleti és evolúciós-elméleti racionalitás-fogalmakkal.

Mindkét esetben arra a megállapításra jutottunk, hogy a korlátos optimalitás előnyösebb, mivel használható, a gyakorlatban is elvárható kritériumokat támaszt a tervezett rendszerrel szemben. Ezzel egy olyan átfo-

gó jósági mércéhez jutottunk, melyel a rendszerek hasznossága egyaránt értelmezhető egyéni, csoportos, illetve csoportközi szinten.

Végeredményben tehát egy olyan, az eddigiekhez képes átfogó elvet körvonalaztunk, mely alkalmas alapját képezheti egy, az intelligens rendszerek egységes tervezésére és elemzésére irányuló elméletnek, áthidalva az intelligens rendszerek tervezésével kapcsolatos eddigi főbb nehézségeket.

### Irodalom

- [1] von Neumann, J., & Morgenstern, O. (1944). *Theory of Games and Economic Behavior*, Princeton University Press: Princeton.
- [2] Good, I. J. (1971). Twenty-seven principles of rationality. In Godambe, V. P., & Sprott D. A. (Eds.), *Foundations of Statistical Inference*, pp.108–141. Holt, Rinehart, Winston: Toronto.
- [3] Simon, H. A. (1982). *Models of bounded rationality*, Volume 2, MIT Press: Cambridge.
- [4] Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press: Cambridge.
- [5] Russell, S. J., & Subramanian, D. (1995). Provably bounded-optimal agents, *Journal of AI Research*, 2:1–36.
- [6] Russell, S. J., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*, Prentice Hall.
- [7] Wooldridge, M. (1999). *Intelligent Agents*, In Weiss, G. (Ed.), *Multiagent Systems*, MIT Press.
- [8] Lója K. (2003). Játékelméleti módszerek, *Híradástechnika*, 4:29–34.
- [9] Nash, J. F. (1951). Non-cooperative games, *Annals of Mathematics*, 54(2):286–295.
- [10] Selten, R. (1975). Reexamination of the Perfectness Concept for Equilibrium Points in Extensive Games, *International Journal of Game Theory*, 4:25–55.
- [11] Harsányi, J. C., & Selten, R. (1988). *General theory of equilibrium selection in games*. MIT Press.
- [12] Aumann, R. (1959). Acceptable Points in General Cooperative n-Person Games, in *Contributions to the Theory of Games IV*, Princeton: Princeton University Press.
- [13] Harsányi, J. C. (1967–1968). Games with incomplete information played by Bayesian players I-II-III, *Management Science*, 14:159–182;320–334;486–502.
- [14] Fudenberg, D., & Tirole, J. (1991). *Game Theory*, Cambridge, MA: Cambridge University Press.
- [15] Fogel, L. J., Owens, A. J., Walsh, M. J. (1966). *Artificial Intelligence through Simulated Evolution*, New York: John Wiley.
- [16] Fogel, D. B. (1995). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, Piscataway, NJ: IEEE Press.
- [17] Schwefel, H. P. (1995). *Evolution and Optimum Seeking*, Sixth-Generation Computer Technology Series, John Wiley & Sons, Inc., New York.
- [18] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley.
- [19] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press.
- [20] Rudolph, G. (1996). Convergence of Evolutionary Algorithms in General Search Spaces, in *Proc. of IEEE International Conference on Evolutionary Computation*, pp.50–54, Nagoya, Japan.
- [21] Weibull, J. W. (1995). *Evolutionary Game Theory*, Cambridge, Mass.: MIT Press.
- [22] Maynard-Smith, J. (1982). *Evolution and the theory of games*, Cambridge University Press: Cambridge.
- [23] Kovács, D. L. (2003). *DINA: Dynamic INtelligent Agents*, Proceedings of IEEE International Symposium on Intelligent Signal Processing, IEEE.

## Hírek

A Sun, amely kezdetben a különösen nagy kapacitású szerverek és munkaállomások területén jelentkezett termékeivel, de már évek óta fejleszti a szabad hozzáférésű Java programokat is, 2004 őszétől újabb területen szeretne sikereket elérni. A **Sun Managed Services** vállalja, hogy karbantartja a vállalatok számítógépparkját, részben pedig preventív szolgáltatásokat igyekszik bevezetni, azaz karbantartást és hibaelhárítást is végez, sőt ahol a Sun gépek más gyártmányú számítógépekkel együttműködésben végzik feladatukat, ott is vállalja a teljes számítástechnikai rendszer felügyeletét és javítását.

A szolgáltatás egyes gépjárműgyártók Assistance-kártyájához hasonlít, amely birtokában a tulajdonos bármilyen műszaki probléma esetén gyors és szakszerű segítséget kap. Ez az eljárás a számítógépek-nél még indokoltabb és gyorsabb lehet, mivel a hibák egy részét a felhasználó telephelye és a szervizelő gépek közötti hálózaton is el tudják hárítani, vagy legalább is felismerik a hiba jellegét és a saját tartalékok felhasználásával gyors segítséget nyújthatnak. Bár a megoldás nem általánosan új, hiszen a digitális alközpontok megjelenésekor az alközpontgyártók már vállaltak ilyet, a Sun azonban reméli, hogy a számítógépes területen távlatilag is sikeres, új üzletágot hozott létre.



# 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 műsorsválaszték 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 műsorszórásban 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 vevőkóderekhez 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 foglalt biztositja 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 Initiative) 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 fejléctet, 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 interoperabilis 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 objektumkarusszel 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 (objektumkarusszelben) 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 vevődekóder számára lehetővé teszi az MHP alkalmazások megtalálását a transzport adatfolyamban. Az AIT-ket 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 objektumkarusszel 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

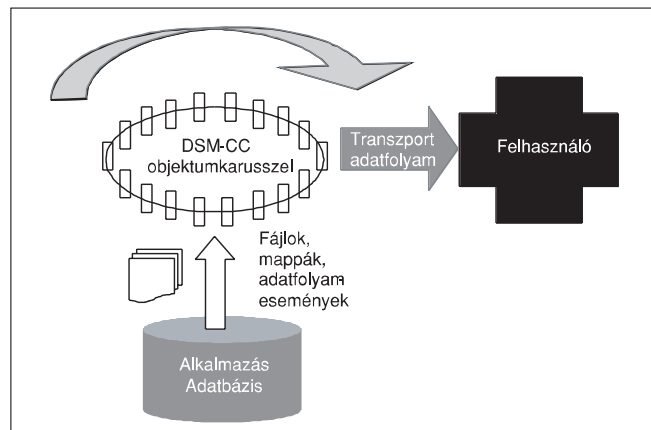
mazás 8 db 100 kb-ot méretű képet és 224 kb-ot Java .class fájlt tartalmaz, akkor a letöltési idő 1024kb-ot/512kbit/s=16 másodperc. Kérdés, hogy a felhasználók mennyi időt hajlandók várni az alkalmazás letöltésére. A tervezőknek ezeket a körülményeket mindenképpen figyelembe kell venniük. Az összefüggésből következik, hogy az objektumkarusszel 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 objektumkarusszelben 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 objektumkarusszel 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 objektumkarusszel első körbefordulásakor az első nagyobb méretű fájl kerül átvitelre a kö-

2. ábra Az adatátviteli rendszer felépítése



vetkező körbefordulásakor 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-

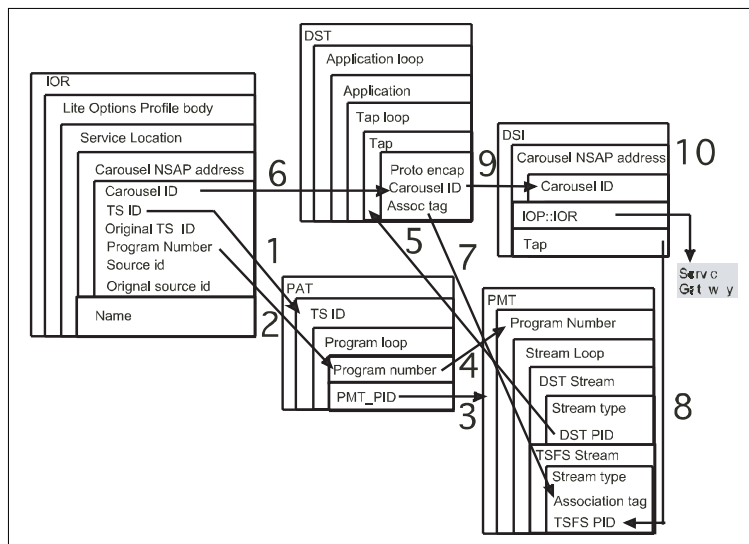
si kapacitása) szegényes voltának tudható be, ami még fontosabb teszi az objektumkarusszal optimalizálását. A fájlrendszer regenerálás komplexitásának bemutatására nézzük a fájlrendszer referencia pontjának (root) kinyeréséhez szükséges műveleteket (3. ábra):

- 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 sorszámának 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 sorszámával.
- 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

# VLAN címke alapú cross-connect funkció a videóhálózatokon

SZEGEDI PÉTER

MATÁV PKI Fejlesztési Intézet  
szegedi.peter3@ln.matav.hu

**Kulcsszavak:** digitális stúdió, Ethernet, NG-SDH, transzport hálózat, címke-kapcsolás

Az utóbbi tíz évben rohamléptekkel fejlődő digitális jelfeldolgozás egyre jelentősebb szerepet tölt be a modern televíziózás területén. A digitalizálás nem csak a végfelhasználók készülékeit érinti, hanem a teljes videóprodukciónak a lánc egészét is. Ahogy a digitális stúdiótechnika egyre nagyobb teret hódít, a műsorszolgáltatók szeretnék tisztán digitális formátumú audió- és videófolyamjaikat csomagkapcsolt transzport hálózatokon továbbítani. A stúdiók közötti digitális programcsere, illetve a stúdiók és a műsorszétesztő pontok közötti csomag-alapú videótranszport igények kiszolgálásához az újgenerációs SDH berendezésekbe integrált Ethernet funkciók kínálnak új lehetőségeket. A következő oldalakon a VLAN (Virtual LAN) címke alapú cross-connect funkció alkalmazásának motivációit és előnyeit igyekszünk bemutatni egy hipotetikus, hazai videótranszport szolgáltatói hálózat architektúrájában.

## 1. Bevezetés

Napjainkban a legtöbb televíziós stúdió már digitális technológiát alkalmaz. A digitális kamerák, vágóasztalok, keverőpultok, rögzítő és archiváló berendezések valamint számtalan egyéb professzionális eszköz lehetővé teszi olyan új audió- és videószerkesztési megoldások alkalmazását, mint a virtuális 3D stúdió, vagy a speciális videó effektek.

A modern stúdiókban helyi hálózatokon (LAN) keresztül jön létre a kapcsolat a digitális stúdió-berendezéseken futó, együttműködő alkalmazások között. Mivel a helyi hálózatok zömében Ethernet technológiát használnak, kézenfekvőnek tűnik az audió- és videófolyamok kezelése Ethernet formátumban. A jelenlegi fejlődési irányokat követve megállapítható, hogy a Gigabit Ethernet technológia nem csupán transzport megoldást kínál, hanem a jövő újgenerációs digitális videó hálózatának (Next-Generation Digital Video Network) alapja is lehet.

## 2. Motivációk

A hagyományos, szinkron audió- és videórendszerek valamint a csomagkapcsolt, aszinkron adathálózatok között definiált átjárás megvalósítása érdekében kidolgoztak egy nyílt szabványú fájlformátumot. Az MXF (Material eXchange Format) fájlformátum lehetővé teszi az audió- és videófolyamok szegmensenként való továbbítását az Ethernet hálózatokon. A folyamatokhoz adat és metaadat jellegű információ is társítható annak érdekében, hogy a fájl alapú együttműködés a különböző professzionális stúdiótechnikai alkalmazások között megvalósítható legyen.

A különböző fájlok továbbítása független azok tartalmától (például az alkalmazott videótömörítési eljárás-

tól), továbbá a nyílt felhasználásának köszönhetően nem szükséges gyártóspecifikus berendezések vagy egyéni transzport megoldások költséges telepítése sem [1].

A szabványosítást célzó törekvésekkel párhuzamosan a legtöbb gyártó egyedi termékskálát is fejlesztet. A piacvezető cégek legújabb sorozatú professzionális kamerái már Ethernet interfészekkel is rendelkeznek, valamint olyan digitális videórögzítők is piacra kerültek, amelyek támogatják a valós idejű felvételeknél akár ötször gyorsabb, nagyfelbontású videófolyamok transzportját Gigabit Ethernet interfészekon keresztül, vagy az MXF fájlformátum továbbítását 100BASE-T hálózati kapcsolatokon. A valós idejű videótranszport lehetőséget áttekintve láttuk, hogy a *Level 3 Connections* mérnökei sikeresen továbbítottak 30 illetve 50 Mb/s-os stúdió minőségű, tömörített, digitális, videó szegmenseket helyi hálózati infrastruktúrára [2].

A nagyvárosi (MAN) és gerinchálózatok (WAN) széleskörűen elterjedt hullámhosszosztásos WDM infrastruktúrára kialakított SDH transzport technológiát alkalmaznak. Ahogy a stúdiókban belül a videótranszport megoldások az Ethernet és Gigabit Ethernet technológia irányába tolódnak kézenfekvőnek tűnik, hogy a műsorszolgáltatók Ethernet Private Line vagy Ethernet Virtual Private Line [3] szolgáltatásokat igényeljenek a távoli helyszíneken lévő stúdiók és a kihelyezett közvetítőkocsik között.

A transzport szolgáltatóknak ezért olyan Ethernet konnektivitásokat kell biztosítaniuk a meglévő SDH/WDM gerinchálózatukon, amelyek megfelelnek az egyedi követelményeknek. Az újgenerációs SDH berendezések (NG-SDH) legújabb fejlesztései már részleges Ethernet funkciókkal rendelkeznek, az így kialakított – harmadik generációs SDH (TG-SDH) [5] – eszközökből felépített szolgáltatói hálózatok optimális infrastruktúrát jelenetnek a videótovábbítás számára.

### 3. Új kliensoldali követelmények

A műsorszolgáltatók előre egyeztetett SLA-nak (Service Level Agreement) megfelelő transzport szolgáltatásokat igényelnek az Ethernet keretezésű tömörített vagy tömörítetlen videófolyamjaik számára a szolgáltatói MAN vagy WAN hálózaton keresztül. Az audio- illetve videófolyamok továbbítása aszinkron, csomagkapcsolt hálózatokon szigorú csomagvesztési, késleltetési és késleltetés ingadozási követelményeket támaszt. A kliensek szolgáltatási követelményei:

- A megbízható videófolyam továbbítás.
- Garantált sávszélességű szolgáltatás.
- Az SLA-ban lefektetett minőség biztosítása.
- Az MXF fájlformátum továbbítása.
- Közel valós idejű (Just in Time) szolgáltatás.

Az új típusú televíziós alkalmazások (mint például a körzeti híradók, országos interaktív vetélkedők, regionálisan célzott reklámblokkok stb.) nagyobb rugalmasságot követelnek meg a hálózati kapcsolatok kialakításában. Főként kliensek által vezérelhető, Just in Time jellegű kapcsolat felépítés szükséges az egyes stúdiók között.

A különböző SLA-k definiálása lehetővé teszi a műsorszolgáltatók számára, hogy flexibilisen skálázhassák a transzport szolgáltatásokat az aktuális igényeiknek megfelelően. Példaként említhető, hogy élő közvetítések műsoranyagának továbbítása központi stúdióba, vagy közvetlen adótoronyba szigorúbb követelményeket támaszt, míg a tárolt műsoranyagok stúdiók közötti cseréjéhez elegendőek a kevésbé költséges, nem védett Ethernet kapcsolatok is.

### 4. Hálózati funkciók és architektúra

A transzport szolgáltatók szempontjából a tisztán kapcsolt Ethernet alapú szolgáltatói hálózatok jól ismert hiányosságokkal rendelkeznek a garantált sávszélesség, a QoS biztosítás valamint a gyors (<50 ms) védelmi/helyreállítási mechanizmusok alkalmazása terén. Másfelől azonban az Ethernet technológia alkalmazásának előnyös tulajdonságai (egyszerű telepítés és üzemeltetés, jó skálázhatóság és granularitás, VLAN alapú biztonság, alacsony költségek, stb.) jól kiaknázhatóak lennének a szolgáltatók számára. Ezért a tisztán kapcsolt Ethernet alapú megoldások hátrányainak kiküszöbölése mellett a technológia adta előnyök maximális kihasználása érdekében a hálózatüzemeltetők Ethernet funkciókkal ellátott újgenerációs SDH berendezéseket telepítenek a hálózataikba.

A szabványban rögzített GFP, VCAT és LCAS funkciók [6] megteremtik a műszaki alapot a kliensoldali követelményeknek eleget tevő transzport szolgáltatások kialakításához. Az NG-SDH alapú hálózati architektúra felett megvalósított Ethernet Private Li-

ne és Ethernet Virtual Private Line [3] transzport szolgáltatások a tradicionális bérelt vonali megoldásokkal egyenértékű szolgáltatási szintet garantálnak a felhasználók számára.

A klasszikus NG-SDH berendezések időrés alapú cross-connect (XC) funkcióval rendelkeznek, amely egy időrés számára lehetővé teszi, hogy az egyik fizikai interfészről a másikra kapcsolják. Ahogy az Ethernet szolgáltatások száma növekszik az SDH hálózatokban, egyre sürgetőbbé válik, hogy keret alapú cross-connect funkció is integráljanak az NG-SDH berendezésekbe [4]. A megoldás lényege, hogy minden Ethernet szolgáltatás egyedi VLAN címke alapján azonosítva van a hálózatban. A keret alapú cross-connect funkció a VLAN címkék információját használja fel a fizikai interfészek közötti továbbítás megvalósítása érdekében. Az integrált, VLAN címke alapú XC funkció alkalmazása a következő előnyökkel jár:

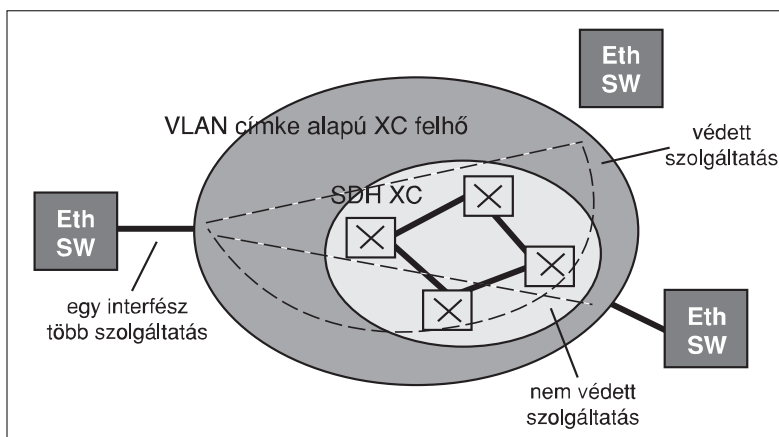
Több Ethernet szolgáltatás is megjelenhet egyetlen fizikai interfészen (VLAN címkével megkülönböztetve), így a kliensoldali Ethernet berendezések gazdaságosabb módon csatlakozhatnak a szolgáltatói oldalt képviselő, VLAN címke alapú XC felhőhöz (1. ábra).

Több Ethernet szolgáltatás oszthatja az SDH hálózat erőforrásait (sávszélességén), nevezetesen több pont-pont jellegű VLAN összeköttetés multiplexálható egyetlen virtuálisan összefűzött SDH konténerbe.

Statisztikus nyereség realizálható az SDH erőforrások felett kialakított virtuális Ethernet kapcsolatok között, azaz egyes Ethernet kapcsolatok maximális bitsebessége bizonyos időablakokban meghaladhatja az átlagot.

Mivel nem szükséges fizikai interfész alapján megkülönböztetni a szolgáltatói hálózat Ethernet hozzáférési pontjait, akár egy nagyságrenddel is csökkenthető a szükséges portok száma. A beruházási költségek csökkentése mellett az egyetlen interfészen reprezentálható több Ethernet szolgáltatás csökkenti az egységre jutó üzemeltetési költséget. A kliensek egyetlen interfészen keresztül több, különböző SLA-nak megfelelő transzport szolgáltatással is elérhetőek, így előfizetőnként nagyobb bevétel realizálható a hozzáférési infrastruktúra fejlesztése nélkül.

1. ábra Szolgáltatói hálózat architektúrája

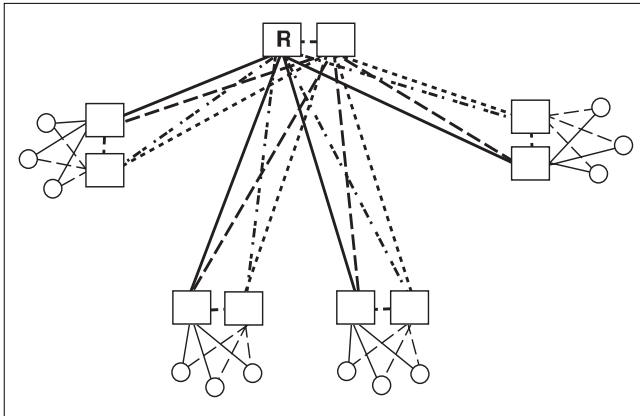


A szolgáltatók számos garantált sávszélességű, pont-pont VLAN kapcsolatot igyekeznek felépíteni az NG-SDH hálózatukon, így a megfelelő flexibilitás és Just in Time jellegű szolgáltatás érdekében a VLAN kapcsolatok automatikus menedzselése lényeges követelmény. A kapcsolatok felépítését és az Ethernet rétegben érvényes VLAN címkék adminisztrálását a GVRP (Generic VLAN Registration Protocol) támogatja. A protokoll automatizálja a hálózatban érvényes VLAN-ok dinamikus karbantartását, az Active Filtering Database információinak frissítésével és terjesztésével. A GVRP protokollt eredetileg nem a tisztán pont-pont VLAN kapcsolatok menedzselésére fejlesztették ki (a VLAN-ok valójában egy feszítőfa részfájára hasonlítanak), de megfelelő átalakításokkal [7] alkalmazható a szükséges funkciók ellátására.

### 5. Alkalmazási példa

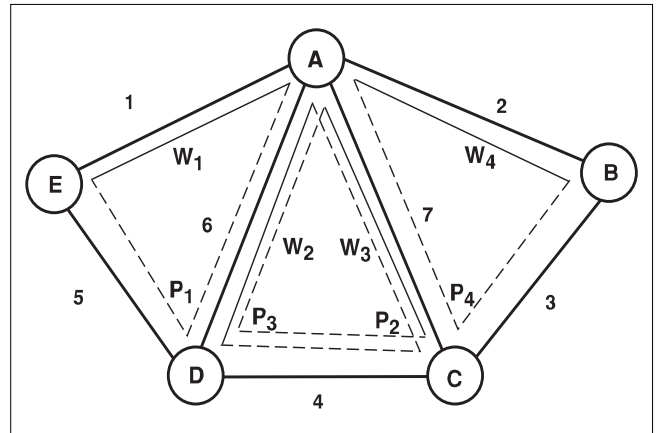
Az integrált VLAN alapú XC funkcióval ellátott NG-SDH hálózati architektúra előnyeinek illusztrálása érdekében egy hipotetikus magyarországi videótranszport hálózatot feltételeztem, öt nagyobb megyeszékhely között. A hálózati architektúra funkcionálisan egy Ethernet és egy SDH réteget tartalmaz.

Ethernet szinten a kisebb regionális stúdiók redundáns kapcsolatokkal csatlakoznak a megyeszékhelyeken elhelyezett, szintén duplikált Ethernet kapcsolókhoz. Az öt helyszín kettős csillag topológiájú logikai hálózattal van összekötve az Ethernet rétegbeli egyszerű hibák elleni védelem érdekében (2. ábra). A logikai hurkokat az STP algoritmus eliminálja.



2. ábra Ethernet logikai topológia

Az SDH transzport réteg fizikai topológiája él független elvezetéseket szolgáltató a redundáns Ethernet összeköttetések számára. Az Ethernet réteg STP algoritmus csak az adott rétegben bekövetkező hibák ellen véd. A fizikai link hibák ellen az SDH réteg 1+1-es útvédelme nyújt megfelelő biztonságot (3. ábra), mivel az STP algoritmus konvergencia ideje nem kielégítően gyors a kliensoldali követelmények teljesítéséhez. A fizikai linkhibákat az SDH védelem elrejtje az Ethernet réteg előtt.



3. ábra SDH fizikai topológia

A regionális stúdiókban a különböző videófolyamok továbbításához igényelt összeköttetések a GVRP protokoll által kiosztott különböző VLAN címkéket kapnak. A VLAN-ok ezután a megfelelő méretű virtuálisan összfeszített VC-4-es SDH konténerekbe kerülnek.

A szolgáltató által kínált különböző SLA-k alapján védett és nem védett, pont-pont VLAN összeköttetések igényelhetők a videófolyamok továbbításához. Az SDH réteg optimális erőforrás-kihasználtsága érdekében csak a védett VLAN igények számára célszerű gyors reagálású, 1+1-es védelmet nyújtani az SDH rétegben, ezért szükség van a hálózatban használt VLAN-ok azonosítására és csoportosítására. A VLAN alapú XC funkció hiányában a különböző VLAN-ok azonosítása csak az összetevő oldali interfészek alapján valósítható meg, míg az itt javasolt megoldásban a VLAN-ok egyedi címkéik alapján, közös interfészen is azonosíthatóak.

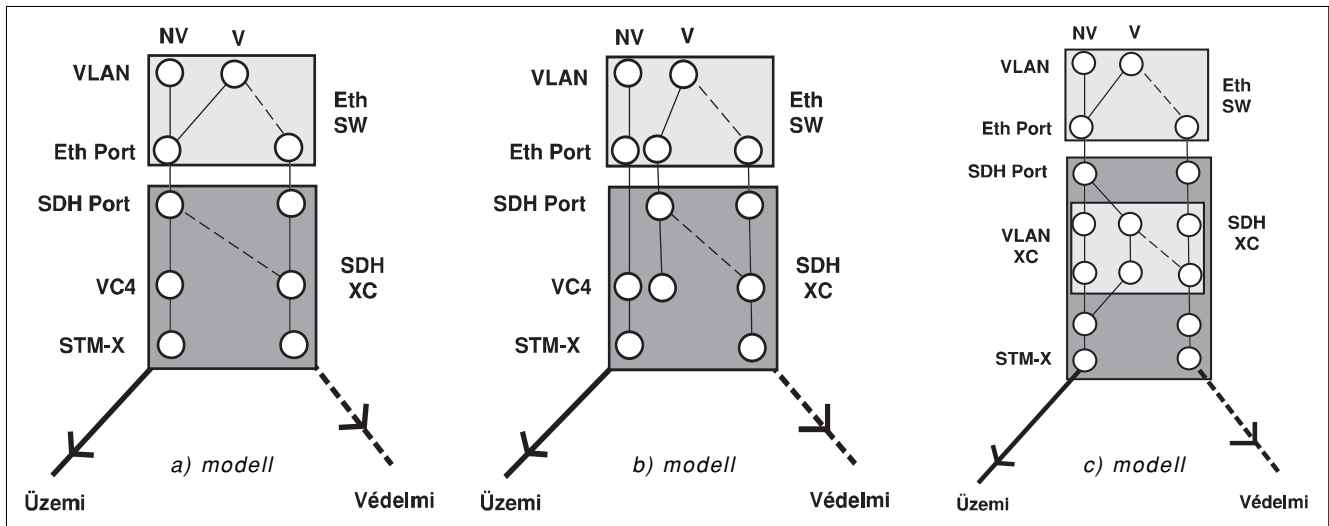
### 6. Hálózati és csomópont modellek

Az integrált VLAN alapú cross-connect funkció alkalmazásának vizsgálatához a megfelelő funkciókat azonosító, részletes csomóponti modellek szükségesek.

Az első csomóponti modell (a. modell) azt a referencia esetet mutatja (4. ábra), ahol minden pont-pont VLAN összeköttetés (védett és nem védett) egyaránt kap SDH védelmet. A VLAN-ok közös interfészen érkeznek az SDH berendezésbe (SDH XC), amely a VLAN alapú cross-connect funkció hiányában nem képes szelektív védelmet biztosítani külön a védett VLAN-ok számára.

Ha ennek ellenére csak a védett VLAN igények számára kívánunk SDH védelmet garantálni (b. modell), a VLAN-okat interfészek (portok) alapján kell csoportosítani az Ethernet kapcsolóban (Eth SW). A külön porton jelentkező VLAN-ok külön VC-4v konténerekbe tehetőek, így már biztosítható szelektív védelem az SDH rétegben. A kevesebb erőforrást igénylő szelektív SDH védelem ára tehát a több port a kliens oldali Ethernet kapcsoló és a szolgáltatói NG-SDH berendezés között.





4. ábra Ethernet – NG-SDH csomóponti modellek (a, b)

Amennyiben a védett (V) és a nem védett (NV) pont-pont VLAN igények sávszélessége (VLAN), az Ethernet interfészek mérete (GbE) és a virtuális SDH konténer összefűzés egysége (VC4) ismert, a következő formulákkal írható le a csomópontonként szükséges interfész szám (#port), valamint az SDH üzemi és védelmi kapacitás szükséglet (#Ü\_VC4, #V\_VC4).

a) modell: Nincs szelektív védelem

$$\#port_{1a} = \left\lceil \frac{\sum VLAN_P + \sum VLAN_{NP}}{GbE} \right\rceil + \left\lceil \frac{\sum VLAN_P}{GbE} \right\rceil$$

$$\#\ddot{U}_{VC4}_{1a} = \left\lceil \frac{\sum VLAN_P + \sum VLAN_{NP}}{VC4} \right\rceil$$

$$\#V_{VC4}_{1a} = \left\lceil \frac{\sum VLAN_P + \sum VLAN_{NP}}{VC4} \right\rceil$$

b) modell: Port alapú szelektív védelem

$$\#port_{1b} = 2 \times \left\lceil \frac{\sum VLAN_P}{GbE} \right\rceil + \left\lceil \frac{\sum VLAN_{NP}}{GbE} \right\rceil$$

$$\#\ddot{U}_{VC4}_{1b} = \left\lceil \frac{\sum VLAN_P}{VC4} \right\rceil + \left\lceil \frac{\sum VLAN_{NP}}{VC4} \right\rceil$$

$$\#V_{VC4}_{1b} = \left\lceil \frac{\sum VLAN_P}{VC4} \right\rceil$$

A második csomóponti modell (c. modell) (5. ábra) már tartalmaz az NG-SDH berendezésbe integrált VLAN címke alapú cross-connect funkciót (VLAN XC). A kliens oldali VLAN-ok az a) modellel megegyező módon, közös interfészen keresztül jutnak az SDH berendezésbe, ahol a VLAN címke alapú XC funkciónak köszönhetően az SDH réteg képes szeparálni a védett VLAN-okat a nem védettektől, így képes szelektív védelmet biztosítani csak a védett VLAN-ok részére is.

5. ábra

Integrált VLAN címke alapú, XC funkciót tartalmazó csomóponti modell (c)

Ez a megoldás kevesebb interfészt igényel, mint az a) modell, továbbá a statisztikus nyereségnek köszönhetően az b) modellnél is kevesebb az SDH erőforrás szükséglete.

Az alábbiakban következő formulák leírják a szükséges Ethernet-SDH interfészek számát (#port), és az SDH üzemi és védelmi kapacitás mennyiségét (#Ü\_VC4, #V\_VC4).

c) modell: VLAN címke alapú szelektív védelem

$$\#port_2 = \left\lceil \frac{\sum VLAN_P + \sum VLAN_{NP}}{GbE} \right\rceil + \left\lceil \frac{\sum VLAN_P}{GbE} \right\rceil$$

$$\#W_{VC4}_2 = \left\lceil \frac{\sum VLAN_P + \sum VLAN_{NP}}{VC4} \right\rceil$$

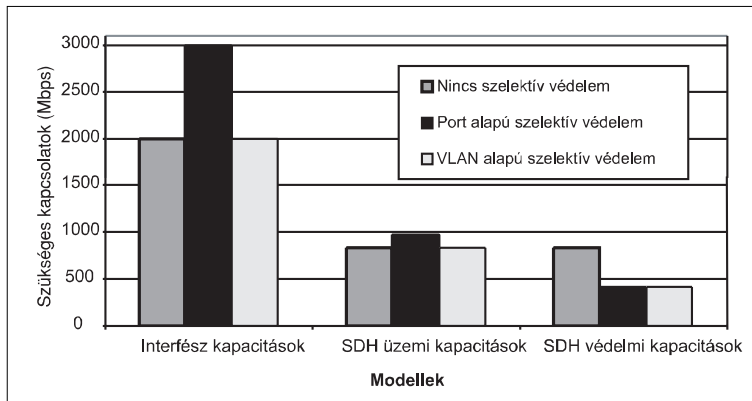
$$\#P_{VC4}_2 = \left\lceil \frac{\sum VLAN_P}{VC4} \right\rceil$$

## 7. Esettanulmány, eredmények

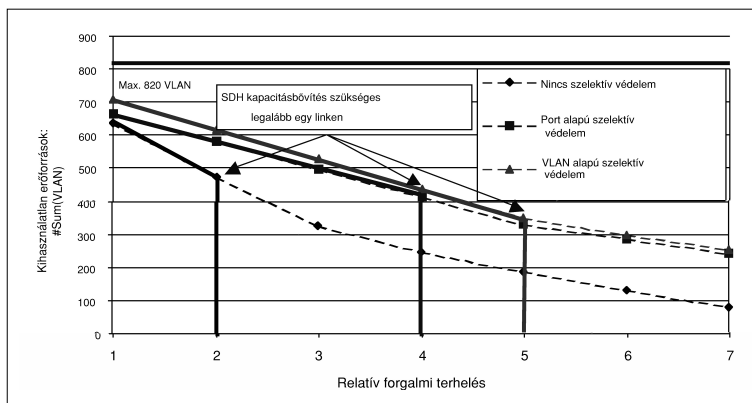
A bemutatott esettanulmány elemzéséhez egy feltételezett igénymátrixot használtam. A forgalmi mátrix a regionális stúdiók közötti pont-pont VLAN igények (különböző videófolyamok) számát tartalmazza.

Egy VLAN-ba kerülő stúdió minőségű, tömörítetlen videófolyam (IEC-601) sávszélessége 165 Mb/s. Az SDH virtuális konténer összefűzés alapegysége a VC-4-es konténer (139,264 Mb/s), mivel az alacsonyabb rendű (például VC-12-es) konténerekből maximálisan csak 64 darab fűzhető össze, amelyek együttes mérete így még kicsinek bizonyul [6].

Referencia esetben (a. modell) egy VLAN egy VC-4-2v összefűzött konténert igényel, a port alapú szelektív védelem megvalósítása érdekében (b. modell) viszont két különböző transzport szolgáltatást igénylő VLAN számára már egy VC-4-4v konténer lefoglalása



6. ábra Szükséges interfész és vonali kapacitások



7. ábra Kihasztnalatlan erőforrások száma a relatív terhelés függvényében

szükséges (2xVC-4-2v). A VLAN címke alapú XC funkcióval kiegészített architektúrában (c. modell) egy VLAN szintén egy VC-4-2v konténer igényel, de két különböző VLAN számára elegendő egy VC-4-3v virtuális konténer lefoglalása is, mivel a kapcsolatok VLAN címkejük alapján azonosíthatók.

Ezzel az egyszerű példával szemléltetve belátható, hogy a bemutatott részletes hálózati és csomóponti modellek alapján számtalan, egyszerű és komplex esettanulmány analizálható. A 6. ábra azt mutatja, hogy a VLAN címke alapú cross-connection funkció implementálásával kevesebb SDH védelmi kapacitás és kevesebb fizikai Ethernet-SDH interfész segítségével valósíthatók meg a kliensoldali követelményeket optimálisan kielégítő transzport szolgáltatások.

A modellek lehetővé teszik összetettebb hálózati szintű elemzések elvégzését is. Hálózati szinten a szükséges erőforrások összessége helyett a transzport szolgáltatók számára lényegesebb kérdés a hálózatban található kihasztnalatlan erőforrások mennyisége, illetve a szükséges kapacitásbővítések várható időpontja. A bemutatott hálózatmodellre alapozottan a következő diagram (7. ábra) adott linkkapacitások mellett a kihasztnalatlan erőforrások számát, illetve a legalább egy linken szükséges kapacitásbővítés időpontját szemlélteti a relatív forgalmi terhelés függvényében.

Ahogy látható, a port alapú szelektív védelem (b. modell) lényegesen kitolja a kapacitásbővítés várható időpontját a teljes védelmet kínáló megoldáshoz ké-

pest (a. modell), ahogy az várható is volt. A diagramról azonban az is leolvasható, hogy a VLAN címke alapú XC funkciót tartalmazó architektúra alkalmazása (c. modell) további erőforrás nyereséget realizál (még jobban kitolja a kapacitásbővítés időpontját), mindemellett lényegesen kevesebb interfészt igényel, mint a port alapú szelektív védelem megvalósítása.

A referencia modellel (a. modell) összehasonlítva a javasolt hálózati architektúra (c. modell) azonos hozzáférési hálózat felett (azonos interfész szám mellett) jobb SDH erőforrás kihasználást, jobb hálózati teljesítőképességet és a port alapú szelektív védelmet megvalósító modellnél (b. modell) alacsonyabb összköltséget garantál.

## 8. Összegzés

A modern, digitális stúdiótechnika által motivált, Ethernet alapú videótranszport szolgáltatások optimális kielégítése érdekében az újgenerációs SDH transzport hálózatokat célszerű kiegészíteni integrált Ethernet funkciókkal. A bemutatott csomóponti modellek alapján az integrált VLAN címke alapú cross-connect funkció előnyei egyszerűen analizálhatók a transzport hálózat erőforrás szükséglete, kihasználtsága, port költsége és összköltsége szempontjából.

## Irodalom

- [1] EBU Technical Review – B. Devlin: MXF – the Material eXchange Format, 2002.
- [2] Sony Press Release – Sony and Level 3 transfer broadcast video segments across Ethernet network directly from tape playback, 2003. <http://news.sel.sony.com/pressrelease/4035>
- [3] Appian Communications – Carrier-class Ethernet: A service definition, 2001. <http://www.appiancom.com/solutions.htm>
- [4] Marconi – G. W. Rees: Physical integration of SDH switching and Ethernet switching – Analyzing the opportunities and constraints, 2002.
- [5] Heavy Reading – The future of SONET/SDH, Vol. 1, No. 6, Nov. 14, 2003.
- [6] ITU-T G.7041/Y.1303 GFP, ITU-T G.707 VCAT és ITU-T G.7042/Y.1305 LCAS
- [7] F. V. Quickenborne, F. De Greve, P. V. Heuven, F. De Turck, B. Vermeulen, S. V. den Berghe, I. Moerman, P. Demeester: Tunnel set-up mechanisms in Ethernet networks for fast moving users, 2004.

# Szerzők, jogok, díjak, kérdőjelek...

## Beszélgetés Bendzsel Miklóssal, a Magyar Szabadalmi Hivatal elnökével

NAGY BEATRIX HAVASKA

*nbh@mailbox.hu*

**Az utóbbi hetekben számos véleményt olvashattunk a sajtóban a szoftverekkel kapcsolatos jogdíjak és szabadalmi díjak problémáiról. Igyekeztünk megismerni egy elismert szakértő véleményét. Ennek eredménye olvasható az alábbi interjúban.**

*Az internetre felkerült anyagoknál hogyan lehet megállapítani, hogy kinek jár és kinek nem jár szerzői jog? Vagy az internetre is automatikusan vonatkoznak a szerzői jogi szabályok? Mi számít illegális letöltésnek az internet esetében?*

Minden, ami végbemegy az interneten, az az e világon található számítógépek között történik, az e világon található kommunikációs csatornákon megy keresztül, az e világban lévő művek és egyéb tartalmak felhasználásával, az e világban élő emberek között. Semmi sem indokolja tehát, hogy ami az internet világában történik (pl. szerződéskötés, kereskedelem, adatvédelem, stb.), arra ne vonatkozzanak a jog előírásai, tehát a szerzői jog szabályrendszere is.

Az internetes tartalomszolgáltatás a szerzői jogi terminológiában nyilvánossághoz való közvetítésnek minősül. Az egyedi lehívásra való hozzáférhetővé tétel folyamatában erre akkor kerül sor, amikor a művek, előadások és hangfelvételek digitális formában eljutnak a közönség azon tagjaihoz, akik hozzákapcsolódtak a szolgáltatáshoz. A már nyilvánosságra hozott irodalmi és zeneművek, előadások, valamint képző- ipar- és fotóművészeti alkotások nyilvánosság számára történő hozzáférhetővé tétele közös jogkezelői jogosítással valósulhat meg. (Kivéve abban az esetben, a szerző vagy más jogosult külön nyilatkozattal „kivonja” a műveit, teljesítményeit a közös jogkezelés alól.) Ez azt jelenti, hogy e szellemi teljesítmények internetes felhasználására az Artisjus-al (az irodalmi és zenei szerzők képviselőjében), az Előadóművészi Jogvédő Irodával (az előadóművészek képviselőjében) és a Hungart-al (a képzőművészek és fotóművészek képviselőjében) kell szerződni. Hangfelvételek és filmalkotások on-line felhasználása nem tartozik a közös jogkezelés körébe, így értelemszerűen szükség van a hangfelvétel kiadók és a filmelőállítók egyedi engedélyére. Az adatbázisok online jogosítására is az érintett jogosultól (az esetek többségében az előállítótól) kell egyenként engedélyt kérni.

Akkor beszélhetünk tehát illegális letöltésről, ha az említett felhasználási engedélyek hiányoznak. Ezek meglétéért elsősorban a kérdéses tartalom (például

egy honlap) előállítója, másodsorban a tartalomszolgáltató felel. A köztes szolgáltatók felelőssége többnyire korlátozott. A nem jogosított tartalmak fogyasztóinak, letöltőinek a felelőssége is megállapítható, hiszen engedély nélkül végeznek szerzői jogi szempontból releváns cselekményeket pl. többszörözést.

*Ön szerint Bill Gates munkái során végzett-e olyan mértékű szellemi tevékenységet, munkát, hogy a világ összes géphasználójától jogdíjat kér? És ha ezek a programok két-három év múltán elévülnek, jogos-e, hogy utána is fizetnek jogdíjat?*

Úgy gondolom, hogy nem a Magyar Szabadalmi Hivatal elnökének tiszte megítélni Bill Gates, illetve a nevével fémjelzett szoftvergyártó cég szellemi investícióját, kreativitását és az ebből származó bevétel (jogdíj) mértékének jogosságát. Azt azonban tudni kell, hogy a Microsoft nem egy emberből áll. Egyfelől e szoftvergyártó óriás, programozók, kutatók százait, ha nem ezreket foglalkoztatja. Hihetetlen összegeket fordít fejlesztésekre, amely befektetésnek – különösen, ha az tömegmérétekből forgalmazott termékben ölt testet – valahol meg kell térülniük. A befektetett és hasznosítható produktumokban materializálódó tudást a civilizált világban minden országában védi a szellemi tulajdon jogi intézményrendszere.

Másfelől Bill Gates nem egyszemélyes tulajdonosa a Microsoftnak. E világcégnek számtalan tulajdonosa van (leginkább kisorosztályosok), akiknek az érdekeit is figyelembe kell venni a cég menedzsereinek. Ez szintén piaci szemlélet: egy cég vezetése nem hagyhatja figyelmen kívül tulajdonosainak legfontosabb érdekét, jelesül azt, hogy a lehető legkisebb ráfordítással a lehető legnagyobb profitot, nyereséget érje el.

A két-három éves elévülésre vonatkozó kérdés nem teljesen világos. Ha a kérdést úgy tesszük fel, hogy indokolt-e jogdíjat szedni azután is, hogy a szoftver már elavult, a válaszom a következő:

Egy szoftver nyilvánvalóan csak addig piacképes, amíg igény van rá. Ha egy szoftverfejlesztő új változattal vagy újabb operációs rendszerrel jelenik meg, akkor az erre átálló fogyasztók (vagy maguk a fejlesztő cégek) többnyire kivonják a régi alkalmazásaikat. Amennyiben a felhasználási szerződés alapján hosszabb, az adott szoftver elavulási idején túlmenő időszakra állapít meg jogdíjfizetési kötelezettséget, akkor az a kötelelem polgári jogi vagy versenyjogi oldalról támadható meg.

*Mi a véleménye az operációs rendszerek használati díjáról?*

Ha a kérdés a szoftverek szabadalmaztathatósága körül kibontakozott vitával és kormányzati állásponttal van kapcsolatban, akkor a válaszom a következő.

A számítógéppel megvalósított találmányok szabadalmazhatóságáról szóló EK irányelvjavaslat az Európai Szabadalmi Egyezmény alapján kialakult szabadalmazási gyakorlatot tükrözi. Célja, hogy a szabadalmazásnak az amerikai mintára történő további liberalizálását megakadályozza, továbbá, hogy ezáltal az Európai Bíróság értelmezési hatáskörét erre a területre is kiterjessze, így befolyásolva az Európai Szabadalmi Hivatal tevékenységét. Érdemes összefoglalni azokat az érveket, amelyek indokolják, hogy Magyarország továbbra is támogassa az irányelvjavaslat tárgyában az EU Versenyképességi Tanácsának 2004. május 18-i ülésén elfogadott politikai megállapodást:

Az irányelvjavaslat az Európai Szabadalmi Egyezmény (ESZE) alapján kialakult status quo-t tükrözi, nem teszi lehetővé olyasféle számítógépi találmányok szabadalmazását, amelyekre eddig az Európai Szabadalmi Hivatal (ESZH) nem adott szabadalmat. Ez volt az Európai Bizottság szándékolt célja is az irányelvjavaslattal, és ennek a célnak a megvalósítását támogatták a tagállamok a Versenyképességi Tanácsban.

Az irányelv hiányában az ESZH könnyebben követhetné – szabadalomengedélyezési gyakorlatának további liberalizálásával – az Amerikai Egyesült Államokban kialakult joggyakorlatot, ami éppen ellentétes volna az irányelvjavaslat ellenzőinek céljaival.

Az irányelvjavaslat főbb gazdasági céljai között szerepel a jogbiztonság erősítése és ezzel a gazdasági döntésekhez szükséges kiszámíthatóság megteremtése. Ugyancsak fontos szempont az egyes tagállamok joggyakorlatában jelentkező különbségek megszüntetésével az egységes belső piac zavartalan működésének a biztosítása. Hangsúlyos cél az európai informatikai ipar versenyképességének erősítése a világpiacra, főként amerikai és japán versenytársaikhoz képest. Végül, de nem utolsó sorban – éppen a Magyar Szabadalmi Hivatal szakpolitikai célkitűzéseivel összhangban – az irányelvjavaslat ösztönözni fogja az innovációt és az alkotó tevékenységet e fontos csúcstechnológiai területen. A tudástőke szerepének felértékelődésével párhuzamosan, Magyarország alapvető érdeke, hogy ösztönözze a kutatási-fejlesztési, illetve az olyan csúcstechnológiai területekre eső befektetéseket, mint az informatika, illetve a „szoftveripar”. Az irányelvjavaslat ellenzői többnyire olyan tanulmányokra hivatkoznak e körben, amelyek a javaslat előterjesztését megelőzően születtek, illetve nem az európai helyzetre, hanem az USA-ban kialakult viszonyokra vonatkoznak. Még e tanulmányok is csak a szoftver önmagában való szabadalmazhatóságát kritizálják, amit az irányelvjavaslat viszont kizárna.

Ha pedig a „vékonyabb pénztárcájú” szoftverfejlesztők mint kisvállalkozások felől közelítjük meg kérdést, akik kétségkívül kis mértékben veszik igénybe a

szabadalmi rendszert (aminek jórészt a kellő ismeretek és a pénzügyi források hiánya lehet az oka), akkor korántsem az a megoldás, hogy a hozzáférés lehetőségét (tehát bizonyos technikai területhez tartozó találmányok szabadalmazhatóságát) szüntetjük meg, hanem az, hogy javítjuk e vállalkozások esélyeit a szabadalmi rendszer előnyeinek hatékony kihasználására.

Szeretném eloszlatni azt a félreértést, hogy akár a szabad szoftver, akár a nyílt forráskódú szoftver ingyenes szoftvert jelentene. Valójában olyan üzleti modellről van szó, amely a szerzői jogra épül és olyan szerzői jogi felhasználási engedélyeket alkalmaz, amelyekkel a résztvevők lemondanak egymás között szerzői jogaik gyakorlásáról a szoftver fejlesztése, használata és terjesztése kapcsán. Ebben az esetben a szerzői felhasználási engedély az ingyenes, viszont kizárna bármiféle jog- és kellékszavatosságot a szoftverrel kapcsolatban, és ugyancsak kizárják a fejlesztő felelősségét a szoftver hibái miatt bekövetkezett károkért.

*Az új szabályok szerint, ha két embernél több hallgat egy előadást, zeneszámot azután már jogdíjat kellene fizetniük. Szélsőségesen megfogalmazva a kérdést, ha egy anya a gyermekeinek este elénekel egy dalt, azután már jár-e a szerzői jogi jogdíj? A törvény megfogalmazása nem pontatlan és szélsőséges egy kissé?*

1999 óta változatlan a szerzői jogi törvénynek azon rendelkezése, miszerint a szerző kizárólagos joga, hogy művét nyilvánosan előadja, és hogy másnak erre engedélyt adjon. Az előadás nyilvánossága esetén az a döntő kritérium, hogy a nézők, hallgatók köre meghaladja-e családiasság körét. Amennyiben meghaladja, úgy a mű előadása szerzői engedélyéhez kötött, és díjazásra tarthat igényt, ellenkező esetben, mint az Ön által felhozott szép példában a gyermekeinek éneklő édesanya, a szabad felhasználások körébe esik. A válasz egyben rámutat arra is, hogy a több évtizedes bírói gyakorlaton és nemzetközi megállapodáson nyugvó törvény egzakttól mentes megoldásokat tartalmaz.

*A jogszabály szerint a kazetta, CD ára, amit a boltban megveszek – az üres is – már tartalmazza a művész és a jogvédő iroda részét. Mégis általánosan elterjedt, hogy ez a gyakorlatban nem így működik. Sőt, többször is megfizettetik velem a díjakat. Hogy is működik ez a gyakorlatban? És hogy működik mindez a megvásárolt koncert-, előadás-, és színházjegyek esetében?*

Az üres hordozók fogyasztói, végfelhasználói árába csakugyan be van építve a jogdíj, ami feljogosítja a kazetta, CD tulajdonosát arra, hogy erre – kizárólag magáncélra, tehát mindenfajta jövedelemszerzési lehetőséget kizárva – hangfelvételt vagy filmalkotást másoljon. Nem árt tudni, hogy mentességet élveznek ez alól az exportált termékek, valamint az üzletszerűen hang- és képhordozó másolását végző vállalkozásoknak történő értékesítés. Néhány éve kis hologramos

matrica igazolja, hogy a kazetta, írható CD után a forgalmazó megfizette a jogdíjat. Érdemes megjegyezni, hogy a 2001/29/EK irányelv (az ún. INFOSOC-irányelv) előírása szerint a magáncélú másolás díjának megállapításakor figyelembe kell venni a hatásos műszaki intézkedések alkalmazását. Ebből az következik, hogy a díjnak tükröznie kell azt is, hogy alkalmaznak-e hatásos műszaki intézkedéseket a másolás megakadályozására, korlátok között vagy ellenőrzés alatt tartására. Konkrétabban: a díj összegének a megállapításánál figyelembe kell venni azt, hogy az érintett üres kép- és hanghordozókat milyen arányban lehet magánmásolásra felhasználni ingyenesen, illetve díjfizetés ellenében. Ez a magyarázata annak, hogy a szerzői jogi törvény 2004. május 1-jétől hatályos módosítása változtatott a díjfelosztási arányokon: 45% illeti meg a zeneszerzőket és az írókat (korábban 50%), 30% az előadóművészeket (korábban is ennyi volt) és 25% a hangfelvétel-előállítókat (korábban 20%). A törvényalkotó tehát éppen hogy visszaigazolta a gyakorlat által felvetett igényeket.

Az üres hordozók esetében tehát nem beszélhetünk többszörös jogdíjfizetésről. Nincs erről szó más zenefelhasználások esetében sem, hiszen minden egyes felhasználás – jellemzően közvetlen (pl. koncert), vagy közvetett (pl. éttermi háttérzene) bevételszerzés célját szolgáló nyilvános zeneszolgáltatás – után külön-külön jogdíjat kell fizetni. A diszkó rendezvények után fizetendő nyilvános előadási jogdíj megfizetése (az Artisjus-nak) nem váltja ki a lemezbemutató műsorszolgáltatás céljából készült CD-R többszörözés után fizetendő jogdíjat (a MAHASZ-nak). Koncertek, egyéb előadások esetében a létesítmény alapterülete vagy befogadóképessége alapján határozzák meg a jogdíj mértékét, amit a fenntartó, szervező minden bizonnyal érvényesít a belépőjegy árában. Ezeket a tarifákat maguk a közös jogkezelő szervezetek állapítják meg, korántsem minden kontroll nélkül. A jogdíjközleményeket évről évre a nemzeti kulturális örökség minisztere hagyja jóvá. Nem árt tudni, hogy a kulturális tárcának 2004-től a Magyar Szabadalmi Hivattaltól is véleményt kell kérnie a közös jogkezelő szervezetek díjszabásairól. A törvény emellett véleményezési jogot biztosított az informatikai és hírközlési miniszternek is az ún. lehívásra való hozzáférhetővé tételre, vagyis az internetes műtovábbításra irányadó díjszabási feltételek tekintetében.

*Van-e arra lehetőség, hogy a szerzői jogdíj helyett valami olyan díjat vagy költséget vezetnének be, ami ellenőrizhetőbb és hatékonyabban működik, mint a jelenlegi rendszer? Meg lehet-e akadályozni, hogy az internetre felkerüljenek zeneszámok, előadások?*

A szerzők, alkotók méltányos díjazására való igény egyidős a modern szerzői jogok történetével. Pillanatnyilag nincs olyan lehetőség, amely a jelenlegi díjazási rendszert ellenőrizhetőbbre és hatékonyabbra váltaná fel. Egyes teoretikusok ugyan felvetették egyfajta globális átalány bevezetésének lehetőségét, de ez jelen-

leg ellentétes a vonatkozó nemzetközi egyezményekkel. Másfelől nem is lenne méltányos, ha a szerzők nem rendelkezhetnének műveik felhasználásának hasznából. Kérdésének második felére is röviden lehet válaszolni; a mai korszerű hangrögzítési technológiák eredményeként bárki, bármilyen zeneszámot, előadást feltölthet az internetre, amely napjainkra elképzelhetően szűles kulturális kínálatot eredményezett.

*Lehet-e egy olyan egzakt módszert találni, ahol a mű értéke és az annak hasznosításából eredő haszon összeegyeztethető-e a szerzői joggal?*

Ilyen „egzakt módszerről” nincsen tudomásom, de utalnék a digitális közös jogkezelés, vagy a lap olvasói előtt ismertebb rövidítéssel nevezve, a DRM fontosságára, amely az európai egységes belső piacon jelentős eszközzé válhat az új digitális szolgáltatások vonatkozásában. A DRM rendszereket a jogok tisztázására, a fizetés biztosítására, a jogkövető magatartás kijelölésére és a jogérvényesítésre alkalmazhatják a jogosultak. A kérdéshez szorosan kapcsolódik, és ezért néhány gondolatban említést tennék arról is, hogy a szerzői jogi alapú ágazatok gazdasági jelentőségének számszerűsítésére irányuló munkálatok nemzetközi szinten a 20. század második felében indultak meg. Az ENSZ szakosított szervezete, a Szellemi Tulajdon Világszervezete (közismert rövidítése: WIPO) ösztönzi a szerzői jog gazdasági jelentőségének vizsgálatát. Egy 2003-ban megjelent módszertani munkában összegzett tapasztalatok szerint elsősorban a fejlődő országokban és az átmeneti gazdaságokban jelenthet problémát a szerzői jogi alapú iparágak gazdasági hozzájárulásának a dokumentálása. A WIPO Lettország mellett hazánkat választotta ki és kérte fel a közép-kelet-európai térségben úttörő jelentőségűnek számító felmérésben való részvételre, amelynek koordinációs munkáit a Magyar Szabadalmi Hivatal látja el. 2003 októberében – a WIPO módszertani útmutató alapján – az Európai Bizottság készítettett egy hasonló felmérést az EU akkori tizenöt országára vonatkozóan. Jól mutatja a szerzői jogi alapuló ágazatok gazdasági jelentőségét, hogy 2002-ben az Unió tizenöt országában együttesen ezek az iparágak adták a GDP 5,3 százalékát!

*Bármilyen termék vásárlása szabad elhatározás. Van-e arra lehetőség, hogy ezeket a vásárlásokat úgy, mint az Internet megjelenése előtt ellenőrizhetővé tegyék?*

Alapvetően az interneten történő vásárlások, az elektronikus kereskedelmi kapcsolatok bármely típusánál, legyen szó akár a gazdálkodó szervezetek egymás közötti vagy a gazdálkodó szervezetek és a fogyasztók között létrejött ügyletekről, a szabályozásnak a forgalom biztonságát kell garantálnia, amelyre a fogyasztóvédelmi rendelkezések az irányadók. Szerzői jogi szempontból a beszélgetésünk kezdetén már szóba került internetszolgáltatók felelősségkorlátozása és az elektronikus törvényben szabályozásra került értesítési-eltávolítási eljárás jöhet számításba.

# Pályázatok

Az Ipar Műszaki Fejlesztéséért Alapítvány pályázata

## „AZ ÉV KIEMELKEDŐ FIATAL MŰSZAKI ALKOTÓJA”

### díjra

Azoknak a felsőfokú végzettségű fiatal tehetséges műszaki alkotóknak, illetve alkotócsoportoknak\* az elismerése, akik valamely jelentős eredményükkel kivívták a szakma vagy közvetlen környezetük elismerését, és akiket példaképpül lehet állítani a magyar műszaki társadalom elé. 2004-ben a 2001-2004. között elért műszaki-tudományos eredményekkel lehet pályázni\*\*.

A pályázaton a 2004-ben legfeljebb 35. életévüket betöltő, magyar állampolgárságú műszaki alkotók vehetnek részt.

#### A pályázat témája és díjazása:

Szakterület: az ipar egésze (kivéve az építészetet és az élelmiszeripart). A pályázat témájának gyakorlati megvalósításáról már be lehessen számolni. A pályázat tartalmazzon összehasonlítást a piaci versenytársak hasonló termékeivel, fejlesztési eredményeivel.

A pályázatot neves szakemberekből álló zsűri bírálja el. Öt pályázatot díjazunk:

I. díj	900 eFt
II. díj	700 eFt
III. díj	500 eFt
IV. díj	300 eFt
V. díj	200 eFt

További különdíjakat adnak ismert hazai nagyvállalatok is. A fenti bruttó összegekből adóelőleg kerül levonásra, mert a díjak személyi jövedelemadó-köteles jövedelemnek minősülnek.

A díjat nem nyert, de színvonalas pályamunkák készítői dicsérő oklevelet kapnak. A pályamunkákat és a pályázókat az Internet honlapunkon, valamint színvonalas évkönyvben mutatjuk be, amelyet szakmai fórumokon terjesztünk.

#### Beadási határidő: 2004. december 15.

A díjak illetve dicsérő oklevelek átadására és a pályamunkák visszaadására ünnepélyes keretek között, a média nyilvánossága előtt, 2005. tavaszán kerül sor.

A pályázatok a következő címre küldhetők:

Ipar Műszaki Fejlesztéséért Alapítvány  
1063 Budapest Munkácsy Mihály u. 16.  
1387 Budapest Pf. 17.

Telefon: 312-2213, Fax: 332-0787

Internet: <http://www.imfa.hu>

A korábbi pályázati eredmények a fenti honlapon találhatóak.

\* Alkotócsoport csak akkor vehet részt a pályázaton, ha a csoport minden tagja megfelel a támogatási feltételeknek.

\*\* A pályázaton legkorábban a diplomázást követő évben lehet részt venni.

A Huszty Dénes Alapítvány Kuratóriuma az alapítvány alapító okiratában foglaltak szerint

## PÁLYÁZATI FELHÍVÁST tesz közzé

A 2001. novemberében bejegyzett Huszty Dénes közhasznú alapítvány célja, hogy az akusztika, vagy az elektroakusztika területén tevékenykedő fiatal szakemberek, felsőfokú tanulmányukat éppen befejező vagy már végzett fiatalok, illetve a pályázat beadásakor 35. életévüket még be nem töltött fiatal akusztikusok olyan kiemelkedő eredményeit jutalmazza, amelyek hozzájárulnak az akusztika egyetemes fejlődéséhez. Az Alapítványnak további célja hogy emléket állítson Huszty Dénes munkásságának, aki az 1950-1979 közötti időszak kiemelkedő akusztikai szaktekintélye volt.

A Huszty Dénes emlékdíj 2005-ben emléklapokból és összesen 300.000 Ft. jutalomból áll. A díjazottak dönthetnek arról, hogy a jutalmat pénz, vagy a hatályos törvényeknek megfelelő természetben megszerzett bevétel formájában (konferencia részvételi díj kifizetése, tudományos közlemény megjelentetésének díja stb.) kívánják majd igénybe venni.

A Kuratórium a 2005. évben két díjat tervez kiadni, melyekre pályaművel lehet pályázni. Pályázhatnak végzett, elsősorban mérnökök, fizikusok saját önálló munkájuk összefoglaló dolgozatával, szakirányú lapban megjelent cikkeikkel, vagy új dolgozattal. A pályázónak lehet más szakirányú diplomája is, de tevékenységét az akusztika területén kell, hogy kifejtse.

#### A 2005. évi pályázat kiemelt témaköre:

Informatikai módszerek alkalmazása akusztikai problémák megoldására.

#### A pályázatok beérkezésének határideje: 2004. december 1.

A pályázatok beadhatók személyesen az Alapítvány székhelyén: a Hírközlési és Informatikai Tudományos Egyesület irodájában (1055 Budapest, Kossuth tér 6-8.), vagy postai úton. A postán feladott küldemények esetén a feladás legkésőbbi időpontja 2004. november 30. lehet.

A pályázatokat a Kuratórium által felkért bizottság értékeli és javaslatot tesz a díjazásra a Kuratóriumnak.

A Huszty Dénes Emlékdíj ünnepélyes kiosztására a 2005. január 15. és február 15. közötti időszakban, az Alapítók képviselőinek jelenlétében kerül sor.

A jelen pályázati felhívás a Hírközlési és Informatikai Tudományos Egyesület valamint az Optikai, Akusztikai és Film- és Színháztechnikai Tudományos Egyesület tájékoztató Hírlevelében és a Zajinfóban valamint az Egyesületek honlapjain jelenik meg.

A Kuratórium nevében:

Dr. Illényi András, a Kuratórium Elnöke

# „Csökkenő piacon is sikeresnek lenni”

## Interjú Hetényi Péterrel, a Siemens Rt. Felügyelő Bizottságának elnökhelyettesével

NAGY BEATRIX HAVASKA

*nbh@mailbox.hu*

*Már a Sicontact-időkben is sikerre vitte a Siemenst. Abban az időszakban mivel tudott sikeres lenni, mi tette vonzóvá a Siemens termékeket?*

Köszönöm a bizalmát, de nem én vittem sikerre a Siemenst; a Sicontactnál nem én vezettem a céget, és ott már akkor is csapatmunka volt, ami ma is jellemző. 1974-ben alakult meg a Sicontact, de 1968-tól már volt kisebb Siemens-jelenlét, egy képviseleti iroda működött az Intercooperation Rt.-nél.

A siker titka az volt, hogy szolgáltatást nyújtottunk a magyarországi vevőknek. Ezt akkor még kevés cég kínálta Magyarországon, szinte mindegyiknél csak közvetítő ügynöki kereskedelem volt. Erről nagyon hamar áttértünk az ügyfélszolgálatra és a szervizre, így természetesen sokkal szorosabb kapcsolat alakult ki a felhasználókkal. A számítástechnikában és részben a híradástechnikában ezzel váltunk piacvezetővé, – megelőzve az IBM-t is. Már azon munkálkodtunk, hogy a képviseleti irodából igazi szolgáltató vállalat váljon. Ez 1974-ben vált lehetővé, amikor nyugati cégek részvételével egyes vállalatok alapítását engedélyezték. Ez egy pénzügyminiszeri rendelet volt, törvényt erre nem mertek hozni. Mi voltunk az első cég, aki ezt a lehetőséget kihasználta, és megalapítottuk az első magyarországi és kelet-európai joint venture-t, a Sicontactot. A Sicontactnál már nagybetűvel volt felírva a „szolgáltatás”. Továbbfejlesztettük a szervizt, és megkezdtük a szoftverfejlesztést.

Az állami külkereskedelmi monopólium fokozatosan kezdett feloldódni. Ezt kihasználva két forgalomirányító számítógépet hoztunk be. Ezzel olyan szolgáltatásokat indíthattunk el, amilyenekre addig nem volt lehetőség. Politikánk mindig az volt, hogy az adott körülmények között a kereskedelmi és üzleti kapcsolati lehetőségekből a legtöbbet kell kihozni. Az akkori elvárás az volt, hogy minél több nyugati know-how és tudás jöjjön be Magyarországra. A nyugati cégek pedig azt várták, hogy minél jobban megnyíljon a magyarországi piac. Ezt a kettőt kellett összeegyeztetni, mert abban a deviza-hiányos, zárt kereskedelmi rendszerben természetes hazai célkitűzés volt, hogy deviza nélkül lehessen hozzájutni a technológiákhoz, árukhoz; míg a külföldi cégek arra törekedtek, hogy megnyíljon a piac. Így alakultak ki pl. a barter üzletek, a licencvásárlás, amit áruval fizettünk meg.

*A híres rendszervásztó tender során neves cégeket győztek le. Többeknek volt már jelentős gyártó*



*háttere. A siker érdekében a Siemens partnerként megnyerte az egyik legjobban felkészült híradástechnikai gyárat, a Telefongyárat. Ez mennyiben volt személyes érdeme, mennyire a kapcsolatok hatása, vagy valóban kiemelkedően jobb volt a Siemens?*

A rendszerváltással tulajdonképpen párhuzamosan indult a rendszervásztó tender. Ez azért volt lehetséges, mert a kereskedelmi liberalizáció megelőzte a politikai rendszerváltást, bár akkor már liberálisabb volt a politika is. Felismerték, hogy a magyarországi távközlési szint tarthatatlan. Ugyanis a rendszerváltás előtt a telefonsűrűség tekintetében Európában csak Albánia volt mögöttünk. Ha jól emlékszem, 100 főre nálunk 8 főállomás jutott, míg Albániában 6. Németországban ebben az időben 60 fölött, Svédországban 70-80 körül volt ez a szám. Nyilvánvalóvá vált, hogy ez a helyzet így nem tartható, mert az egész nemzetgazdaságot negatívan befolyásolja, ráadásul akkor a világban már elkezdődött az információs forradalom is. Ennek kapcsán megszületett az a kormányzati döntés, hogy egy rendszervásztó tendert kell kiírni a telefonhálózat radikális modernizációjára, elsősorban kapcsolástechnikára. Kellott egy döntés, mely szerint két központrendszert választanak úgy, hogy azokat Magyarországon kell gyártani. Akkor ez volt a trend az egész világban – ahol ilyen rendszervásztás volt, ott elvárták, hogy a központokat gyártsák az országban.

Sikerünk titka az volt, hogy a Siemens akkori központtechnikája, az EWSD, az innováció csúcsán állt. Bár Magyarországon sokkal ismertebb volt az Ericsson, a Nortel, az ITT, de az EWSD kiemelkedő technológiája igen jó értékelést kapott. A másik ütőkártyánk az volt, hogy felajánlottuk: a rendszervásztó tender kapcsán

privatizáljuk a Telefongyárat. Tehát nem azt ajánlottuk fel, hogy joint venture-t alapítunk, nem azt ajánlottuk fel, hogy gyártást hozunk Magyarországra, hanem azt mondtuk, hogy egy nagy hagyományokkal rendelkező magyar távközlési céget privatizálunk, átveszünk, és itt valósítjuk meg a gyártást. Ezt az akkori döntéshozók pozitív ajánlatként értékelték. Tegyük hozzá, ez példaként szolgált több helyen is. Ez után a privatizáció után több nyugati cég keresett meg, hogy hogyan lehet Magyarországon privatizálni, hogyan lehet Magyarországra befektetni. Tehát ennek jelentős példaértéke is volt.

*Mint sikeres szakembert, biztos többször megkeresték mások is jobbnál-jobb ajánlatokkal. Mi az oka, hogy mégis a Siemest választotta, és hűség maradt a céghez?*

Valóban megkerestek páran. A döntésben fontos szerepet játszott, hogy a Siemensnél Magyarországon is tudtuk alakítani a cég helyi stratégiáját: tulajdonképpen a képviselői iroda alapításától egészen a egyes vállalt megalapításáig mindig irányítani tudtuk a jövőnket. Az anyaház részéről nem direktíva-rendszer volt, hanem az anyaházi tervek és célok mentén saját magunk alakítottuk ki a hazai stratégiát. Ez vonzó volt. Ismertem jó pár céget még a rendszerváltás előtt, akik képviselővel voltak jelen; mindig azt tapasztaltam, hogy sokkal kevesebb mozgásterük volt. Magunk alakítottuk a jövőnket, vállalkozásunkat. Ez az egyik, ami nagyon tetszett a Siemensben – a másik meg a széles tevékenységi kör. A világon szinte egyedülálló az a széles portfólió, amivel dolgozunk, tehát az erőműtechnikától kezdve a közlekedésen át, az orvostechnikát, az informatikát is beleértve. Az nagyon izgalmas dolog, ha az ember sokfajta vállalkozásban érdekelt, sokféle technikával ismerkedhet meg. Úgy gondolom, hogy számomra ez olyan vonzó volt, amit egyik ajánlattevő sem tudott megközelíteni.

*Az Ön irányítása alatt a cég profíljá, termékei, szolgáltatásai és azok megoldásai mennyit és miben változtak, módosultak? Egy időben több független gyáregység volt (vasút, kábel), most mi a helyzet? Hogy tartoznak össze?*

Magyarországon dinamikusan és folyamatosan változtunk. Kezdetben a Telefongyár privatizálása mellett még privatizáltuk a VIV Rt.-t, ahol közép feszültségű berendezéseket gyártottak, szereltek; megvettük továbbá a Magyar Kábel Műveket. Ezek kapcsán vásároltunk még jó pár kisebb céget, és kétségtelen, hogy a múltban a mainál sokkal több vállalat képezte a Siemens-csoportot. Később arra törekedtünk, hogy az aktivitásokat összevonjuk, méghozzá két csoportba. Az egyik csoportba azok taroznak, akik magyar piacra dolgoznak: ezek a Siemens Nemzeti Vállalat, azaz a Siemens Rt. köré csoportosuló vállalatok. Itt tulajdonképpen már homogén a cégcsoport, mert rajtunk kívül magyarországi forgalmazás szinte nincs.

A másik csoport, akik a globális munkamegosztásban dolgoznak a Siemens-világon belül. Ezen azt kell

érteni, hogy a Siemens anyaház különböző területeinek vannak Magyarországon gyártásaik. Ilyen a Siemens Transzformátor Kft., a Siemens Erőműtechnika Kft. és a Siemens PSE Kft. Ez utóbbi a Siemens rendszerfejlesztő házának hazai leányvállalata, ahol rendszereket fejlesztenek, és főleg szoftverfejlesztéssel foglalkoznak. Ezek a vállalatok nem a Siemens regionális rendszerében vagy a Nemzeti Vállalat stratégiai vezetése alatt működnek; nekik valamely Siemens-ágazat globális munkamegosztásában kell szerepet vállalniuk. Például a szolgáltatásokban, a telephelyben kihasználjuk azt az előnyt, ami a vállalatcsoport létéből adódik: egymásnak nyújtunk szolgáltatást és a telephelyben is megpróbálunk olyan értelmes megoldásokat találni, hogy ez mindannyiunk versenyképességét növelje. A változásokról egy könyvet lehetne írni. A lényeg az, hogy mi az összes termékgazdaságunkhoz megvalósítottuk Magyarországon azt a regionalizálást, ami szükséges és célszerű.

Mit értek ez alatt? Vannak olyan működési területeink, ahol a fejlesztéstől egészen az üzembe helyezésig, az üzemeltetésig szinte mindent csinálunk. Ezek olyan területek, ahol széles üzletről van szó. Azután vannak olyan területek, ahol 10 évben fordul elő egy üzlet, mint például egy erőmű vagy villamos teherelosztó, vagy jármű-területen a mozdonyok esetében, ahol a regionalizáció attól függ, hogy hosszú távú üzletet tudunk-e kötni, vagy csak esetleges az üzlet. Mindkét esetben megpróbálunk minden területen rögtön egy szervizt vagy szolgáltató bázist is felépíteni, és igyekszünk ezt a felhasználóknak mint versenyelőnyt eladni.

Ebben változott a cég profíljá, nemcsak árukereskedelmet kínálunk, hanem minden területen szolgáltatást, vevőszolgálatot is működtetünk. Utalhatok a távközlésre: a MATÁV-nak, MÁV-nak, nagyon sok kormányzati vevőnek üzemeltetünk is. Az informatikában is szinte az egész okmányirodai rendszerre kiterjedő üzemeltetői szerződést kötöttünk. A közlekedésben a behozott kétnormás mozdonyokra és dízelvonatokra is vállaljuk a karbantartást. A fő célunk az, hogy minden területen ne csak kereskedjünk, hanem szolgáltatassunk is: ez a strukturális változás lényege az elmúlt időszakban.

*Mint tapasztalhattuk, az elmúlt évtizedekben a világban hatalmas változások következtek be. Hogyan lehet ezekben a rohamosan változó időkben az innovációt, jó minőséget, és a tisztességes fogyasztói árat összeegyeztetni?*

Ez érdekes téma. A világban megváltoztak a vásárlási szokások, csak erről sokan nem vesznek tudomást. Az ügyfél egyre inkább azt tekinti: mi hoz neki a legtöbb hasznot. Régen a műszaki adatokat, azaz a teljesítményt vetették össze az árral, és a kettő hányadosa alapján rangsorolták a termékeket. Most már, azt hiszem, a legtöbb vevő azt nézi az összevetésben, hogy melyik termék hozza a legtöbb hasznot, és melyikkel tudja költségeit leginkább csökkenteni. Az innováció is erre felé halad. A mobiltelefonnál sem az a döntő már,



hogy egy készülék 10 évig jól működjön, hanem hogy kövesse az innovációs ciklust. Tehát a cél: olyan haszonorientált berendezéseket és rendszereket fejleszteni, amelyeknél összhangban van az életciklus és a minőség. A szolgáltatások pedig igazi hasznot hozzanak, és ne csak kedvtelésre – just for fun – szolgáljanak.

Ezzel kapcsolatban megindult a világban egy szabványosítási folyamat, ami természetesen hasznos volt, mert kialakultak készülék- és berendezés-csoportonként az azonos platformok. Viszont ma még ezek a platformok nem tudnak egymással kommunikálni. Jelenleg például mobiltelefonról nem lehet működtetni a háztartási berendezéseket vagy a tűzjelző, betörésvédelmi berendezéseket. Azt hiszem, a jövőben a jelenleg eléggé elszigetelt szabványok között közös platformokat kell találni, és ezáltal ki tudjanak alakulni interfészek, például a közlekedés-elektronika és a távközlés között, vagy a mobiltávközlés és a vasútbiztosítás között. Folynak ilyen fejlesztések, csak ezeket fel kellene gyorsítani, hogy az innovációs lehetőségeket jobban ki lehessen használni. Ebben látom a jövőt.

Örök kérdés, hogy mit értünk tisztességes fogyasztói ár alatt. Ezt szinte lehetetlenség meghatározni. Ahol nem a piaci mechanizmusok működnek, ott szerintem nem tisztességes a fogyasztói ár. Ugyanakkor, ahol viszont működnek a piaci mechanizmusok, ahol a kereslet és igény határozza meg a piaci árat, ott nincs tisztességtelen piaci ár.

*Mi az Ön véleménye szerint ma az etikus viselkedés? Mint ismeretes, erről a témáról Ön visszafogottan beszélt a „Piac és Info” Gerbeaud-házban tartott értekezletén...*

Nem visszafogottan beszéltem, hanem a kollegáimmal együtt először is azt kezdtük el boncolgatni, hogy mi az etikus viselkedés. Az etika tulajdonképpen a társadalom kultúrájához, szokásmechanizmusához alkalmazkodó viselkedés. Ezért például az etika Svédországban egészen más, mint Spanyolországban. Én úgy gondolom, az üzleti élet nem olyan, mint egy futóverseny, ahol elég egyszerűen lehet a valós teljesítményeket lemérni, mert a startvonalra állítanak mindenkit és eldördül a pisztoly, akkor azonos feltételek mellett győz a jobb. Az üzleti élet nem ilyen egyszerű és nem ilyen vertikális.

Van egy nagyon fontos szó, tényező, amit egy tender sem értékel, és ez a bizalom, ami általában egész máshol bújjik meg. Az üzleti verseny inkább hasonlítható a műkorcsolyázáshoz vagy tornászbemutatóhoz, mint egy futóversenyhez. Egyrészt természetesen az értékelésnél döntően azt „pontozzák”, hogy milyen versenyfeltételeket biztosít számukra az adott termék vagy rendszer, és ez nemcsak attól függ, hogy mit tud önmagában, hanem számít illeszkedése a környezethez és a bizalmuk a cégben. Ezért az üzleti etikában az az etikus viselkedés, amikor a vállalkozó azt nyújtja, amit a felhasználója elvár. Ha arra törekszünk, hogy a felhasználó versenyelőnyhöz jusson eszközeinkkel, ak-

kor véleményem szerint etikusan viselkedem. Ha viszont nem erre törekszünk, hanem a rövid távú üzlet kedvéért ezt figyelmen kívül hagyjuk, akkor ez etikátlan viselkedés, ami megbosszulja magát: nem tudni megtartani a vevőt. Vélhetőleg egy önszabályzó rendszer alakul ki, hiszen uniós tagságunkkal együtt egy kibővített versenypiacra kerültünk, ahol aki nem ezt követi, az előbb-utóbb ráfizet. Az embernek először azt kell tisztázni, hogy mi az üzleti etikus viselkedés. Ha a szállításaim és szolgáltatásaim használatával a vevő üzleti sikereket ér el, akkor szinte „meg van ágyazva” a további üzletnek. Legyünk őszinték: az első üzleten az ember mindig kevesebbet nyer – az igazi nyereség a folytatásban lehet. Ha ügyfelünk üzleti sikereket ér el, és az üzlet multiplikálásával nagyobb profithoz jut, akkor etikus, ha én ebből részesedve szintén növelni tudom nyereségemet.

*Hogyan védik ki a támadásokat? Előzetes információink vannak a kínai és lengyel sikerekről. A világ más kontinensein is terjeszkedik a Siemens. Mekkorra és milyen a konkurencia?*

A Coca Cola után a Siemens a második cég, amely a legtöbb országban jelen van, persze nem minden országban egyforma erővel. Vannak olyan országok, ahol bizonyos területeken a konkurencia jóval erősebb. Általános jelenség, hogy a több mint 100 üzleti terület mindegyikén, ahol dolgozunk világszerte, a konkurencia támad. Azokon a területeken, ahol lemaradtunk, vagy ahol nem törtünk be valamely országba, megpróbálunk erősíteni és a piaci részesedést növelni. Egy cég akkor igazán sikeres, ha az elsők közt van. Lehet vitatni, hogy az első kettő vagy három közt, de az elsők közt kell lennie ahhoz, hogy keresni is tudjon. A profit a jövő alapja, mert így jut pénz a fejlesztésekre. Tehát az egész megújulási ciklusnak a nyereség az alapja. A támadások éppen ezért teljesen természetesek. Itt a kivédés legjobb eszköze a támadás, vagyis hogy minél innovatívabb termékekkel, minél kedvezőbb áron jelenjünk meg, és versenyképes megoldásokat nyújtunk. Ez állandó küzdelem, ezért nem unalmas az üzleti élet.

Mondok egy idevágó példát. Magyarországon négy évvel ezelőtt a mobilpiacon a „futottak még” kategóriájába tartoztunk, ma pedig fej-fej mellett váltogatjuk a vezetői helyet a Nokiával. Ebben természetesen szerepet játszik a termék-innováció, és az is, hogy hogyan szolgáljuk ki a mobilszolgáltatókat, milyen szervizt, milyen támogatást adunk. Ez a sava-borsa az egésznek. Konkurencia mindig van, a harc pedig mindig akkor a legélesebb, amikor egy piac elkezd szűkülni. Ez nehezebben kezelhető helyzet, mert a versenytársak egymástól próbálják meg elvenni a piaci részeket. Ilyenkor az embernek a költségcsökkentésre kell koncentrálnia és arra, hogy a beszükkült piaci feltételekhez alakítsa az erőforrásait, például a munkaerő leépítése által.

A világ ilyen, – nem lineáris növekedésben, hanem hullámzások közepette élünk. Az embernek nemcsak növekvő piacon kell sikeresnek lennie, hanem csökkenő piacon is.

### **SIGNAL CODING OVER MULTIPLE ACCESS OR CHANNEL**

*Keywords: multiple access, combinatorics, code constructions, estimation methods*

Signal coding is analyzed over a multiple access OR channel where up to  $M$  users out of the total  $T$  are active. Some particular applications are also demonstrated. The underlying mathematical problem is the same as with  $M$ -uncovered set systems in extreme combinatorics. Lower and upper limits are provided for the  $n(T,M)$  minimum code word length and some code constructions are described.

### **ASPECT-ORIENTED PROGRAMMING**

*Keywords: aspect-oriented programming (AOP), interlacing aspects*

AOP is a favorable complementation to the more widely used object-oriented paradigm. This paper discusses some major AOP concepts based on the most popular implementation, the AspectJ approach. The problems of interlacing aspects are introduced then some possible opportunities are outlined.

### **IMPROVING THE PREDICTION OF QOS PARAMETERS WITH DETECTION AND REMOVAL OF OUTLIERS**

*Keywords: statistics, feature estimation, prediction of QoS parameters, outlier detection algorithms*

The real-time applications appearing on the Internet require the provision of appropriate quality-of-service parameters for users. To this end the accurate predictions of network traffic is necessary.

This can be achieved with the measurement of the traffic followed by the analysis of the results, then the association of a suitable model to the set of data and finally with the definition of the parameter. The accuracy of prediction is considerably decreased by certain outstanding samples (outliers) in the series of data. We developed therefore a tool which can quickly and automatically detect these outliers and then remove them.

### **LATIN AND MAGIC SQUARES FROM GAME APPLICATIONS TO SECURITY**

*Keywords: drawing squares, orthogonal system, summation rules*

The phenomenon of inventions developed in the field of games becoming later major chapters in mathematics is not unparalleled in the history of mathematics. This was the happened in the case of the theory of Latin squares, a nearly 300 years old matter in combinatorics. This study discusses the manifold, classic and quite modern applications and possibilities from the mathematical entertainment through data transmission of the 21st century to cryptography.

### **IMPACT OF OPTICAL FILTERS ON THE DIMENSIONS OF TRANSPARENT WDM NETWORKS**

*Keywords: wavelength multiplexing, diffraction grid, transparent optical coupler, bit error rate*

The extension (number of nodes) of optically transparent networks is limited since the information content of optical signals travelling through these "islands" must not decrease under a certain bit error rate. On the other hand, signal regeneration is a slow and expensive process. In my experiment add/drop multiplexers with three different optical filters were linked together forming a chain with ten nodes each and the resulting bit error rate deterioration was compared using a simulation program.

### **UNIFORM DESIGN OF INTELLIGENT SYSTEMS**

*Keywords: agent, evolution, game theory, rationality, plan making*

This paper argues that it is mostly the concept of "rationality" that could help in the unification of agent, game and evolution theories or the development of a comprehensive system specification policy assisting the design of intelligent systems. The opportunity of linking game theory with agent theory is demonstrated then the notion of player with limited optimum is introduced. This is followed by an overview of the consequences of linking these two theories. After game theory the evolution theory is discussed where the role of genetic algorithms is in focus. Finally the consequences of linking together all three theories is discussed.

### **STUDY OF THE TRANSMISSION OF INTERACTIVE TELEVISION APPLICATIONS ON THE DVB-T PLATFORM OF COMPANY ANTENNA HUNGÁRIA**

*Keywords: DVB, MHP, interactive television, data turnover, carousel*

Watching television is a part of our life which is now undergoing revolutionary changes. Digitalization allows not only the provision of a wider choice of programs but also offers more information for customers using interactive applications. In addition to the introduction of the system necessary to transmit interactive applications this article discusses also the data structures and bandwidth optimization techniques used for the transmission.

### **VLAN LABEL BASED CROSS-CONNECT FUNCTION IN VIDEO NETWORKS**

*Keywords: digital studio, Ethernet, transport network, label switching*

Digital signal processing advancing with giant strides during the past ten years is now playing a more and more important role in the field of advanced television. Ethernet functions integrated into new generation SDH equipment offer new ways for meeting the packet based video transport requirements in digital program exchange between studios or between studios and program distribution points.

This paper outlines the motivation behind and benefits of using VLAN label based cross-connect functions in a hypothetical Hungarian video transport network architecture.

# Contents

<i>RISK OR SECURITY</i>	1
<b>QUALITY AND SECURITY</b>	
<b>Sándor Győri</b> Signal coding over multiple access OR channel	2
<b>László Lengyel, Tihamér Levendovszky</b> Aspect-oriented programming	8
<b>László Kovács, Dorottya Vass, Attila Vidács</b> Improving the prediction of Quality of Service parameters with detection and removal of outliers	13
<b>Tamás Dénes</b> Latin and magic squares from game applications to security	19
<b>METHODS OF DIMENSIONING</b>	
<b>Ákos Szödényi</b> Impact of optical filters on the dimensions of transparent WDM networks	25
<b>Dániel László Kovács</b> Uniform design of intelligent systems	29
<b>BROADCASTING</b>	
<b>Szabolcs Krémer</b> Study of the transmission of interactive TV applications on the DVB-T platform of Antenna Hungária	39
<b>Péter Szegedi</b> VLAN label based cross-connect function in video networks	44
<b>Beatrix Havaska Nagy</b> Interview with Miklós Bendzsel, president of Hungarian Patent Office	49
Interview with Péter Hetényi, president and chief of Executive Board of Siemens Corporation	53

Cover: *Rapid changes can be traced not only in information technology but also in arts (Apple harvest by Imre Szabotka, 1930)*

---

## Szerkesztőség

HTE Budapest V., Kossuth L. tér 6-8.  
Tel.: 353-1027, Fax: 353-0451, e-mail: hte@mtesz.hu

## Hirdetési árak

1/1 (205x290 mm) 4C 120.000 Ft + áfa  
Borító 3 (205x290mm) 4 C 180.000 Ft + áfa  
Borító 4 (205x290mm) 4 C 240.000 Ft + áfa

## Cikkek eljuttathatók az alábbi címre is

BME Szélessávú Hírközlő Rendszerek  
Budapest XI., Goldmann Gy. tér 3.  
Tel.: 463-1559, Fax: 463-3289,  
e-mail: zombory@mht.bme.hu

## Előfizetés

HTE Budapest V., Kossuth L. tér 6-8.  
Tel.: 353-1027, Fax: 353-0451  
e-mail: hte@mtesz.hu

## 2004-es előfizetési díjak

*Hazai közületi előfizetők részére:*  
1 évre bruttó 31.200 Ft  
*Hazai egyéni előfizetők részére:*  
1 évre bruttó 7.000 Ft

## Subscription rates for foreign subscribers:

12 issues 150 USD,  
single copies 15 USD

www.hte.hu

Felelős kiadó: MÁTÉ MÁRIA  
Lapmenedzser: Dankó András

---

HU ISSN 0018-2028

Layout: MATT DTP Bt. • Printed by: Regiszter Kft.