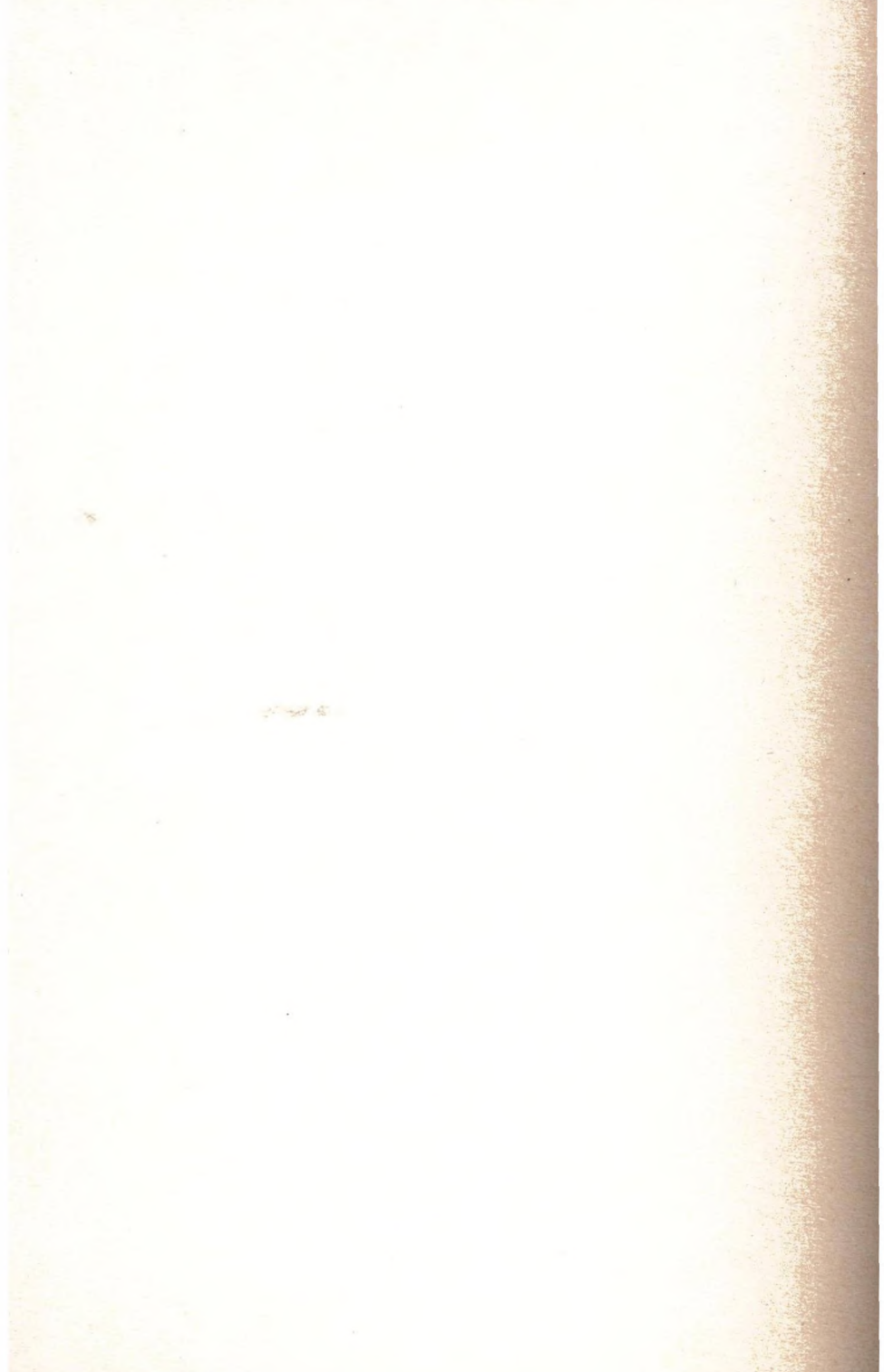


# SzámOkt 2004

XIV. SZÁMÍTÁSTECHNIKA  
AZ OKTATÁSBAN  
NEMZETKÖZI KONFERENCIA

14<sup>TH</sup> INTERNATIONAL CONFERENCE  
IN COMPUTER SCIENCE AND EDUCATION







ITA / 429

Erdélyi Magyar Műszaki Tudományos Társaság  
Hungarian Technical Scientific Society of Transilvania

## **SzámOkt 2004**

**Az emberközeli gép**

**XIV. Számítástechnika az oktatásban**  
Nemzetközi konferencia

14<sup>th</sup> International Conference  
in Computer Science and Education

Kolozsvár 2004. március 25-28.  
Cluj, March 25-28, 2004

**Kiadó**

Erdélyi Magyar Műszaki Tudományos Társaság

**Felelős kiadó**

Dr. Köllő Gábor

**Nyomdai előkészítés**

Prokop Zoltán

**Nyomtatás**

Incitato nyomda – Kolozsvár

Felelős vezető: Biró Á. Attila

**Támogatók**

Illyés Közalapítvány – Budapest

Pro Technica Alapítvány – Kolozsvár

**Descrierea CIP a Bibliotecii Naționale a României**

**INFORMATICA ÎN EDUCAȚIE. CONFERINȚĂ INTERNAȚIONALĂ  
(14 ; 2004 ; CLUJ-NAPOCA)**

**SzámOkt 2004 : XIV. Számítástechnika az oktatásban : nemzetközi konferencia, Kolozsvár 2004. március 25-28. = 14<sup>th</sup> International**

**Conference in Computer Science and Education : Cluj, March 25-28, 2004. – Kolozsvár: Erdélyi Magyar Műszaki Tudományos Társaság, 2004**

**ISBN 973-86097-8-X**

004:37

## **Konferencia szervező / Organising Institution**

Erdélyi Magyar Műszaki Tudományos Társaság Számítástechnika Szakosztálya  
Hungarian Technical Scientific Society of Transilvania,  
Department of Computer Science

## **Védnökök**

Dr. Arató Péter, a Neumann János Számítógép-tudományi Társaság elnöke  
Dr. Pusztai Kálmán, a Kolozsvári Műszaki Egyetem  
Automatika és Számítástechnika Karának dékánja

## **Konferencia elnök / Chairman**

Dr. Sebestyén György

## **Tudományos bizottság / Scientific Committee**

Dr. Arató Péter, Neumann János Számítógép-tudományi Társaság  
Dr. Pusztai Kálmán, Kolozsvári Műszaki Egyetem  
Dr. Dávid László, Sapientia – Erdélyi Magyar Tudományegyetem  
Dr. Keresztes Péter, Széchenyi István Egyetem  
Dr. Sebestyén György, Kolozsvári Műszaki Egyetem  
Dr. Haller Piroska, Petru Maior Egyetem  
Borbély Endre, Budapesti Műszaki Főiskola  
Kovács Lehel, Babeş-Bolyai Tudományegyetem  
Somodi Zoltán, Kolozsvári Műszaki Egyetem

## **Szervező bizottság / Organising Committee**

Brem Walter  
Horváth Erika  
Kovács Enikő  
Matekovits Hajnalka  
Pap Tünde  
Prokop Zoltán

## A konferencia programja

### 2004. március 25., csütörtök

Helyszín: Bethlen Kata Diakóniai Központ

- 17,00 – 21,00 bejelentkezés  
19,00 – 21,00 vacsora a Bethlen Kata Diakóniai Központban

### 2004. március 26., péntek

- 9,00 indulás az egésznapos kirándulásra  
a Bethlen Kata Diakóniai Központ elől  
útvonal: *Kolozsvár – Torda – Tordai Sóbánya – Torockó –  
Torockószentgyörgy – Kolozsvár*  
ebéd Torockón a Tóbiás-házban

### 2004. március 27., szombat

Helyszín: Bethlen Kata Diakóniai Központ

- 8,15 – 9,15 bejelentkezés  
9,15 a konferencia megnyitója  
9,30 plenáris előadások  
11,00 kávészünet  
11,30 plenáris előadások  
13,00 ebédszünet  
15,00 szekcióelőadások  
16,40 kávészünet  
17,00 szekcióelőadások  
19,30 állófogadás

### 2004. március 28., vasárnap

Helyszín: Báthory István Elméleti Líceum

- 9,30 – 12,30 előadások, laborgyakorlatok  
13,00 ebéd

## Plenáris előadások

ülésvezető: Dr. Sebestyén György

- 9,30 Dr. Németh Géza  
BME, Távközlési és Médiainformatikai Tanszék  
*A beszédtechnológia alkalmazási lehetőségei  
a hatékony ember-gép kapcsolatban*
- 10,00 Borbély Endre  
BMF, Kandó Kálmán Villamosmérnöki Főiskolai Kar,  
Híradástechnika Intézet  
*e-kereskedelem, e-vásárlás*
- 10,30 Dr. Giese Piroska  
Központi Fizikai Kutató Intézet, RMKI  
*Videokonferencia alkalmazása a távmunkában  
végzett európai magfúzió-kutatásban*
- szünet
- 11,30 Dr. Selinger Sándor  
Gábor Dénes Főiskola, Erdélyi Konzultációs Központ  
*Távoktatás a világhálón. Az internet alapú oktatás  
egy kommunikációs folyamat*
- 12,00 Dr. Kása Zoltán  
BBTE, Matematika és Informatika Kar  
*Hogyan viselkedjünk az Interneten!*

**Szekcióelőadások**  
**Számítástechnika és oktatás**

ülésvezető: Dr. Kallós Gábor

- 15,00 Szalay Istvánné  
Központi Fizikai Kutató Intézet,  
RMKI Számítógép Hálózati Központ  
*PhysHun – a magyar fizikus szakmai közösség  
tudásvagyonának integrálása  
a nemzetközi PhysNet rendszerbe*
- 15,20 K. Princz Mária  
Debreceni Egyetem, Műszaki Főiskolai Kar  
*Információkeresési stratégiák a weben  
és tanításuk tapasztalatai*
- 15,40 Bujdosó Gyöngyi  
Debreceni Egyetem, Matematikai és Informatikai Intézet,  
Komputergrafikai és Könyvtárinformatikai Tanszék  
*A szöveg- és kiadványszerkesztés oktatásának  
néhány tapasztalata*
- 16,00 Dr. Kallós Gábor  
Széchenyi István Egyetem, Győr  
*Elektronikus jegyzet  
a párhuzamos programozás oktatásához*
- 16,20 Kátai Zoltán  
Sapientia EMTE, Marosvásárhely  
*Howyan tanítsuk a programozási technikákat?*

szünet



## Számítástechnika és oktatás

ülésvezető: Nagy-Imecs Vilmos

- 17,00 Farkas Károly  
Szolnoki Főiskola és Budapesti Műszaki Főiskolák,  
Neumann Informatika Kara  
*Mit adott nekem a Logo?*
- 17,20 Törtély Éva  
Templom Téri Általános Iskola, Pilisvörösvár  
*Játékos Informatika -- Logo és programozás*
- 17,40 Nagy Imecs Vilmos  
Eötvös József Mezőgazdasági Szakközépiskola,  
Székelyudvarhely  
*Számítástechnika tanár = Internet pedagógus !?  
(vagy csak a tanár dilemmája)*
- 18,00 Máthé Zsolt, Görög Levente,  
Komáromi Lóránd, Szilágyi Sándor Miklós  
Kolozsvári Műszaki Egyetem,  
Sapientia EMTE, Marosvásárhely  
*Mesterséges intelligencia társasjátékokban*

## Korszerű számítástechnikai fejlesztések

ülésvezető: Kovács Lehel

- 15,00 Dr. Dávid László  
Sapientia EMTE, Marosvásárhely  
*Optimális vezérlési stratégia megvalósítása mesterséges neuronhálók segítségével*
- 15,20 Bakó László, Brassai Sándor Tihamér  
Sapientia EMTE, Marosvásárhely  
*Fejlett neuronmodellek szimulációja és megvalósítása*
- 15,40 Kovács Lehel István  
Babeş-Bolyai Tudományegyetem,  
Matematika és Informatika Kar  
*A dinamikus öröklődés*
- 16,00 Dr. Héray Tibor  
Széchenyi István Egyetem, Győr  
*A mikroprocesszoros rendszerek megjelenésének hatása a vasúti irányítórendszerekre*
- 16,20 Kovács Barna  
Papiu Ilarian Nemzeti Kollégium, Marosvásárhely  
*Prediktor-korrektor és konzervatív integrátorok az égi mechanikában - A korlátozott háromtest-probléma integrálása prediktor-korrektor és konzervatív integrátorral*

szünet

## Számítógép és kommunikáció

ülésvezető: Dr. Haller Piroska

17,00 Dr. Haller Piroska  
Petru Maior Egyetem, Marosvásárhely  
*Multimédiás adatátvitel teljesítményelemzése  
processz algebra segítségével*

17,20 Budai László  
Sapientia EMTE, Marosvásárhely  
*Tűzfalak – működési elvek, tervezésük  
és működtetésük egy oktatási hálózatban*

17,40 Somodi Zoltán  
Kolozsvári Műszaki Egyetem  
*DoS támadások elleni  
védelmi mechanizmusok az IPv6-ban*

18,00 Dr. Sebestyén Pál György  
Kolozsvári Műszaki Egyetem  
*Ipari kommunikációs hálózatok*

18,20 Varjasi Norbert  
Széchenyi István Egyetem, Győr  
*A mobil eszközök és az objektum-orientált  
programozás: A Java2ME*

2004. március 28., vasárnap

9,30 – 12,30

Dr. Benedek Dezső  
Georgia-i Tudományegyetem  
*Online nyelvtanítás egyetemi szinten*

Dr. Kovács Zoltán  
BBTE, Pszichológia és Neveléstudományi Kar  
*Az AEL program bemutatása*

## **e-kereskedelem, e-vásárlás**

### **e-sales, e-purchase**

**Borbély Endre**

okl. villamosmérnök, okl. mérnöktanár, mestertanár  
Budapesti Műszaki Főiskola  
Kandó Kálmán Villamosmérnöki Főiskolai Kar  
Híradástechnika Intézet

#### **Abstract**

*In the past few decades, years, the forms of shopping have greatly changed all over the world. More and more people would do their shopping through the Internet rather than go into various shops and stores.*

*In Hungary, too, several shoppers prefer sitting in an armchair and doing the purchases*

*through e-stores. In 2002 over HUF 4.5 billion were spent by use of e-purchase.*

*A wide range of products, various benefits, mail service, discount prices are the advantages that increase the volume of e-sales. Furthermore, aspects of economics, market, policy, regulations, consumer protection, as well as the rights of customers and the duties of salespeople are considered as matters of great importance.*

Az utóbbi évtizedekben, években az egész világon a vásárlási szokások sokat változtak. Egyre többen nem az utcai boltokban, hanem a világháló adta lehetőségeket használják beszerzéseiknél.

Magyarországon is sokan karosszékben intézik vásárlásaikat. 2002-ben több mint 4.5 millárd forintért vásároltak az e-boltokban.

A széles termékválaszték, a különböző kedvezmények, a posztai szállítás, fizetési kedvezmények is fokozzák az e-boltok forgalmát. A gazdasági, a piaci, a politikai, a jogi, fogyasztóvédelmi, a vásárló jogai, az eladó kötelességei is fontos kérdések.

#### **Széles termékválaszték**

Világosan látszik, hogy a piaci kezdetekre jellemző szűkös termékválaszték után, ma már szinte nincs olyan termék kategória, amely ne lenne megtalálható az interneten is.



Az online boltok túlnyomó része száz és kétezer közötti terméket forgalmaz, de itt is jellemző a kínálat koncentrációja, néhány nagy webáruház adja a teljes termékkínálat több mint felét.

### **Kedvezmények**

Az üzletek 15%-a nyújtott állandó jelleggel árkedvezményt termékeire, amelynek jellemző mértéke 6% volt. Az állandó kedvezmények mellett időszakos árkedvezményt is adnak a vásárlás növelése érdekében.

### **Postai szállítás, készpénzes fizetés**

Az online boltokban leggyakrabban a postai szállítást választhatjuk (78%), de lehetőség van futárszolgálat igénybevételére (29%) és néhány esetben (22%) a boltok is vállalják a kiszállítást. A boltok átlagosan egy hetes szállítási határidőt, bankkártyával egyelőre csak a boltok alig egytizedében fizethetünk.

### **Javuló tájékoztatás**

Magyarországon is terjed a megfelelő szintű tájékoztatás gyakorlata, azonban ezen a téren még van mit fejlődnie a magyar e-boltoknak. Már található sugó, és ugyanekkor az aránya azoknak a boltoknak is, ahol a fizetési feltételekről tájékozódhatunk a vásárlást megelőzően. A garanciális feltételekről és az adatvédelemről is olvashatunk. A marketing célú adatgyűjtés nem jellemző. A vásárlástól való elállás jogáról olvashatunk. Az e-boltok ügyfélszolgálatára elérhető e-mailen keresztül és kérdéseinkre általában már néhány órán belül pontos tájékoztatást kaphatunk. Az üzletek működtetnek telefonos ügyfélszolgálatot. Mindent figyelembe véve a széles termékkínálat, az árkedvezmények, vásárló számára biztonságot nyújtó szállítási és fizetési feltételek, valamint a tájékoztatás és a szolgáltatás minősége alapján elmondható, hogy a online beszerzése akár otthonról, a karosszékéből is kényelmesen megoldható.

### **Online vásárlás**

Az internet felhasználói táborának növekedésével párhuzamosan a hagyományos internetes tevékenységek mellett – mint például az elektronikus levelezés vagy a világháló böngészése – az elmúlt években Magyarországon is fejlődésnek indult az online vásárlás. A világhálón értékesített termékek és szolgáltatások köre igen széles: a kézzelfogható, fizikai termékek dominálnak, de megtalálhatók a szolgáltatások, valamint a digitális, információs

javak is. Az online áruházak a világháló által nyújtott lehetőségeket használják ki, melyek mind a vásárlók, mind az eladók számára számos előnyt rejtenek magukban. Ilyen előny például az eladó számára, hogy nem kell bolthelyiséget fenntartania, a vásárló szempontjából pedig a helytől és nyitva tartási időtől független, kényelmes vásárlási lehetőség.

### **E-boltok Magyarországon**

A hazai elektronikus boltok száma és a boltok által lebonyolított forgalom évről-évre nő. Jelenleg megközelítőleg 300 e-bolt működik Magyarországon. A GKI Gazdaságkutató Rt. által végzett felmérés szerint az összes bolt 2002-ben 4,5 milliárd forintot bonyolított le, az online vásárlók aránya az internethasználók körében 6%-ra emelkedett

Az Egyesült Államokban, ahol az elektronikus vásárlás kultúrája igen fejlett, az online vásárlók általában az alábbiakat jelölik meg az internetes vásárlás indokaként:

- az utazási idő megtakarítható,
- nincs kötött nyitva tartási idő,
- az ünnepi tumultus elkerülhető,
- megvan a lehetőség arra, hogy kedvezőbb árakat találjunk,
- könnyebb megtalálni a termékeket a világhálón,
- fizikai boltban nem árusított termékek is megvásárolhatók online.

Magyarországon az internetezők az alábbi szempontokat tartják a legfontosabbnak:

- biztonság,
- árkedvezmények,
- gyorsaság,
- kényelem,
- széles körű termékinformációk,
- 24 órás elérhetőség.

### **Üzleti modellek**

Az online boltokat több csoportra oszthatjuk aszerint, hogy rendelkeznek-e hagyományos üzlettel, illetve kereskedelmi háttérrel:

- *offline háttérrel rendelkező e-bolt:* Ebbe a kategóriába azok az e-boltok tartoznak, amelyek hagyományos, offline áruházi jelenléttel és kereskedelmi háttérrel rendelkeznek.
- *tisztán internetes bolt:* A tisztán internetes boltok kizárólag az interneten értékesítik termékeiket, hagyományos bolttal nem rendelkeznek.



- *önálló e-bolt*: Az önálló, független e-boltok kizárólag saját termékeket kínálnak online áruházukban.
- *elektronikus bevásárlóközpont*: Ebbe a kategóriába azok az online áruházak tartoznak, amelyek több kereskedő számára biztosítanak internetes értékesítési felületet.

## Termékkínálat

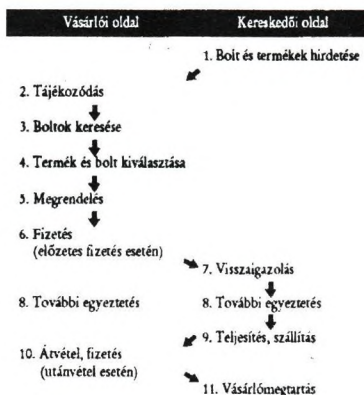
A magyar online boltok kínálatja gyakorlatilag a hagyományos áruházakban kapható legtöbb termékkörre kiterjed. A legnagyobb kínálattal rendelkező termékkategóriák a következők:

- könyv,
- kép- és hanghordozó (CD, DVD, audió- és videókazetta, hanglezem),
- hardver, szoftver.

A felsorolt három termékkör adja a boltok kínálatának több mint négyötödét.

## Az internetes vásárlás folyamata

Az elektronikus boltokban történő vásárlás – akár csak a telefonos rendelés – a termék kiválasztását és megrendelését, a kiszállítást és az áru átvételét foglalja magába. Ebben a folyamatban a vásárló és a kereskedő, illetve a kereskedővel kapcsolatban lévő egyéb piaci szereplők – mint például futárszolgálat, posta, bank – vesznek részt. Az online vásárlás folyamata több lépésre bontható, ezt szemlélteti az alábbi ábra:



## Az online és offline vásárlás kapcsolata

A világhálón és a hagyományos boltokban történő információgyűjtés és vásárlás sok esetben egymásba fonódik, a vásárlók az internetet és a hagyományos boltokat egyaránt felkeresik, információt gyűjtenek mindkét helyről. Összetettebb termékek esetén nem ritka az alábbi, mind az elektronikus, mind a hagyományos boltokat érintő vásárlási folyamat. A vásárlás szakaszai:

1. Tájékozódás

➔ 2. Megtekintés, kipróbálás

3. Vásárlás



## Információgyűjtés az interneten

Az e-boltok jellemzője, hogy a termékek kiválasztásától a megrendelésig – és néhány esetben a fizetésig – a vásárlási folyamat a világhálón keresztül zajlik. Számos honlap van azonban, amelyen ugyan nem lehet a rendelést leadni, de a termék- és árinformációk begyűjthetők.

Azok a jó e-boltok, melyekben a következő módon lehet vásárolni:

- kosárral lehet vásárolni, vagyis a boltban katalógusból közvetlenül kiválasztható a termék, a kosárba tett termékek pedig együttesen megrendelhetők,
- a megrendelés teljes folyamata az interneten történik,
- bárki megkötés és korlátozás nélkül használhatja és vásárolhat,
- a honlap és az ott elhelyezett információk, adatok magyar nyelvűek és
- az üzlet termékválasztéka részben vagy egészben beleillik a tipikus karácsonyi termékek körébe.

## Céginformációk

A vásárlás során a vevő és az eladó nem kerül személyes kapcsolatba egymással, ezért a vásárlói bizalom megteremtésében, a problémák, reklamációk kezelésében fontos szerepe van annak, hogy a vevő pontosan tudja kivel áll kapcsolatban. Információ kérés vagy reklamáció esetén e-mailen vagy telefonon elérhető kell legyen az ügyfélszolgálat.

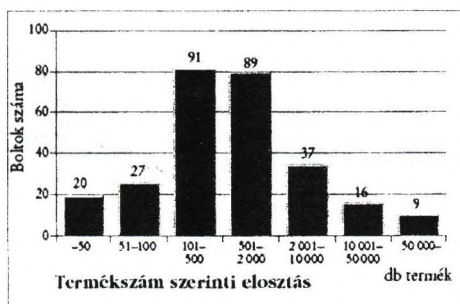
## Üzletszabályzat, súgó

A vásárlás feltételeit, körülményeit a vevő jogait, a kapcsolódó fogalommagyarázatokat az e-boltok többsége üzletszabályzatok, tájékoztatók, általános tudnivalók formájában, összegyűjtve ismerteti a vásárlókkal, vagyis ezek az információk egy helyen megtalálhatók.

## Termékinformációk

A vizsgálat eredményei igazolták, hogy az internet, az ott található információk mennyisége miatt ideális eszköz a vásárlást megelőző információgyűjtés egyszerű és hatékony lebonyolítására.

## Termékválaszték



A virtuális üzletek termékínálata változatos képet mutat. Az interneten egyaránt megtalálhatók a néhány terméket forgalmazó kis szaküzletek és a hatalmas áruválasztékkal rendelkező nagyáruházak.

## Kedvezmények

Az e-boltok egy része valamilyen konkrét kedvezménnyel próbálja vásárlásra ösztönözni a látogatókat.

## A honlapok használhatósága

Amerikai kutatások szerint [7] az e-boltok felhasználóbarát kialakítása nagyban befolyásolja azok sikerességét, hiszen nem elég sok információt és gazdag áruválasztéket kínálni a látogatóknak, de a termékszámmal összhangban, áttekinthető

E-boltok termékínálata					
Termékkör	Termékkör			Termék	
	Boltok száma (db)	%	Termékszám (db)	%	Átlagos termékszám
Kétség, folyóirat	78	13,15	768.080	51,29	9 847,2
Zenei cd, dvd, video (kép, hanghordozók)	79	13,32	483.279	32,46	6 142,8
Hardver, szoftver	84	14,7	48.463	3,24	576,9
Indószet	31	5,23	41.834	2,79	1 349,5
Exozita	19	3,20	28.575	1,91	1 503,9
Élelmiszer	28	4,72	27.456	1,83	980,6
Szabóéj (belső, benne burkolás)	32	5,40	21.203	1,42	662,6
Ruházati cikk (ruha, cipő)	25	4,22	10.685	0,71	427,4
Telefon és tartozéka	22	3,71	10.627	0,71	483
Ják	22	3,71	10.015	0,67	455,2
Szórakoztató elektronika	33	5,56	7.717	0,52	233,8
Sport cikkek, camping cikkek	20	3,37	6.683	0,45	334,2
Szépéjéj, kozmetika	20	3,37	5.845	0,39	292,3
Fotó	22	3,71	4.997	0,33	227,1
Bútor	18	3,04	4.572	0,31	254
Háztartási készülékek	19	3,20	4.548	0,30	239,4
Ajándék	19	3,20	4.357	0,29	229,3
Gépjármű, autósalkatrész	6	1,01	3.216	0,21	536
Ön-ápolás	11	1,85	1.994	0,13	181,3
Kapcsolódási alkatrészek	5	0,84	1.502	0,10	300,4
Összesen	593*	100,00	1.497.648	100,00	2.525,5

## Szállítás, fizetés

Az e-boltoknál a leggyakoribb kiszállítási mód a postai szállítás, és fizetésnél az utánvét.

Szállítási módok megoszlása			Fizetési módok gyakorisága		
Szállítási mód	Gyakoriság		Fizetési mód	Gyakoriság	
	db	%		db	%
Futár	85	29	Utánvét	252	87
Posta	226	78	Bankkártya	26	9
Bolt szállítja ki	63	22	Készpénz, boltban	92	32
Helyben lehet átvenni	101	35	Átutalás	44	15
Nincs információ	26	9	Csekk	12	4
			Nincs információ	21	8

## Kiegészítő szolgáltatások

Jellegző kiegészítő szolgáltatások	Gyakoriság	
	db	%
Termékfigyelés	16	6
Rendeléskövetés	20	7
Termék összehasonlítás	4	1
Hírvétel	114	39

## Fogyasztóvédelem az interneten

A fogyasztók védelméről szóló jogszabályok az online vásárlásra ugyanúgy vonatkoznak, mint a hagyományos kereskedelemre. A jótállási, garanciális és egyéb fogyasztóvédelmi jogok természetesen online vásárlás esetén is megilletik a vásárlót.

## Szerződéskötés online vásárlás esetén

Internetes vásárlás során az eladó és a vásárló nincs közvetlen kapcsolatban egymással, a termékek megrendeléséről szóló szerződés távol lévő felek között jön létre. A szerződéskötés folyamata az alábbi szakaszokra bontható:

- Bolt ajánlata vagy felhívása
- Megrendelés
- Visszaigazolás



## Az eladó kötelezettségei

Az alábbiakban röviden összefoglaljuk az internetes kereskedők néhány fontosabb kötelezettségeit. Vásárlás előtt érdemes megnézni, hogy a bolt ezeknek a kötelezettségeinek eleget tesz-e:

- Cégadatok
- Vásárlási tájékoztató
- Szerződési feltételek
- Visszaigazolás:

Szállítási költségek				
Boltlap neve	Ára	Vásárlás szállítási és csomagdíjai költség	Fixített végösszeg	Különbözet
Alexandra ONLINE	2 890	734	3 245	-379
Gardtop	890	1 125	2 115	0
eBok Műszaki Áruház	6 756	1 000	7 756	0
Fotomet	27 990	0	27 990	0
FotóMárka	25 500	0	25 500	0
Gift Rt.	4 857	1 180	6 244	211
Groby Elektronizációk	1 199	475	1 674	0
Haldokló	1 890	1 055	3 045	0
HaziPatika.com	5 890	900	6 790	0
Jások-Market.hu	3 696	n.a. <sup>2</sup>	4 551	0
NetPac Online Áruház	3 790	890	4 680	0
Office Depot	274	2 000	1 329	-98 <sup>3</sup>
Otto	1 890	935	2 925	0
Wesel-Webshop	4 811 <sup>4</sup>	0	4 811	0

## A vásárló jogai

Az alábbiakban a vásárlók három fontosabb, az internetes vásárlásra vonatkozó jogát mutatjuk be. Ezek a jogok minden vásárlót megilletnek tetsszöleges vásárlás esetén.

- *Meg nem rendelt termékek*
- *Megrendeléstől való elállás joga*
- *Vásárlástól való elállás joga:*

## Mire figyeljünk a vásárláskor?

Online vásárlásnál – akárcsak a hagyományos üzletekben történő vásárlásnál – érdemes odafigyelni néhány dologra

- *Nézzük meg a bolt adatait, elérhetőségét*
- *Olvassuk el az üzletszabályzatot*

- *A rendelés előtt tájékozódjunk a szállítási és fizetési feltételekről*
- *Győződjünk meg személyes adataink védelméről*
- *Mielőtt rendelésünket véglegesítjük, győződjünk meg a teljes költségről*
- *Őrizzük meg a rendelés adatait*

### **Praktikus tanácsok**

- *Az internetes vásárlás nem feltétlenül olcsóbb, csak más előnyei vannak a hagyományos vásárláshoz képest.*
- *Ha információt gyűjtünk a legjobb megoldás az internet*
- *Az interneten rengeteg termékinformáció található*
- *Nem a méret számít*
- *Ha valami nem világos kérdezzünk*
- *Célszerű munkahelyi címet adni, hogy napközben kiszállíthassák a terméket.*
- *Rendeljük meg idejében a terméket*
- *Használjuk ki a kedvezményeket*

### **Irodalomjegyzék:**

- [1.] 17/1999. (II.5.) Kormányrendelet a távollévők között kötött szerződésekről
- [2.] 2001. évi CVIII. törvény az elektronikus kereskedelmi szolgáltatások, valamint az információs társadalommal összefüggő szolgáltatások egyes kérdéseiről ([http://www.complex.hu/kzldat/t0100108.htm/t010\\_0108.htm](http://www.complex.hu/kzldat/t0100108.htm/t010_0108.htm))
- [3.] CyberAtlas: E-tailers Will See Green  
[http://cyberatlas.internet.com/markets/retailing/article/0,,6061\\_3105491,00.html](http://cyberatlas.internet.com/markets/retailing/article/0,,6061_3105491,00.html)
- [4.] GKI Gazdaságkutató Rt., Az online áruházak helyzete, 2002. IV. negyedév
- [5.] GKI Gazdaságkutató Rt., Az online kereskedelmi áruházak helyzete, 2003. II. negyedév
- [6.] NRC-TNS, E-kereskedelem és interaktív szolgáltatások, 2003.
- [7.] Jakob Nielsen: E-Commerce User Experience, 2001.
- [8.] Hírlevél 2. HTE 2004. február

### **Fogalommagyarázat**

- *adatkezelési nyilvántartási azonosító: Az Adatvédelmi Törvény rendelkezései szerint minden adatkezelést be kell jelenteni az adatvédelmi nyilvántartásba, a nyilvántartási számot az adatkezelő (jelen esetben a kereskedő) ekkor kapja.*



- *B2C (Business to Consumer)*: A B2C rövidítés a fogyasztói elektronikus kereskedelmet, e-kiskereskedelmet jelöli. Termékek vagy szolgáltatások interneten történő értékesítése a fogyasztók felé. Az e-boltok, e-áruházak, netplázák a B2C piac szereplői.
- *bankkártyás fizetés*: A bankkártyás fizetés lényege, hogy a vásárló a megrendelés időpontjában, bankkártya adatainak megadásával a megrendelt áruk ellenértékét kiegyenlíti.
- *cookie*: A cookie, vagy más néven süti, a kereskedő által a vásárló számítógépére küldött rövid adatsor, amely lehetővé teszi, hogy a későbbi vásárlások során az e-bolt a vevőt automatikusan felismerje és egyéb kényelmi szolgáltatást biztosítson számára. A cookie-kat csak az a honlap tudja olvasni, amelyik elhelyezte.
- *e-bolt*: Termékek értékesítésével foglalkozó internetes honlap. Az e-boltokban a vásárlás teljes folyamata az információnyújtástól a termék megrendeléséig az interneten keresztül valósul meg.
- *elállás joga*: A vásárló a megrendelt terméket a termék átvételétől számított 8 munkanapon belül visszaszolgáltathatja a kereskedőnek, a kereskedő pedig köteles a termék árát harminc napon belül visszatéríteni. A vásárlástól való elállást a vevőnek nem kell indokolnia. A termék visszaszolgáltatásának díját és a termék nem rendeltetésszerű használatából eredő károkat a vásárlónak kell állnia.
- *futárszolgálat*: Csomagok, küldemények kézbesítésére, célbajuttatására vállalkozó társaságok, akik jellemzően a postai kézbesítésnél rövidebb határidővel vállalják a szállítást.
- *hírlevél*: A boltok gyakran kínálják fel a vásárlóknak, hogy e-mailen rendszeresen tájékoztatják őket új termékeikről, akcióikról.
- *hűségprogram*: Több bolt is kedvezményeket biztosít gyakori vásárlóinak, törzsvásárlóinak. Egyes boltoknál a vásárlók hűségpontokat, lojalitás pontokat gyűjthetnek, amiket később termékek vásárlására fordíthatnak, máshol a vásárlók a boltnál elköltött összeg alapján különböző árkedvezményekre jogosultak.
- *kívánságlista*: Néhány e-bolt oldalán a vásárlók a termékeket a kosáron kívül saját kívánságlistájukra is helyezhetik. A kívánságlistába tett termékek adatait a bolt tárolja, így később bármikor megrendelhetők.
- *kosár*: Az internetes kosár lényegét tekintve hasonló a hagyományos bevásárlókosárhoz: online vásárlás során a kiválasztott termékeket lehet beletenni és kivenni belőle. A kosár folyamatosan jelzi a vásárló számára, hogy mit, milyen mennyiségben és milyen összegben válogatott össze a termékek böngészése során, a vásárlás végén a kosárban található termékek alapján történik a megrendelés.

- *partnerprogram*: A partnerprogram keretében az áruház termékeit kisebb-nagyobb honlapok is hirdetik, akik bizonyos százalékban részesednek az általuk generált eladásokból.
- *regisztráció*: Számos e-boltnál alkalmazott megoldás, melynek lényege, hogy a vásárló csak egyszer adja meg adatait (szállítási, számlázási adatok, beállítások stb.), minek után kap egy azonosítót és egy jelszót. További vásárlások során pedig elég megadnia ezt az azonosító-jelszó párost.
- *rendeléskövetés*: A rendelés állapotának nyomon követésével a vásárló megtudhatja, hogy a feladott rendelése megérkezett-e az ügyfélszolgálathoz, a termék raktáron van-e, mikor szállítják.
- *termékfigyelés*: A boltok egy része a vásárló által megadott feltételek alapján értesítést küld új termékeiről.
- *utánvétel*: Magyarországon az elektronikus vásárlások során leggyakrabban alkalmazott fizetési mód. Lényege hogy a vevő a termék átvételekor, utólag fizet a megrendelt áruért. A vételárát a terméket kiszállító kézbesítőnek, postásnak, futárnak kell átadni.

## Távoktatás a világhálón

Az Internet alapú oktatás egy kommunikációs folyamat

Distance Learning, a Communication Tool

**Dr. Selinger Sándor**

Gábor Dénes Főiskola

### Abstract

*Looking at Internet communications from an educational technology point of view, the following years pose the problem of the design of individually tailored learning modalities. We have to find such possibilities and methodologies that make it possible to everyone to acquire culture in a way that is individual, has the desired depth and is appropriate to their lifestyle.*

A XXI. század első évtizedei minden bizonnyal a tanuló társadalom koraként kerülnek majd a történelemlétkönyvekbe.

Oktatástechnológiailag megközelítve a kérdést, az elkövetkező évek feladata az egyénre szabott tanulási módozatok megtervezésének problematikáját vetítik elénk. Olyan lehetőségeket és módszertanokat kell megtalálnunk, melyek lehetőséget teremtenek mindenki számára, hogy a maga élethelyzetének megfelelően, önállóan a maga tervezte mélységben, és idő alatt vehesse birtokba a kultúra egészét.

A tudásalapú társadalom kialakítása megköveteli a tudáshoz való hozzáférésre szolgáló eszközök biztos használatát, a széles körben elterjedő számítógép-ismeretet, kommunikációs eszközök használatára irányuló készséget és a szakirányú képességfejlesztést.

Ezzel párhuzamosan mintegy „szükséges velejáró” megjelenik egy újfajta analfabetizmus veszélye is, mely alatt a kommunikációs eszközök használatának képtelenségét értjük.

Az információs társadalom a tudás folyamatos gyarapításának társadalma. A tudás, a jólét legfontosabb forrása, előállítására a gazdaság legfőbb célja. Az info-kommunikációs társadalomban a tudás jellege megváltozik, multimédiássá, transzdiszciplinárisá és gyakorlatiassá lesz.

A tudás az egyedüli olyan erőforrás, amelyik a használat által gyarapodik.

A tudás definiálásával, materializálásával, anyagi erőforrássá való konverziójával a '90-es évek közepén kezdtek el intenzíven foglalkozni a tudósok. A cél a tudással való magabiztos bánásmód, versenyelőny-teremtés egy



úgy nevezett tudásmenedzsment segítségével. A tudásmenedzsmentről azonban egyelőre még nagyrészt akadémikus viták zajlanak világszerte.

Ennek az egésznek szerves, kézzelfogható és már működő része az e-learning, a modern kommunikációs eszközök használatán alapuló oktatás ill. tanulás.

Az e-learning a képzés, az oktatás új formája, melynek kialakulását a technológiai fejlődés tette lehetővé. Az e-learning így technológia és tudás találkozását jelenti, másképpen fogalmazva: ebben az oktatási formában nem lehet szétválasztani a tudás terjesztését szolgáló technológiai keretrendszert és magát a tananyagot.

Az e-learning a képzés egy olyan új formája az oktatásnak, melynek alapja az internetes technológia.

Az Internet alapú oktatás egy kommunikációs folyamat.

Az e-learning fejlődését az egyre kifinomultabb internetes technológiák (pl. HTML, Flash, Java, Java Script) megjelenése tette lehetővé. Kezdetben egy e-learninges web-hely nem volt egyéb, mint egyszerű, statikus weboldalak gyűjteménye, amely valamilyen témában jegyzetszerűen, esetleg tartalomjegyzékkel kiegészítve tartalmazta a szükséges információkat.

Jelenleg is nagyon sok ilyen oktatási anyagot lehet találni a világhálón.

A szolgáltatások egyik ága, az új technológiák elterjedésével az egyre látványosabb, interaktívabb, a képi és mozgóképi információkat előtérbe helyező tartalmakat fejleszt. Másik ága azonban, sokkal inkább a tartalomra és az adminisztrációra, ezek rendszerszintű áttekinthetőségét helyezi előtérbe.

Az info-kommunikációs technológia (ICT) támogatta oktatás új paradigmája a hatékonyság, azaz: kit, mennyi idő alatt, mennyiért, lehet adott ismerethez juttatni, mely azonnal értékesíthető a munkaerőpiacon?

Az elkövetkező évtizedekben az iskolák világára a globalizációs tendenciákból elsősorban a kulturális globalizáció fog döntő hatást gyakorolni. Mindez azonban a nemzeti identitásokra, hagyományokra felépülő kultúrákat, ezen belül a műveltség összetevőit fogják majd érzékenyen érinteni, s ezért már a tananyagtervezés szintjén sem lehet lesz majd ettől eltekinteni.

A kulturális globalizációt több tényező is befolyásolni látszik, ilyenek a transznacionális médiatársaságok, a tartalomszolgáltatók, a telekommunikációs nagyvállalatok és a szoftverfejlesztő óriáscégek.

Mikor mondható el, hogy kulturális globalizációval állunk szembe, és melyek e jelenségnek az ismérvei?

A média kutatók véleménye szerint a globalizáció akkor következik be, ha az interaktív információs és kommunikációs rendszerek globálisan teret nyernek, ha a helyi nyilvánosság és információs közösségek felértékelődnek, ha a nemzetek feletti információs monopóliumok kialakulnak, és ha megjelenik a globálisan manipuláló tömegkommunikáció.

Melyek a globális kultúra legfontosabb vonásai?

Egyik legfontosabb jellemzője a területenkívüliség, mely lehetőséget ad a széles körben ható kulturális minták és értékek megteremtésére és terjesztésére. A globális kultúra azonban nem fogható fel a nemzeti kulturák összességként, sőt háttérbe szorítja a nemzeti kulturális mintákat és értékeket.

Ezek után nyilvánvalóvá vált, hogy a jelenlegi oktatási rendszerek reformjára van szükség.

Az e-learning mint eszköz, a hagyományos oktatási modellek megkérdőjelezésekor jelent meg. Az e-learning (elektronikus tanulás) pontos meghatározása, hogy valójában mit is jelent még vita tárgya?

A Google – keresőmotor segítségével az e-learning fogalommal kapcsolatos kérdésköröknek, az elmúlt hónapokban az alábbi találatszámokat regisztráltuk:

e-learning meghatározása	27.600
e-learning-el foglalkozó cikkek	445.000
e-learning fogalom a WEB-en	1,8 x 10 <sup>6</sup>
e-learning tartalomszolgáltatás	1,9 x 10 <sup>6</sup>
e-learning WEB oldalak	668.000

Az e-learning-et mint eszközt definiálva, fizikai értelemben egy olyan **csatorna**-ként képzelhető el, mely úgy is felfogható, mint a kommunikációs stratégiák összessége.

Ez a gyakorlatban, egy tartalomban gazdag, teljesítmény orientált, interaktív, multimédiára épülő informális hálózatot képez, mely az oktatási folyamatban a résztvevő számára önálló haladási tempót biztosít, rugalmas időbeosztásban, azaz bárhol, bármikor igénybe vehetően. Elsődleges célja a motiváció és az élményszerűség megvalósítása úgy hogy az egyén számára a tanulási folyamatban lehetővé tegye a feszélyező helyzetek kiküszöbölését.

Egy a közelmúltban elvégzett felmérés szerint a gyakori internetezők 75 %-ának az a véleménye, hogy alig, 15%-ának pedig az, hogy egyáltalán nincs lehetőség az Interneten keresztül tanfolyamot elvégezni.

A csatorna csak akkor tudja optimálisan feladatát elvégezni, ha megalapozottak stratégiai pillérei, melyeket a teljesség igénye nélkül megpróbálunk felsorolni.

Ezek szervezeti- (adminisztratív elemek), vezérlési- (a tanulási környezet menedzselése), technológiai- (maga az infrastruktúra), kezelőfelületi- (a felhasználói interfész), pedagógiai- (a képzési célok, valamint a képzési program tervezése és kialakítása) értékelési- (a hallgatók és az oktatási tartalom) illetve etikai pillérek (személyeségi és kisebbségi jogok).



Az oktatási-tanulási folyamatban a tananyaggal szemben kialakított „személyes viszony” egyfajta motiváltságot is kell, hogy kiváltson.

A távoktatással kapcsolatban azonban fenntartások is megfogalmazódnak.

A technológia – szabványok – tartalomfejlesztés hármasszögének szorításában leggyakrabban megjelenő **e-learning problémák** is megfogalmazódnak, melyek az alábbiak lehetnek:

- az e-learning rendszerelemek egymással történő együttműködési hiányosságai,
- a nagyfokú hallgatói lemorzsolódási arányszám,
- helytelen ismeretátadási módszerek,
- behatárolt tanulási élményszerűség,
- megváltoztathatatlan tanulási technikák,
- a gyorsan elavuló technikai háttér.

A **Gábor Dénes Főiskola** keretében kifejlesztett e-learning tananyagoknak, egy meghatározott belső szabványrendszer által támasztott követelményhalmaznak kell megfeleljenek, mely megköveteli a:

1. szemléletességet,
  - tartalmában jól tagolt,
  - megjelenítésében multimédiás tananyagot (azaz kép-, hang-, video-, grafikus-, szöveges állományok halmazát).
2. interaktivitást, mely feltételezi, hogy:
  - a tananyag változatos feladatbázissal rendelkezzen,
  - a feladatok megoldási ellenőrzésének a visszajelzése és
  - a feladatoknak, az eredeti feltételi környezetben történő megoldásának lehetősége biztosított.
3. személyre szabottságot, azaz
  - hatékony tutor - tanuló kommunikációt,
  - az oktató által időnként ellenőrizhető tanulási folyamatban a hallgató által bármikor igénybe vehető mail-, chat-, fórumszolgáltatás biztosítását.
4. technikailag állítható WEB-es szolgáltatást (a sávszélesség vagy a felhasználó gépi kapacitásának függvényében).
5. strukturáltságot, azaz
  - a tartalom legyen jól strukturált, értelmezhető alfejezetekre történő felbontásban és lényeg-kiemelési módzatok megadási lehetőségekkel kapcsolatos információkat is tartalmazzon,
  - a tartalom önálló moduláris tananyagegységekre történő bontása a rugalmas tanulást segítse,



- a modulok szintjének egymásra utaltságának a megjelölése, a különböző belépési szintek meghatározása,
- az egyes modulok időigényének konkrét tanulási órákban történő megjelölése,
- a tanulási időtartam egyenletes ütemezése.

#### 6. szemantikus reprezentációt:

- leképező: tudni, hogy mit?,
- operatív: tudni, hogy hogyan?

Ezáltal meghatározhatóvá válnak az e-learning rendszer feladatai:

- statikus tartalom-kezelés: tananyagok, jegyzetek, kiegészítő anyagok létrehozása, tárolása
- dinamikus tartalomkezelés: oktató-diák és diák-diák kommunikáció, kérdések, válaszok megadása; a tananyag közös munkával történő bővítése és fejlesztése; a feladatok kiadása, megoldása, értékelése és megbeszélése; tesztek és vizsgák,
- statikus- és dinamikus tartalom megjelenítés,
- adminisztráció, a hozzáférés szabályozása.

Ugyanakkor megfogalmazódnak a moduláris tananyagokkal szemben elvárt követelmények, is mint a forgatókönyv léte, a jól meghatározott tematikus háttér biztosítása és a célravezető ismeretek felsorolása. Egy valós probléma megoldásának a kitűzése, élményszerű animációs segédletek biztosítása, mely által a tananyag innovatív gondolkodásra serkentővé válik és lehetőséget ad a gyakorlati tapasztalat kialakítására.

Fontos elem a tartalom és a technikai háttér szinkronizációja.

Új tutorálási modellek kialakítására is szükség nyílik, melyek új eszközöket vonultathatna fel, pl. az oktatás egész folyamán elérhető virtuális előadót.

A tananyagírás megkövetel egy komoly didaktikai tervezést, illetve ismeretanyag feldolgozást. A tapasztalat azt mutatja, hogy míg egy hagyományos egyetemi jegyzet 3-5 hónap alatt készülhet el, addig egy e-learning tananyag, ismert technikai háttér és magas színvonalú szakértelem esetén is akár 1-2 év.

A tananyagtervezés folyamata megkövetel bizonyos feladatok kitűzését, mint: a szakanyag/forrásanyag elkészítését, a tananyag elektronikus tanulásra való előkészítését, a megfelelő média-elemek (kép, hang, grafika, animáció) kiválasztását és nem utolsósorban értékelési elemek beépítését.

Egy jó tananyag elkészítésének feltétele bizonyos kompetenciáknak, képességeknek egyidejű birtoklását is feltételezi, mint: a szakmai-, didaktikai-, motivációs-, információ rendszerezési képességet és a vizuális kommunikáció képességet ill. számítástechnikai ismereteket.

# Hogyan viselkedjünk az Interneten!

## Good Behaviour on the Internet

Kása Zoltán

Babeş-Bolyai Tudományegyetem, Matematika és Informatika Kar

### Abstract

*We deal with the following topics:*

- *netiquette, the rules of behaviour when writing emails*
- *how can we protect our computer against viruses spreading by Internet*
- *how must we realize our homepages to avoid current errors.*

Előadásunkban a következő három fő témakörrel foglalkozunk:

- – Netikett – viselkedési szabályok
- – Védekezés a vírusok ellen
- A honlaptervezés hibái

### Netikett

A hálózaton való viselkedés szabályait először Arlene H. Rinaldi foglalta rendszerbe, minden későbbi változat ezen alapszik. A kultúrált viselkedéshez legalábbis számítógépi és hálózati alapismeretek szükségesek. A legnagyobb veszély, hogy az Internet személytelen, ezért az emberek hajlamosak olyan megnyilvánulásokra, amelyekre a mindennapokban sohasem vetemednének. Alapelveként kell elfogadni, hogy minden megnyilvánulást alaposan meg kell gondolni, beleértve a legegyszerűbb levél elküldését is. Ezzel sok kellemetlenségtől menthetjük meg magunkat. Érdemes figyelembe venni a következőket:

- a levél formája és tartalma,
- a listára küldött levél más, mint a magánszemélynek küldött,
- a címzés ellenőrzése,
- csatolt állományok minimálisra csökkentése (listára egyáltalán ne küldjünk),
- válaszevél írása, milyen hamar illik válaszolni stb.

Alapelv: legyünk liberálisak a levelek fogadásakor és konzervatívak a küldéskor.

## Védekezés a vírusok ellen

Régen azt tartottuk, hogy a vírusok elleni védekezés legjobb formája, ha csak törvényes programokat használunk, és nem másolunk innen-onnan. Ez ma is igaz, csak kevés. Az Internet nem eléggé körültekintő használata észrevétlenül is megfertőzheti gépünket. Mivel a legtöbb vírus a levelező programot vagy a böngészőt fertőzi meg, vagy annak segítségével terjed, jó ha nem a legközismertebb programokat használjuk levelezésre és böngészésre. Szerencsére terjedőben van, főleg egyetemi körökben, a Linux operációs rendszer, amely amellet, hogy biztonságosabb, igen nagy előnye, hogy ingyen kapható és terjeszthető, akárcsak a különféle Linuxos programok.

## A honlaptervezés hibái

Mivel a honlapokat több száz vagy ezer ember is látogathatja, akár naponta is, nem mindegy, hogy ezek milyenek. A honlaptervezésnek is vannak szabályai, akárcsak a városépítésnek, mert nem mindegy, hogy milyen térben, legyen az bár virtuális is, töltjük időnk egy részét. Természetesen, ebben az esetben nem lehet tiltó szabályokat felállítani, csupán ajánlatokat tehetünk. De ezeket jó figyelembe venni, különben ellenkező hatást válthatunk ki, és honlapunkat egyre kevesebben nézik meg.

Jakob Nielsen már évek óta figyel az Internetet, és évenként megfogalmazza kifogásait, ezek elérhetők különböző internetes helyeken, érdemes őket elolvasni, a bennük lévő kifogásokat megszívlelni.

## Könyvészet

- [1.] Magyar netikett, <http://www.isys.hu>
- [2.] Jakob Nielsen honlapja, <http://www.useit.com/alertbox/>



# Információkeresési stratégiák a weben és tanításuk tapasztalatai

## Web Search Strategies and Experiences of their Teaching

K. Princz Mária

Debreceni Egyetem Műszaki Főiskolai Kar

### Abstract

*There is a lot of information on the Web, but can we find the necessary one every time? What kind of strategies can we followed? What is worth paying attention to?*

*There are some strategies that we can use: guessing the URL, subject directories, search engines, meta-search engines, web-based databases.*

*Search engines are important tools for locating and publicizing information on the Web but most of search engines deals with only the surface of information of the web. Knowledge of their features helps us to search efficiently and build search-friendly HTML pages.*

*The invisible web is a part of the web what search engines cannot find.*

### 1. Bevezetés

Az Internet használata egyre inkább tért hódít a mindennapi életben, s különösen igaz ez az oktatás területén. Hatalmas mennyiségű információ érhető el a weben keresztül, de vajon megtaláljuk-e mindig a számunkra éppen szükségeset?

A Debreceni Egyetem Műszaki Főiskolai Karán az első éves hallgatók körében vizsgáltuk, mennyire képesek tájékozódni a weben, hogyan, milyen hatékonysággal keresnek. Tapasztalataink megegyeztek a CyberAtlas honlapján közzétett iProspect tanulmánnyal:

- Az Internetet használók mintegy  $\frac{3}{4}$ -e használ kereső szoftvert.
- A felhasználók 52 %-a ugyanazt a keresőszoftvert vagy tematikus keresőt használja az információk keresése esetén és csak 35% használ alternatív kereső eszközt. 13% esetében igaz, hogy különböző típusú kereséseknél különböző keresőszoftvert használ.
- A keresőszoftvert használók 16%-a csak az első egy-két eredményt nézi meg, 32% végignézi az első eredményoldalt, 23 % eljut a második eredményoldalig, 10,3 % a harmadikig, 8,7% még a harmadik

után néz, és 10 % végignézi az összes eredményt, ha az nem oldalak tucatja.

- Az első keresés sikertelensége esetén a felhasználók csupán 7,5%-a finomítja tovább a kereső kérdést, 27,2% pedig más keresőszoftverrel próbálkozik.

A hiányokat érzékelve a webes keresés tanítását felvettük az általános gyakorlati tananyagba, s erről a hallgatói visszajelzések pozitívak.

## 2. A web megoszlása

### 2.1. A látható web (*Surface Web* vagy *Visible Web*)

A web része, amely a weben át elérhető, az általános keresőszoftverek által begyűjtött és indexelt dokumentumokat tartalmazza.

### 2.2. A láthatatlan web (*Invisible Web*)

A webnek azon része, amely a keresőszoftverek számára láthatatlan. Ezen dokumentumokat a keresőszoftverek nem tudják (technikai korlátok) vagy nem akarják (elvi döntések) adatbázisukba beemelni.

Különböző becslések találhatók a weben a láthatatlan web nagyságáról. A legtöbb becslés a látható webnek 400-550-szeresére értékeli, de van, aki ettől is nagyobbra tartja.

A láthatatlan web részei:

#### *Nem átlátható web (Opaque Web)*

Azon dokumentumok tartoznak ide, amelyek könnyen elérhetőek lennének a begyűjtő robotprogramok számára, de valamely okból mégis kimaradnak a keresőszoftverek adatbázisaiból.

Ennek oka lehet a begyűjtés mélységének korlátja, az egy domainről begyűjthető dokumentumok számának korlátja, a frissítési periódus alatt megváltozott, megjelent oldalak, a dokumentumszigetek.

#### *Magán web (Private Web)*

Az Interneten át elérhető, de különböző megoldásokkal a nyilvánosan látható oldalak közül kizárt anyagok, az intranet hálózatokon található dokumentumok tartoznak ide.

A kizárás történhet a robot.txt fájl alkalmazásával vagy a noindex meta tag használatával, de a tűzfalal, jelszóval való védelem is lehetséges.



### *Szabadalmazott web (Proprietary Web)*

Olyan adatbázisok, dokumentumok tartoznak ide, amelyeket tartalom-szolgáltatók állítanak elő, és ezért regisztráció után előfizetéssel vagy külön díjért tekinthetők csak meg

### *Mély web (Deep Web)*

A láthatatlan web legnagyobb részét teszi ki. A mély web a weben át szabadon elérhető adatbázisokat tartalmaz (pl. menetrendek, sárga oldalak, stb.). Ezen adatbázisok tartalma technikai korlát miatt láthatatlan a begyűjtő robotok számára, hiszen a lekérdezés megadására (begépelés, kiválasztás) e programok képtelenek.

### *Az igazán láthatatlan web (Truly Invisible Web)*

A nem HTML formátumú dokumentumokat (audio, video, képek) nehezen vagy sehogy sem értelmezik a keresőszoftverek, ezért nem is építik be az adatbázisukba. Számos egyéb formátumra (pdf, flash, office fájlok, stb.) is igaz, hogy időigényes indexelni őket, ezért számos keresőszoftver nem foglalkozik ezen típusú dokumentumokkal.

Elvi okból kerülnek kizárásra a dinamikusan generált web oldalak, valamint a valós idejű tartalmak.

## **3. A webes keresés eszközei**

A weben lévő információ keresésekor néhány jól bevált stratégiát követhetünk: a jónak vélt URL cím beírása, keresőszoftverek (search engines), témakatalógusok (subject directories), metakeresők (meta-search engines), webes adatbázisok alkalmazása.

### *3.1. Keresőszoftverek*

A weben lévő dokumentumok keresésében a keresőszoftverek szerepe elsődleges. A keresőszoftverek tulajdonságainak ismerete segít a lekérdezések minél hatékonyabb megfogalmazásánál, de hasznos a jól kereshető web oldalak írásánál is.

A keresőszoftverek három elkülönülő részből épülnek fel:

#### *Begyűjtő rész*

Minden keresőszoftvernek megvan a saját robotprogramja, amely a begyűjtést végzi. A különböző robotok a weben lévő dokumentumok csak egy részét indexelik. Különböznek abban, hogy mely szervereket tekintenek kiindulási pontnak, egy adott domainről hány dokumentumot gyűjtenek be, milyen frissítési periódust használnak.

### *Indexelő rész*

Az indexelő részben a begyűjtött dokumentumok különböző elemeiből (pl. a dokumentum neve <TITLE>, fejléc információk, címsorok, horgony elemek, kiemelt szövegrészek) a keresőszoftver saját adatbázist épít fel vagy tovább bővíti azt.

### *Lekérdező rész*

A lekérdező rész további három részből áll:

- a lekérdezési interfész (egyszerű, összetett, részletes keresés lehetősége)
- a lekérdezést megvalósító egység (a keresőszoftver adatbázisából veszi elő a dokumentumokat),
- az eredményeket rangsoroló rész.

Valamennyi kereső úgy rendezi a keresés eredményét, hogy az eredménylista elejére az általa legfontosabbnak tartott dokumentumok kerüljenek. A rangsorolási algoritmusok keresőnként különböznek (szövegek statisztikai analízise, hivatkozások analízise, klaszterezés).

A web át elérhető információk döntő hányadánál a keresőszoftverek nem segítenek minket az információk megtalálásában.

A különböző keresőszoftvereken feltett kérdések találati listája nagyon különböző, és kevés az átfedés közöttük. Ez azt jelenti, hogy célszerű minél több keresőt használni, ha valamely témában alaposan át szeretnénk nézni a weben tárolt dokumentumok tömegét.

### *3.2. Metakeresők*

A metakeresők használatával számos weben keresztül elérhető adatbázist kérdezhetünk le egy időben, egységes lekérdező interfészen keresztül, így a web nagyobb részéből kapunk találatokat.

### *3.3. Témakatalógusok*

A témakatalógusok az Interneten található dokumentumokra mutató hiperhivatkozások sokszor hierarchikus gyűjteménye, ahol tartalom szerint felépülő könyvtárakban kereshetünk. A témakatalógusok a webnek szűkebb részét fedik le, mint amit a keresőszoftverek adatbázisai tartalmaznak.

### *3.4. Webes adatbázisok*

Számos hasznos információ különböző adatbázisokból nyerhető. (pl. telefonszámok, menetrend, stb.)

A láthatatlan web részét képező források, adatbázisok elérését segítik a rendszerezett hivatkozások gyűjteményei, a témakatalógusok (pl.

www.invisible-web.net, de egy-egy adatbázis kezdőlapjának megtalálásához a keresőszoftverek is jól használhatók. Ekkor a keresés megfogalmazásakor célszerű a kulcsszavak megadása mellett a lekérdezést bővíteni az *adatbázisok OR archívumok OR gyűjtemények* megadásával, majd a kapott nyitóoldalon keresni az adatbázis saját lekérdező eszközeivel.

#### 4. A webes keresés tanításának tapasztalatai

Hatékonyabban, erőteljesebben kereshetünk, ha ismerjük a rendelkezésre álló lehetőségeket, éppen ezért fontos a webes információkeresést tanítani, növelni a felhasználók tudatosságát a keresőeszközök használatakor.

A tapasztalat azt mutatja, hogy a hallgatók jelentős része nem ismeri a lehetőségeket egy-egy információ keresésekor. A többség keresőszoftvert használ minden esetben, de legfeljebb egy-két keresőszoftver URL címét ismeri. Nem ismerik a különbségeket e szoftverek működésénél (pl. milyen típusú információt gyűjtenek), így választani sem tudnak, mikor melyiket érdemes használni.

A sikeres kereséshez a megfelelő keresőszoftver kiválasztásán túl fontos annak tulajdonságainak ismerete is: nem mindegy, hogyan fogalmazzuk meg a lekérdezést, hiszen az az eredményhalmazt döntően befolyásolja.

Érdemes tudatosítani, hogy az összetett lekérdezések könnyítésére egyre több keresőszoftver lehetővé teszi a részletes keresést is, így nem kell megjegyeznünk a pontos beírási szabályokat, de bonyolultabb összetett kereséseket nem lehet megfogalmazni a részletes keresés oldalon.

Számos portál lehetővé teszi nyitólapján a weben való keresést, de csak egy mezőt kínál fel a keresés megfogalmazására, ami nehezíti a lekérdezés megfogalmazását.

#### Általános hibák:

- Sokszor előforduló hiba, hogy a hallgatók magyar nyelvű szövegek keresésekor begépeléskor nem használják az ékezetes betűket (pl. *epiteszet építészet* helyett).

Ez esetben számos jelentős honlap kimarad az eredménylistából, s csak azok az oldalak jelennek meg, amelyeknél az URL címben szerepel a keresési kifejezés (pl. *epiteszet.html*), esetleg a szöveg létrehozója meta adatként keresési kulcsként hibásan is szerepelteti (gondolva az ékezet nélküli begépelésekre), vagy a dokumentum szövegében is helytelenül írva jelenik meg a keresett szó.



- Szintén kizárunk releváns oldalakat, hibásan szűkítjük a találati listát rossz helyesírással megadott keresési kulcsszóval. (Pl. *szinész* – 1200 találat, *szinész* – 37600 találat a Google esetében.)
- Sok esetben túl messziről közelítenek a hallgatók egy-egy keresésnél. Ha a gótikus építészet stílusjegyeire vagyunk kíváncsiak, akkor kevés az *építészet* keresési kulcs megadása.
- Érdeemes tudatosítani, hogy keresőnként különböző eredményeket kapunk, ha a szavakat egyes vagy többes számban szerepeltetjük. A \* helyettesítő karaktert sem ismeri minden kereső (pl. Google).
- Sokan nem tesznek különbséget szavak és kifejezések keresése között. Főlöleslegesen bővítjük az eredményhalmazt, ha kifejezéseket keresési kulcsszavanként adunk meg (osztott rendszerek "osztott rendszerek" helyett). Szókapcsolatok keresése esetén érdemes az idézőjelet kitenni vagy a részletes lekérdezés ablakánál a megfelelő mezőbe írni a szavakat, mert így az eredménylistát lényegesen szűkíthetjük.
- Megszorítások adhatók a dokumentum előfordulási helyére domainenként, site-onként. Szűkíthetjük az eredménylistát, ha megadjuk a dokumentum nyelvét, típusát, méretét, a dokumentumok létrehozására vonatkozó időkorlátot, stb. Természetesen keresőszoftverenként változik, hogy milyen szűkítést enged meg, s milyen módon fogadja el a megadást.
- Legyünk körültekintők az összetett keresések megfogalmazásánál! Ha egy keresőszoftver nem ismer egy logikai operátort, ami a lekérdezésben szerepel, akkor azt a megadott keresési kulcsszavak mellett egy újabb kulcszónak veszi, ezáltal alapértelmezésének megfelelően hibásan tovább szűkíti (pl. AllTheWeb) vagy főlöleslegesen bővíti az eredményhalmazt.

A keresések megfogalmazásától lényegesen függ a találatok száma. Az alábbi táblázat erre mutat néhány példát:

Kereső kifejezés	AltaVista	AllTheWeb	Google	Heuréka	Vizsla
osztott rendszerek	687	1571	4980	1189	114545
osztott rendszer	1373	3321	4110	1775	109047
"osztott rendszerek"	66	93	142	-	142
"osztott rendszer"	31	49	75	-	120
osztott AND rendszerek	687	428	4980	1189	115923
osztott OR rendszerek	28493	140	69100	46022	2332276
tanszék	14356	95526	78100	19455	112993
tanszék site:kite.hu	-	924	904	615	1145
tanszék host:delfin.kite.hu	2	141	145	39	193

*A lekérdezések eredményei<sup>1</sup>*

<sup>1</sup> "-" tartalmú cella: a keresőgép nem támogatja az adott műveletet



Mára a legtöbb keresőszoftver nyitólapjáról biztosítja valamely válogatott, rendszerezett hivatkozásgyűjtemény elérését, illetve a tematikus keresőként induló szolgáltatók is lehetővé teszik nyitólapjukon a weben lévő keresést. A tanórák alatt lényeges tudatosítani a hallgatókban, hogy mikor melyik eszközt célszerű használniuk, s érdemes rávilágítani a gyakori hibákra.

### Irodalomjegyzék

- [1.] *Search Engine Watch* <http://searchenginewatch.com>
- [2.] *WebReference* <http://www.webreference.com>
- [3.] *Search Engine Showdown* <http://www.searchengineshowdown.com>
- [4.] *S.Brin, L.Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine WWW7 / Computer Networks 30(1-7), pp. 107-117, 1998*  
<http://www.stanford.edu>
- [5.] *C. Sherman, G. Price: The Invisible Web: Uncovering Information Sources Search Engines Can't See. CyberAge Books, 2001*
- [6.] *CyberAtlas*  
[http://cyberatlas.internet.com/markets/advertising/article/0,,5941\\_1500821,00.html](http://cyberatlas.internet.com/markets/advertising/article/0,,5941_1500821,00.html)
- [7.] J. Nielsen, Search: Visible and Simple Alertbox, May 13, 2001  
<http://useit.com>
- [8.] *K. Princz Mária: Systems to access information in the Web MicroCAD'2000 International Computer Science Conference, Miskolc*
- [9.] *K. Princz Mária – Rutkovszky Edéné: Content Discovery of Invisible Web 6<sup>th</sup> International Conference on Applied Informatics, Eger 2004*
- [10.] *K. Princz Mária: A webes keresők tanításának tapasztalatai E-learning alkalmazások a hazai felsőoktatásban, Budapest, 2003*
- [11.] *K. Princz Mária: A weben lévő információk hozzáférhetősége NetworkShop, Pécs 2003*
- [12.] *K. Princz Mária – Rutkovszky Edéné: Ismeret reprezentáció a weben NetworkShop, Eger 2002*

# A szöveg- és kiadványszerkesztés oktatásának néhány tapasztalata

## Experiences on Teaching Word Processing and Some Parts of Typography

Bujdosó Gyöngyi

Debreceni Egyetem

Matematikai és Informatikai Intézet

Komputergrafikai és Könyvtárinformatikai Tanszék

### Abstract

*At universities and secondary schools, there are more and more courses on how to make web pages, at which the learners get a big amount of useful information on technical issues in detail that helps them in constructing complex, multifunctional web pages. Is it enough to make well-usable web pages? Do the teachers emphasize that the easy-of-use needs the data be in well-arranged forms and have an appropriate layout? Can or should the web page makers use the typographical rules and traditions in DTP? Are the typographical rules dispensable?*

*This paper deals with a method of teaching how to form web pages or other types of documents (e.g. dissertations, CV's, invitation cards). The author demonstrates the main issues, the successes and the resultlessnesses of the course.*

*Kivonat.* Mind a felső-, mind a középfokú képzésben egyre elterjedtebb és egyre népszerűbb a weboldalak készítésének intézményes oktatása. A kurzusok során számos hasznos technikai fogásról kaphatunk részletes tájékoztatót. Fektetünk-e azonban hangsúlyt arra, hogy nemcsak a tartalom, hanem a forma és az áttekinthetőség is nélkülözhetetlen kelléke egy jó weboldalnak? Segítjük-e a hallgatókat abban, hogy álláskeresésnél jó tartalmú és formájú önéletrajzot készítsenek? Felkészítjük-e a jövő tanárait arra, hogy esztétikus külsőt adjanak egy általuk készített oktatási segédanyagoknak?

Az általános vélemény sajnos az, hogy ezek lényegtelen dolgok, a külalak nem, csak a tartalom lényeges. Ez pedig – mint azt a gyakorlatban jól tudjuk – valójában nem így van, a külalak sok esetben nagyon is lényeges.

E cikkben a szöveg- és kiadványszerkesztési ismeretek hiányának néhány negatív következményén túl bemutatásra kerül egy viszonylag jól működő módszer, amellyel fel lehet kelteni a nyomtatott és elektronikus oldalak készítését tanulók igényét az esztétikus külső kialakítása iránt.

## Mit tanítunk a DTP keretében?

Az, hogy mi a DTP, viszonylag jól körülhatárolt foglom. Mit tanítunk azonban a DTP-t oktató kurzusokon? Tapasztalatom szerint a DTP oktatásának keretében rendszerint a programok használati módját, lehetőségeit ismerhetjük meg a tanulókkal. Vajon nincs szükségünk egyébre? Elegendő ennyi ahhoz, hogy megfelelő nyomtatott vagy webes oldalakat hozzunk létre? A gyakorlatból láthatjuk, hogy nem, hiszen nem kellemes például „összedobott” weboldalakat olvasni, és sokszor elszomorító látvány egy-egy dolgozat külalakja is.

## Tanítani kell(ene) a kiadványszerkesztést

Ezzel az állítással bizonyára sokan nem értenek egyet, hiszen mindenki tud kezelni legalább egy szövegszerkesztőt, és mindenki tud készíteni akár ötszáz oldalas könyvet is. A sok éves tapasztalat mégis azt mutatja, hogy az MS Wordöt naponta használó hallgatók sem ismerik magát a programot sem. Alig akad közöttük olyan, aki tudja, hogy megkönnyíthetné munkáját például a fejezetek automatikus számozásával vagy a tartalomjegyzék automatikus generálásával. A szöveg formázásának kultúrája pedig szinte meg sem jelenik: százból kettő állítja át az alapértelmezetten megadott Times New Roman, illetve Arial betűtípust, és ennél sokkal rosszabb arányt kapunk, ha a bekezdésszínekről vagy a sablonokról kérdezzük őket.

Kicsit jobb a helyzet a weboldalak esetében, ugyanis ezeken általában megjelenik a technikai tudás. Ez részben annak köszönhető, hogy a DTP e területe hivatalosan az informatika részét képezi, így a weboldalkészítés technikáját lehetőségünk van kurzusokon elsajátíttatni. Számos olyan weboldallal találkozhatunk azonban, amelyeken az információ rejtve marad a túlszűfoltóság, a letölthetetlen mennyiségű beágyazott programocska, a sok pulzáló, forgó, menetelő „díszítő” részlet, a rossz színösszeállítás vagy elhelyezés miatt. Ezekről már kevés szó esik a kurzusokon.

A a kiadványszerkesztés oktatásával segíthetünk abban, hogy a különböző nyomtatott és webes dokumentumokat szerkesztők a technikai ismeretek mellett a forma kialakításának különböző eszközeivel is fel legyenek vértvezve, és legalább ilyen fontos, hogy kialakuljon bennük az esztétikus küllemű munka létrehozása iránti igény.

## Tanítjuk-e legalább a szövegszerkesztést?

A fenti tények ellenére hazánkban alig van olyan felsőoktatási intézmény és szak, ahol kötelező tantárgy lenne akárcsak a szövegszerkesztés. Ebből arra lehet következtetni, hogy a tanárok sem tartják lényegesnek ezt a



fajta tudást. Valóban lényegtelen? Nem tartjuk már annak, amikor kezünkbe veszünk egy-egy dolgozatot, diplomamunkát vagy magánkiadásban megjelent könyvet, és nézzük a szétszórt sorokat, az áttekinthetetlen, kusza szerkesztést, a minden esztétikát nélkülöző oldalkialakítást.

Ha létezik szövegszerkesztést oktató kurzus, mi annak a tematikája? Ha megtanítjuk, hogyan hozhat létre automatikusan számozott címsorokat, megmutatjuk-e, hogyan érdemes átalakítani azok alapértelmezett megjelenését? Ha megnézzük, melyik menüpont alatti párbeszédablakban lehet a margókat átállítani, ejtünk-e szót arról, hogy milyen értékeket érdemes használni? Elegendő-e megmutatni, hogyan lehet a kurzív, félkövér, kiskapitális betűstílusokat beállítani, vagy hívjuk-e fel mellette a figyelmet arra, hogy ezeket hol, mire érdemes használni? Ha a szóaláhúzásról esik szó, elmondjuk-e hogy lehetőség szerint kerülendő, hangsúlyozzuk-e hogy címsoroknál sem elfogadható? Beszélünk-e a sok helyen látható, kiemelésre vagy sorkizárásra használt betűritkításról?

Sajnos a legtöbb esetben a válasz: nem. Megelégszünk a technikai tudással, ami a tanuló munkáját leegyszerűsíti ugyan, a végeredmény esztétikájára azonban javító hatással nincsen. Sok esetben még romboló erejű is lehet, ha minden tanács nélkül megmutatjuk például, hogy hogyan lehet a betűket ritkítani vagy szövegrésztetet aláhúzni.

Miért ne adhatnánk meg az alapokat az iskolai évek alatt? Hiszen mindenki fog később is szöveget szerkeszteni. Ez egy olyan tevékenység, ami szaktól és foglalkozástól függetlenül mindennapi életünkben egyre inkább jelen van.

## Akadályok

A szöveg- és kiadványszerkesztés minden szakon történő kötelező tantárgyként való oktatásának jelenleg két alapvető akadálya létezik.

A legerősebb gátló tényező az az általánosnak mondható nézet, mely szerint mindenki megtanulhatja autodidakta módon fél óra alatt. Ennek a véleménynek az alábbi közvetlen következményei vannak:

- Nem fektetünk energiát a hallgatók esztétikai igényének kialakítására.
- Azt sugalljuk, hogy a forma lényegtelen, csak a tartalom a fontos.
- Arra utalunk, hogy ez a terület nem tartozik az alaptudáshoz, ez egy olyan szakma, amihez semmi közünk, számunkra lényegtelen.

Ha segíteni akarjuk a hallgatókat későbbi munkájuk során, meg kell változtatnunk ezt a hozzáállást, mert

- elvárják egy számítógépet használni tudó fiataaltól – főleg ha az felsőfokú végzettséggel, különösen akkor, ha informatika szakkal rendelkezik –, hogy esztétikus külsejű munkát adjon ki a kezéből, legyen az például önéletrajz, beszámoló, pályázat vagy hivatalos levél;



- sokszor a forma legalább olyan lényeges, mint a tartalom – gondoljunk csak a weboldalakra, egy beadott kérvényre, önéletrajzra vagy egy elkészített oktatási segédanyagra;
- az informatikának ez egy olyan határterülete, mellyel mindenki, szinte naponta kapcsolatba kerül.

A másik gátló tényező az, hogy nem állnak rendelkezésünkre e témában járatos szakemberek. Az ideális az lenne, ha nyomdaipari szakemberek, tipográfusok tartanák e kurzusoknak legalább egy részét. Sajnos azonban az anyagi vonzatok e megoldást rendszerint nem teszik lehetővé. Miért ne szervezhetnénk azonban térítéses, részben támogatott oktatást e témáról? Bizonyára sokan áldoznának rá. Egy másik megoldás az lehetne, ha támogatnánk a téma iránt érdeklődő oktatóink ilyen irányú továbbtanulását, képzését. Ehhez viszont el kellene fogadnunk azt, hogy ez az informatikának egy létező határterülete, amire érdemes és szükséges is áldozni.

### **Egy létező kurzus tervrajza**

Ebben a fejezetben egy olyan megoldást ismertetünk, amely visszaigazoltan hatásos módon foglalkozik a DTP technikai és esztétikai vonatkozásaival is.

#### *Eszköz*

A DTP oktatására két különböző programrendszert alkalmazunk. Az informatikában kevésbé járatosak az MS Office programjait használják, melyekhez illeszkedően alakul a tananyag is, szót ejtve az adott eszközzel automatikusan vagy egyáltalán nem megoldható dolgokról is. A programozásban jártasabbak, illetve a lelkes, akár bölcsész szakos hallgatók a nehezebben kezelhető, ám nagyobb eszköztárral rendelkező TeX kiadványszerkesztő programcsomag használatán keresztül kapnak bevezetést a kiadványszerkesztés alapjaiba.

#### *Élő minták*

A lehető legtöbb pozitív és negatív élő minta bemutatása nagy hatással van a hallgatókra. Itt bemutatásra kerülnek könyvek mellett prospektusok, meghívók, üdvözlő- és névjegykártyák, valamint weboldalak is.

#### *Reprodukálendő mintalapok*

Minden óra anyagához mintalapot, mintaoldalt kapnak a hallgatók, amely maga is mintául szolgál a szerkesztéshez. Ez azt jelenti, hogy a lapok a szabályoknak megfelelő módon épülnek fel, jó példát mutatva a forma

kialakításához, nem zsúfolják egy oldalra az adott témához tartozó összes technikai megvalósítást. A mintalapokat a hallgatóknak pontosan kell reprodukálniuk, ez megkívánja a témához tartozó technikai tudást. A mintalapok egymásra épülnek – alkalmazva a korábban elsajátított részek anyagát is – ami azon kívül, hogy átismétli az előző anyagokat, elkerüli az egyes mintalapok túlzásfoltta válását.

### *Nyelv: nyelvtan és speciális karakterek*

A nyelvvel kapcsolatosan két területet kell folyamatosan hangsúlyoznunk. Az egyik a nyelvtan, miszerint ragaszkodjunk a magyar helyesírás szabályaihoz, és ne nyelvújításként, hanem hibaként értelmezzük például a -val/-vel rag sajnos egyre elterjedtebb -al/-el formáját. Informatika szakosoknál mindenképpen lényeges a betűszavak, illetve idegen szavak ragozási szabályainak „felelevenítése” is. A másik terület a nyelv jellegzetesen használandó karaktereinek köre, mint például a szóköz (hova tegyünk és hova ne!), a központoszás, idézőjelek és zárójelek, a kötőjel, a hosszú kötőjel és gondolatjel, valamint a kivonás jelének megkülönböztetése – alakját és használati módját illetően egyaránt. A mintalapok összeállításánál fontos szempont, hogy a leggyakrabban előforduló, legcsúnyább hibák helyes megoldása több helyen is látható legyen a szövegekben. A problémás esetek minden egyes előfordulásánál tárgyalásra kerül a vonatkozó nyelvtani szabály is.

### *Kiadványszerkesztés alapjai*

A tananyag úgy áll össze, hogy a program megismerésével párhuzamosan ismerkednek meg a hallgatók a kiadványszerkesztés alapjaival. A programrendszer egy-egy lehetőségének bemutatásakor, mielőtt még elkezdenénk a mintalap reprodukálását, a hallgatók megismerkednek az adott részre vonatkozó alapszabályokkal, szerkesztési javaslatokkal. Az alábbi főbb területeket vesszük végig:

**Betűk** • A formai jellemzők áttekintése után, megnézzük, hogy egy adott feladatra (szakdolgozat, könyv, újság, weboldal) mely típusú és fokozatú betűket érdemes választani, szót ejtünk a betűkeverés veszélyeiről. Nagy hangsúlyt fektetünk a különböző betűfajták (kurzív, félkövér, kiskapitális), illetve a verzál szedés alkalmazási köréről.

**Bekezdések** • A szövegtörzs alapformája (behúzások, sortávolság, fatyúsor, elválasztás, szóköz mérete stb.) után a felsorolások és szövegek közötti kiemelések különböző formáit, majd a lábjegyzetek, irodalomjegyzék lehetséges szédési módjait tekintjük át.

**Ábrák, táblázatok** • Az ezekre vonatkozó formai tanácsokat, elkészítési és beillesztési módokat taglalja a következő rész. Ezen belül megnézzük azt is, hogyan helyezhetjük el és formázhatjuk az ábraalírást, illetve a táblázat címét.



**Címsorok** • A szöveg rendszerezéséül szolgáló címrendszer kialakításával (betűméret, közök, sortörés, szintek) viszonylag sok időt töltünk. Itt ejtünk szót részletesebben a kiemelések fontosságáról, alapkövetelményeiről és megvalósíthatósági lehetőségeiről. A címsorokkal együtt tárgyaljuk a tartalomjegyzék stílusát és elhelyezését is.

**Oldal** • Az oldal kialakításánál foglalkozunk a margók kiszámításával, az élőfej kialakításával, az oldalszám elhelyezésével. Szót ejtünk a sötét háttérrel való munka során felmerülő újabb problémákról, nevezetesen a megfelelő betű és az alkalmazott színek kiválasztásáról.

**Dokumentum** • Ezek után ejtünk szót egy teljes dokumentum, praktikusán egy szakdolgozat kialakításáról, részéről, azok elhelyezéséről és tartalmáról is. E témakör keretében beszélünk a címoldal elkészítéséről. A legalább két féléven keresztül tartó kurzusokon a szöszedet készítését és formázását is áttekintjük.

**Egyéb** • A fentiekben túl főliák/diák szerkesztéséről, hivatalos levelek, magyar és amerikai típusú önéletrajzok részéről, formáiról és tartalmáról tanulunk.

### Szakirodalom

A későbbi tanulást is figyelembe véve érdemes megadni a témához kapcsolódó összes forgalomban lévő, és a legjelentősebb kölcsönözhető művek listáját, illetve a területen legnevesebb szakemberek nevét.

### Sikertelenségek

Számonkérésnél sajnos szinte mindig az derül ki, hogy a hallgatók nem tartják lényegesnek a *megfelelő arányok számszerű tudását*. (A sikerek közé sorolandó azonban, hogy az aránytalanságokat a bemutatott – mások által készített – mintákon már nagy százalékban felfedezik.)

A legsikertelenebb terület a kurzus folyamán a *nyelvtani információk elsajátítása* terén észlelhető. Felsőbb éves hallgatók is gyakran felháborítónak tartják ezen ismeretek „átismétlését”, s főleg azok számonkérését. Annak ellenére, hogy igen nagy gondot fordítok erre a területre, számonkérésnél csak nagyon kis százalékban jelölik be a helyes megoldást, és az is felfogható úgy, mint az egyválasztásos tesztek véletlenül eltalált helyes megoldása.

### Sikerek

A kurzusok folyamán észrevehetően nő a *hallgatók igénye* a szép kivitelű munkák iránt. Egyre több rossz példát vesznek észre a mindennapi életben (napilapokban, könyvekben, weboldalakon). Ez azt mutatja, hogy nemcsak a tartalomra, hanem a formára is figyelnek már. Ez is arra utal, hogy megnő az

igényük az esztétikus formák iránt, ami pozitív hatással van nyomtatott dokumentumaik és webes oldalaik formátumának kialakítására is.

A kurzus folyamán a legintenzívebb jegyzetelés a *szakdolgozat* egyes részei *tartalmának* áttekintésénél látható. Érzékelhetően új információként hallgatják azt, hogy milyen részekből állik állnia egy szakdolgozatnak, hogy hogyan kell összeállítani egy bevezetőt vagy egy összefoglaló részt, hogyan kell elhelyezni az irodalomjegyzékre történő hivatkozásokat, illetve hogy miknek kell feltétlenül szerepelnie a hivatkozásjegyzékben.

Rendkívül nagy érdeklődés övezi az *önéletrajzokban* szükségesen, valamint ajánlottan szerepeltetendő *adatok felsorolását*, valamint a magyar és amerikai típusú önéletrajzok különbségeiről szóló információkat is.

### Összességében...

Még a kreditszerzés miatt jelentkező hallgatók nagy részénél is jelentős haladás tapasztalható – mind a technikai tudás, mind a kiadványszerkesztési ismeretek elsajátítása terén – az egy féléves kurzust végzők esetén is. Tudatosabban és nagyobb odafigyeléssel készítik kisebb terjedelmű munkáikat is.

A visszajelzések azt tanúsítják, hogy azok a hallgatók, akik a kurzusok során nem veszik komolyan az oktatott ismereteket (mert például csak kreditek szerzése miatt, gyenge eredménnyel végezték el a kurzust), az egyetemről kikerülve, a munkahelyen már átnézik és alkalmazzák azokat.

Mindezek azt bizonyítják, hogy valóban szükség van, szükség lenne a kiadványszerkesztéssel kapcsolatos ismeretek átadására.



# Elektronikus jegyzet a párhuzamos programozás oktatásához

An Electronic Lecture-book  
for Studying Concurrent Programming

Dr. Kallós Gábor  
Széchenyi István Egyetem, Győr

## Abstract

*Concurrency plays and will play a very important role in today's and future computer systems. At our University we teach the students the most important techniques and methods in concurrent programming since years, and we have presented also an electronic version of the lecture-book for (partial) self-study. Besides the concurrent programming, we partly cover the topics of concurrent algorithms and concurrent hardware architectures. The programming part is mostly based on the language Pascal-FC, which is a functionally concurrent extension of Pascal. The aim of this lecture is to present some highlighted topics of the semester, using parts of the electronic lecture-book.*

A párhuzamos programozás a „jövő tudománya”, amely a párhuzamos hardver architektúrák terjedésével egyre nagyobb gyakorlati fontosságot kap. Intézményünkben évek óta oktatjuk a párhuzamos programozás elméleti ötleteit és – szerényebb mértékben – gyakorlati módszereit is a hallgatóknak. A tárgy két éve a távoktatás tematikájában is megjelent, ezért szükségessé vált önálló tanulásra is alkalmas elektronikus jegyzet kidolgozása. Jelen előadás témája a tananyag egyes részeinek bemutatása a jegyzet néhány kiemelt részletének felhasználásával.

A párhuzamosság a számítástechnikában háromféle módon jelenik meg, ezeket tananyagunk is tartalmazza, közülük az utolsót kiemelten tárgyaljuk:

1. Szekvenciális algoritmusok párhuzamosítási lehetősége ill. párhuzamos algoritmusok
2. Párhuzamos hardver architektúrák
3. Párhuzamos programozás, ennek eszközei a Pascal-FC nyelvben, klasszikus problémák a párhuzamos programozásban, más párhuzamos nyelvek (pl. Ada)

A kurzus elvégzéséhez nem szükséges kiemelten sok előismeret, elegendő a minden informatikai képzés tantervében szereplő egy-két féléves „Algoritmuskészítés” című tárgy, amelynek célja, hogy a hallgatókat megtanítsa az

algoritmusok tervezésére és elemzésére, azaz az algoritmikus gondolkodás alapjaira. Szükségesek Pascal ismeretek, előnyt jelent más programnyelv(ek) ismerete. A tematika kidolgozásához angol és magyar nyelvű szakkönyveket, internetes forrásokat és saját tapasztalatokat használtunk fel. A hallgatók a félév során sok elméleti és gyakorlati feladatot kapnak.

### **Párhuzamos technikák, párhuzamos algoritmusok**

A párhuzamos számítások lehetősége az algoritmusok szervezésében sokkal nagyobb szabadságot biztosít, mint a szekvenciális megközelítés. Ezt az alábbi hasonlattal szemléltethetjük:

- a szekvenciális algoritmus egy dimenziós konstrukció, euklideszi térben;
- a párhuzamos algoritmus több (változó) dimenziós konstrukció, változó térben.

Jelenleg még elsősorban szekvenciális gépeken dolgozunk, ezért a párhuzamos algoritmusok főként a szekvenciális algoritmusokban amúgy is meglévő párhuzamosítási lehetőségeket használják. A párhuzamosítás alapvetően négyféle módon képzelhető el:

- a) triviális párhuzamosítási lehetőségek (nem feltétlenül triviális klasszikus algoritmusokban, pl. Gauss elimináció);
- b) klasszikus algoritmusok alkalmas részeinek vagy egészének párhuzamos végrehajtással való felgyorsítása (pl. mátrix-vektor szorzás, aritmetikai kifejezések párhuzamos kiértékelése);
- c) új elemek beépítése klasszikus algoritmusokba egyes részfeladatok párhuzamos megoldására (pl. aritmetikai kifejezések párhuzamos kiértékelése);
- d) teljesen új elveken alapuló párhuzamos algoritmusok kifejlesztése.

#### *A Gauss-elimináció*

A lépésszám és a műveletigény négyzetes mátrixra hagyományosan  $O(n^3)$ . Ha van  $n$  darab processzorunk, és ki tudjuk osztani a részfeladatokat úgy, hogy párhuzamosan dolgozzanak (például 1-1 sort egymástól függetlenül számoljanak végig), akkor a lépésszám  $O(n^2)$  lesz (ugyanakkor a műveletigény továbbra is  $O(n^3)$  marad). Tovább gyorsítható az eljárás  $n^2$  processzorral, ekkor minden mátrixelemhez rendelünk egy processzort, és az azzal dolgozik. Így a lépésszám  $O(n)$  lesz. Megjegyzendő, hogy a gyakorlatban a processzorok közötti kommunikációra fordított idő a fenti elméleti határokhoz képest jelentősen lelassítaná a feladat megoldását.

Feladat: Gyorsítható-e a végrehajtási idő újabb processzorok rendszerbe állításával? (1,5)

Mo: A probléma részeredményei tovább már nem függetleníthetők egymástól, tehát ha ennél több processzort használunk, akkor már nem érhető el sebességnövekedés.

Megj. Néhány fontos gyakorlati problémával nem foglalkoztunk, amelyek szintén komoly lassulást okozhatnak a megoldás során. Ilyen például az eliminációra sajátosan jellemző ún. együttható növekedési probléma: az egymás utáni lépésekben egyre több jegyű számokkal (gyakran egyre hosszabb törtekkel) kell dolgozni. Ennek részbeni kivédésére javasolhatnánk a közelítő számításokat, de ez sem oldja meg a gondot: eleve elveszítjük a pontos megoldás lehetőségét, ráadásul a közelítések hibái a lépések során összegződnek, így rossz esetben nem is kapunk megoldást a számolás végén! Csak ügyes egyszerűsítések alkalmazásával hárihatók el a fenti problémák általános esetben.

### *Polinom-faktorizáció*

Ld. az elektronikus jegyzet megfelelő része.

## **Párhuzamos hardver architektúrák**

### *A Flynn-féle osztályozás*

Az általános célú párhuzamos rendszerek legsikeresebb, napjainkban is széles körben elfogadott osztályozása M. J. Flynn nevéhez fűződik (1972), aki az adatok kezelése (hány adattal hány műveletet végzünk egy időben) alapján a következő kategóriákat állította fel:

	Egy utasítás – Single instruction	Több utasítás – Multiple instruction
Egy adat – Single data	SISD (Neumann-féle)	MISD
Több adat – Multiple data	SIMD	MIMD

Az MISD kategória üres, hiszen nehezen képzelhető el, hogy egyszerre több utasítás dolgozik ugyanazon az adatrészen. Egyes szakemberek azonban a pipeline (csővezeték) technikát használó egyes gépeket ebbe a kategóriába sorolják.

Flynn osztályozását később kiterjesztették az egyes kategóriák további megosztásával. Bevezették az SIMD csoporton belül a vektor- illetve a tömb-processzorokat, az MIMD-n belül pedig a multiprocesszorokat (van közös memória) és a multiszámítógépeket (nincs közös memória, csak saját).

Ld. az elektronikus jegyzet megfelelő része.



## A párhuzamos programozás fogalma

### A szekvenciális programozás jellemzői

A napjainkban is széles körben használt egyprocesszoros számítógépek a Neumann-elvet követik, amely szerint a processzor egyszerre csak egy utasítást hajt végre, és a végrehajtási sorrend előre meghatározott: mindig meg tudjuk mondani, hogy mi volt az előző utasítás és mi lesz a következő.

Egy magas szintű programnyelven írt program valamely egyszerű részlete a következő utasítássorozattal reprezentálható:

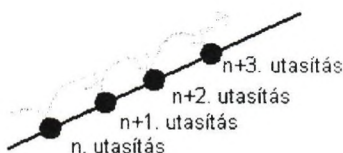
P;  
Q;  
R

Ebben a szemléltető példában P, Q és R egyes utasítások (például értékadások) vagy utasításcsoportok lehetnek. A program minden futtatására teljesül, hogy a P rész végrehajtása megelőzi Q végrehajtását, és Q végrehajtása megelőzi R végrehajtását (jelölés:  $P \rightarrow Q \rightarrow R$ ). Azt a jellemzőt, ami minden futtatásra teljesül, a program tulajdonságának nevezzük. Gépi szinten minden utasításcsoport, ami P-ben, Q-ban, illetve R-ben van, felírható elemi utasítások sorozataként. Így minden futtatásra az is teljesül, hogy

$p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow \dots \rightarrow p_m \rightarrow q_1 \rightarrow q_2 \rightarrow \dots \rightarrow q_n$ ,

tehát elemi szinten sem lehet átfedés az utasítások részei között. Igaz továbbá, hogy az elemi utasításokra fennáll a teljes rendezettség, azaz két elemi utasítás sorrendje minden futtatásra ugyanaz: előre meg tudjuk mondani, hogy  $x \rightarrow y$  vagy  $y \rightarrow x$  teljesül minden esetben.

A fentiek alapján a szekvenciális program végrehajtása egy egyenesen szemléltethető:



### Elszakadunk a szekvenciális programozástól...

Most tegyük fel, hogy sok processzor áll a rendelkezésünkre. A szekvenciális megközelítés ekkor nem minden esetben hatásos, illetve szükség-szerű.

Például tekintsük a következő egyszerű programrészletet:

```

X:=0;          (* P *)
Y:=0;          (* Q *)
Z:=0;          (* R *)

```

Ez esetben nem kell betartani a  $P \rightarrow Q \rightarrow R$  sorrendet, hiszen az értékadások egymástól függetlenek, így tetszőleges sorrendben végrehajthatók. A három programrész futhat azonos időben akár három különböző processzoron. Ha sorrendiséget állapítanánk meg közöttük, az túlspecifikálás lenne, azaz olyan megkötés, amely az algoritmus logikájából nem következik. Ugyanakkor ez a megközelítés nem alkalmazható teljesen szabadon minden esetben. Vannak olyan programrészek is, ahol fontos rögzíteni a végrehajtás sorrendjét, még akkor is, ha több processzor áll a rendelkezésünkre. Ezek a részek akár értékadások is lehetnek, ha például az egyik változó értékéből következik egy másik változó értéke:

```

X:=10;         (* P *)
Y:=X*2         (* Q *)

```

Általánosan tehát egy programban bizonyos esetekben megengedjük a párhuzamos végrehajtást, máshol pedig rögzítenünk kell a végrehajtási sorrendet az algoritmus logikája szerint. Lényeges, hogy ezt a rögzítést csak ott végezzük el, ahol feltétlenül szükséges.

A párhuzamos végrehajtás lehetőségének jelölésére egy új nyelvi struktúrát vezetünk be: `cobegin ... coend`, ahol a „co” a concurrent (párhuzamos) szó rövidítése.

A `cobegin ... coend` részben megengedett (de nem kötelező) a párhuzamos végrehajtás.

Egy egyszerű párhuzamos programrészlet lehet a következő:

```

J;
K;
cobegin      //concurrent begin
  P;         //itt nincs sorrendi megkötés, a végrehajtás le-
het párhuzamos
  Q;
  R;
coend;
L;
M

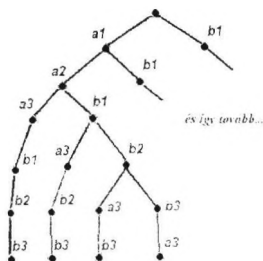
```

A programrészlet minden futtatására igaz, hogy  $J \rightarrow K \rightarrow (\text{cobegin ... coend}) \rightarrow L \rightarrow M$ , míg a `cobegin ... coend` struktúrán belül megengedett a párhuzamos végrehajtás.

Példa: Egy cobegin ... coend közötti programrészben két folyamat helyezkedik el (a és b), mindkettő három elemi utasítást tartalmaz. Hányféle végrehajtási sorrend lehetséges egyprocesszoros környezetben?

a: a1, a2, a3  
b: b1, b2, b3

```
cobegin
  A;
  B;
coend;
```



A teljes megoldás egy részletét az ábrán láthatjuk, de nemcsak rajzos úton kaphatjuk meg a lehetőségek számát.

Az elméletileg lehetséges 6!-féle sorrendből ki kell zárni azokat, amikor az  $a1 \rightarrow a2 \rightarrow a3$  és  $b1 \rightarrow b2 \rightarrow b3$  végrehajtás sérülne (ismétléses permutáció). Így a megoldás:

$$6! / (3! * 3!) = 20.$$

F: Hogyan változik a helyzet, ha kétprocesszoros környezetben dolgozunk? Segítség: itt három lehetőség megengedett az első szinten:  $a1, a1 \parallel b1$  és  $b1$ . Az elemzés hasonlóan folytatható lefelé haladva. (2,5)

Összefoglalva az eddigieket, a párhuzamos programban

- a szövegbeli elhelyezkedés nem határozza meg a műveleti sorrendet;
- a műveletek végrehajtása átfedheti egymást;
- különböző rendezettség tapasztalható különböző futtatások esetén, tehát a párhuzamos program nem determinisztikus.

Mindez a párhuzamos nyelv programozója számára meglepő következményekkel jár. A legfurcsább ezek közül talán az, hogy egy párhuzamos program azonos input adatokkal általában különböző eredményt ad (egyszerű példa: betűkiíratás). Párhuzamos környezetben a hibák sokkal veszélyesebbé válnak azáltal, hogy nehéz őket felderíteni. Egy nagy végrehajtási fa esetében gyakran előfordulhat, hogy egy-két „pici” hibás ág benntarad a programban. Nyilván nagyon kicsi az esélye, hogy a program oda ténylegesen be is fog futni, de gyakorlati teszteléssel ez nem zárható ki: akár 1000 hibátlan végrehajtás után is bekövetkezhet egy hibás. Ilyenkor tehát a teszteléssel a hibák elméleti előfordulását kell megakadályozni.

### Folyamat állapotok és átmenetek

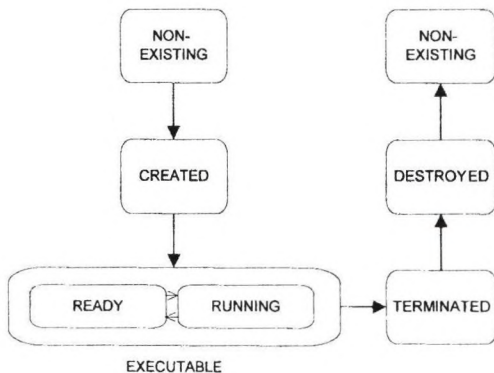
Bár a Pascal-FC nyelv folyamat modellje az egyszerűbbek közé tartozik, így is sokféle állapoton haladhat át egy adott folyamat az élete során.



Az állapotok közül a *created* (létrehozott) a folyamat deklarációja után, az *executable* (futtatható) a *cobegin ... coend* struktúra belsejében, de a terminálás előtt következik be. A *terminated* (terminált) állapot a folyamat futásának befejezése után, de a *cobegin ... coend* struktúra lezárása előtt él, míg *destroyed* (szétrombolt) állapotba a folyamat a *coend* után kerül. A program futásának befejezése után következik be újra a *non-existing* (nem létező) állapot.

A végrehajtható állapotot részletesebben is elemezzük, ez implementációs kérdéseket is felvet. Mivel nem feltétlenül ugyanannyi fizikai processzorunk van, mint amennyi folyamat, ezért a *cobegin ... coend* struktúra belsejében, a terminálás előtt

- egy adott folyamat nem biztos, hogy fut, ezért futtatható állapotban van;
- vannak futó (running) folyamatok és futásra kész (ready) folyamatok, utóbbiak közül választ az ütemező, ha felszabadul egy fizikai processzor.



Egyprocesszoros rendszerben például mindig csak 1 éppen futó folyamat lehet, a többi a készenléti állapotban várakozik.

Az *executable* állapot részeit tekintve nyilvánvaló, hogy egy folyamat a *created* állapotból először mindig a *ready*-be kerül át (az *executable* állapoton belül), és mindig a *running*-ból kerül át a *terminated* állapotba.

Kissé nehezebb a válasz a következő kérdésre: vajon mikor történhet *ready* ↔ *running* átmenet?

*Ready* → *running* átmenet akkor következik be, ha felszabadult egy processzor, és az adott folyamat kapta meg a futási jogot.

*Running* → *ready* átmenet akkor történik, ha időosztással dolgozik a processzor, és lejárt az adott folyamat időkerete (például többfelhasználós környezetben); illetve ha az éppen futónál fontosabb taszk indul.

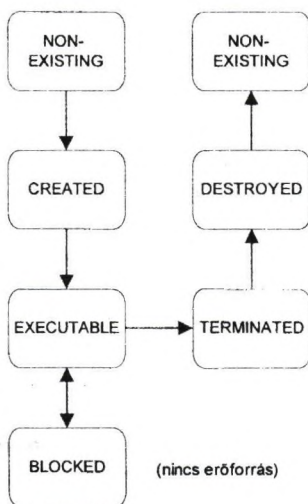
- Egyes taszkok időnként nem futtatható (non-executable) állapotban is lehetnek. Az ilyen állapotokat a továbbiakban blokkolt állapotnak nevezzük.

A blokkolt folyamat tehát már túljutott a *created* állapoton, de még nem terminált, és valamely esemény bekövetkeztére várakozik. Ennek alapvetően két oka lehet:

- a folyamat most nem akar futni (nincs dolga);
- a folyamat valamely erőforrásra vár.

Lényeges különbség van egy blokkolt és egy processzor hiányában nem futó folyamat között, hiszen újabb processzorok rendszerbe állításával az előbbi nem kerül automatikusan futó állapotba (példa: a biztonsági öv rögzítését végző taszk).

A folyamat életciklusát a blokkolt állapottal kiegészítve a következő ábrát kapjuk:



Az *executable* állapot részeit tekintve végiggondolhatjuk, hogy csak *running* → *blocked* → *ready* átmenet lehetséges, tehát például a blokkolt állapottól a folyamat először mindig a készenléti állapotba kerül vissza.

## Hogyan tanítsuk a programozási technikákat?

How Should We Teach  
Programming Technics?

Kátai Zoltán

SAPIENTIA-EMTE, Marosvásárhely

### Abstract

*The main pedagogical goal of this paper is to show how programming technics can be taught in a way that students look at them from an upper view. I mean by „upper view” that:*

- 1. I see entities being analysed standing next to each other*
- 2. We can see only what is important from the point of view of the analysis*
- 3. Similarities and differencies are evident, relationships are obvious*

*The method helps students see such minor but basic differences and similarities, too, in the programming technics, which otherwise might remain beyond their attention. Further, it enables the student to recognise weak and strong points of some technics and also to distinguish which should be used in certain situations and why.*

A dolgozat elsődleges pedagógiai célja, bemutatni, miként lehet úgy tanítani a programozási technikákat, hogy a tanulók, úgymond felülnézetből lássák őket. Ez segíthet nekik meglátni olyan elvi, alapvető és árnyalatbeli különbségeket, illetve hasonlóságokat a technikák között, amelyek különben elkerülhetnék a figyelmüket. A módszer bemutatását egy általános leírással kezdem, majd arról lesz szó miként alkalmazható a programozási technikák oktatásánál.

A bemutatott módszer kimagasló eredményekhez vezetett azokban az osztályokban amelyekben alkalmaztam, a tehetségesebbeket a megye, sőt az ország legjobbjai közé juttatva.

A tanulmány anyaga az informatikát oktató tanároknak személyesen is hasznos lehet. Összeállításakor olyan területek világosodtak meg előttem amelyeket azelőtt homály fedett az elmémben. Több kollegámnak is akik átolvasták a tanulmányt, hasonló tapasztalatuk volt.



## A módszer általános leírása

Egy felülnézet kialakításához szükséges lehet egy ún. „absztrakt platform”, amelyen az elemzett entitások úgy helyezhetők egymás mellé, hogy szembetűnővé váljanak a vizsgálat szempontjából lényeges tulajdonságok.

Az alábbiakban definiálom, mit értek felülnézet alatt ebben a cikkben:

1. „egymás mellett” látom a vizsgálat tárgyát képező összes entitást,
2. csak az látszik ami a vizsgálat szempontjából lényeges,
3. nyilvánvalóak a hasonlóságok és a különbségek, szembetűnők a kapcsolatok.

Egy ilyen felülnézet általában három lépésben valósítható meg:

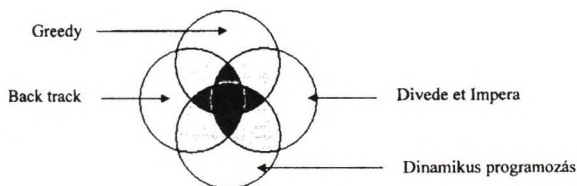
1. A tanulmányozandó entitások egymás mellé helyezése
2. Absztrakció által a tanulmányozás szempontjából lényegtelen részletektől való elvonatkoztatás
3. A hasonlóságok és különbségek kiemelése

A módszer célja olyan helyzetbe juttatni a tanulót, hogy ilyen rálátása legyen a tanulmányozás alatt álló anyagra. Természetesen különböző felülnézetek alakíthatók ki attól függően, miként valósítjuk meg az említett lépéseket. Ez kívánatos is, hiszen minden újabb felülnézet egy másik szemszöveget jelent. A módszer egyik erősségének számít az is, hogy segít a diákoknak a vizsgált entitások egymáshoz viszonyított „értékét” is felmérni. Például a programozási technikák tanulmányozása esetén nyilvánvalóvá válnak az egyes technikák erős- illetve gyengepontjai, valamint az, hogy adott helyzetben melyiknek az alkalmazása a legcélszerűbb és miért.

## Programozási technikák felülnézetből

Hogyan alakíthatunk ki felülnézeteket a programozási technikák (Greedy, Back-track, Divide et impera, Dinamikus programozás) tanításakor?

A kihívás abban áll, hogy minden egyes technikának megvan -többé kevésbé- a saját felségterülete. Az alábbi ábra ezt szemlélteti. Az egyes körök azon feladatok halmazát ábrázolják amelyek *megoldhatók* az illető technikával, *függetlenül attól*, hogy optimális megoldást nyújtanak-e vagy sem, hatékony az algoritmus vagy nem.



Vegyük észre, hogy a körök metszik egymást. Ez azt jelenti, hogy léteznek olyan feladatok amelyek több technikával is megoldhatók, sőt egyesek az összessel.

A következő példa azt szemlélteti, hogyan alkalmazható a módszer a gyakorlatban.

### 1. lépés

Az egyik legjobb módja ahogy a technikák egymás mellé helyezhetők, ha ugyanazt a feladatot oldatjuk meg velük. Például sokatmondó felülnézet alakítható ki, ha feladatnak az alábbi optimalizálási problémát választjuk, amelyet 1992-ben adtak Svédországban a Nemzetközi Informatika Olimpiáson:

*Feladat:*

Egy  $n$  soros négyzetes mátrix főátlóján és főátló alatti háromszögében természetes számok találhatóak. Feltételezzük, hogy a mátrix egy  $a$  nevű kétdimenziós tömbben van eltárolva. Határozzuk meg a „leghosszabb” csúcsból ( $a[1][1]$  elem) alapa ( $n$ -ik sor) vezető út, figyelembe véve a következőket:

- egy úton az  $a[i][j]$  elemet vagy az  $a[i+1][j]$  (függőlegesen le), vagy az  $a[i+1][j+1]$  elem (átlósan jobbra) követheti, ahol  $1 \leq i < n$  és  $1 \leq j < n$ .
- egy út „hossza” alatt az út mentén található elemek összegét értjük.

Például, ha  $n=5$  esetén a mátrix az alábbi, akkor a „leghosszabb” csúcsból alapa vezető út a besatírozott, hossza pedig 37:

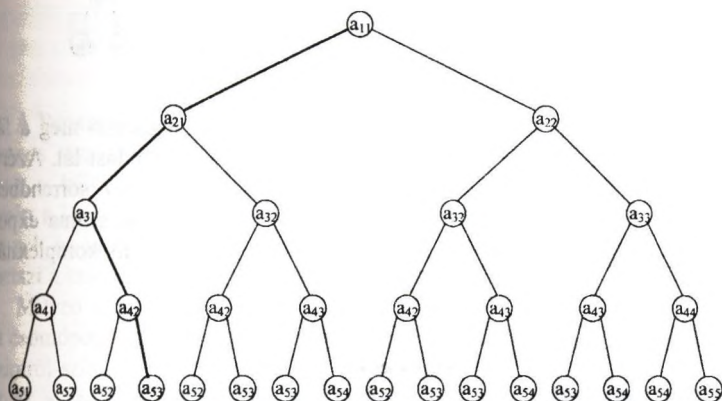
7				
5	9			
10	1	4		
2	7	3	1	
2	5	8	3	1

### 2. lépés

A felülnézet kialakításának második lépése az absztrakció. Ez a lépés elengedhetetlen ahhoz, hogy konkrét feladatok megoldásával, általános érvényű következtetésekhez tudjunk eljutni. Az alábbiakban azt mutatom be, hogy a jelen feladat esetében miként alakítható ki egy absztrakt platform,

amelyen majd szembeötlővé lehet tenni az egyes technikák közti elvi hasonlóságokat és különbségeket.

Megfigyelhető, hogy egy olyan optimalizálási feladatról van szó, amelyben az optimális megoldáshoz  $n-1$  döntés nyomán juthatunk el és mindenik döntésnél 2 választásunk van (melyik irányba lépjek tovább, függőlegesen le vagy átlósan jobbra). Az alábbi ún. megoldásfa, jól szemlélteti mindezt a példaként megadott mátrix esetében. Ezen megoldásfa fogja betölteni az absztrakt platform szerepét.



### 3. lépés

Azért, hogy szembeötlődjenek a technikák közötti alapvető hasonlóságok és különbségek, a következő két kérdéssel szabhatunk irányt a vizsgálódáshoz:

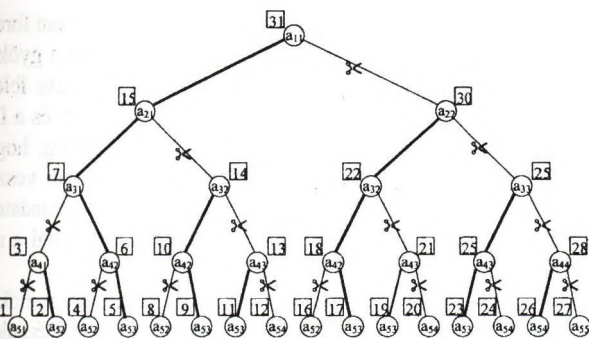
1.) *Hogyan járják be, és miként „metszik meg” a megoldásfát?*

(Az ábrákon négyzetekben megszámoztam, hogy milyen sorrendben foglalkoznak illetve metszik meg az egyes technikák a megoldásfa csomópontjait.)

**Greedy**, a gyökértől a levelek felé haladva metszi meg a fát, egész részfákat vágva le róla. Egyetlen gyökér-levelet jár be, a leggyorsabb algoritmust adva ezáltal a kezünkbe (komplexitása lineáris), de sajnos - legalábbis jelen esetben - semmilyen garanciát nem tud nyújtani arra vonatkozóan, hogy megtalálja az optimális megoldást.

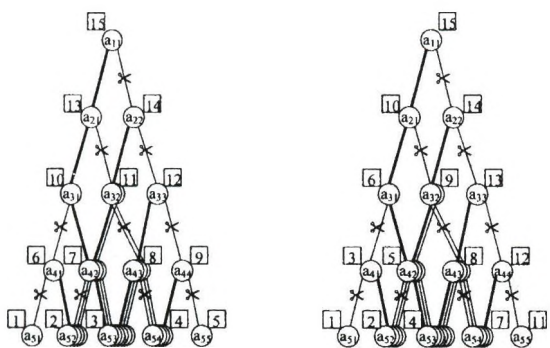






A Dinamikus programozás először is úgymond „egymásra helyezi” az azonos részfákat, létrehozva ezáltal egy „leszűkített” megoldásfát, amely csak a különböző részfeladatokat képviselő csomópontokat tartalmazza (ezért a Dinamikus programozás akkor tudja igazán értékesíteni erősségeit, ha sok az identikus részfeladat). Ezt követően mindenik csomóponttól lemetshi a száraz ágakat, akárcsak a Divide et impera.

Milyen sorrendben metszi meg a Dinamikus programozás a leszűkített fa csomópontjait? Klasszikus változatában lentről felfele halad végig a fán, szintről-szintre. A rekurzív megvalósítás ugyanabban a sorrendben foglalkozik a csomópontokkal, mint a Divide et impera, csakhogy kihagyja az ismétlődőket. Tekintettel arra, hogy a leszűkített fa csomópontjainak száma  $n(n+1)/2$ , algoritmusának komplexitása négyzetes.



2.) Hogyan építik fel a megoldást?

A Greedy illetve a Back-track technikák fentről-lefele építik fel a megoldást illetve a megoldásokat, levélben hirdetve eredményt.

A **Divide et impera** és a **Dinamikus programozás**, pontosan fordítva, *lent-ről-felfele* építik fel az optimális megoldást. Mindkét technika a gyökérben hirdeti eredményt, a Dinamikus programozás klasszikus változata felérkezve, a Divide et impera pedig visszaérkezve ide. A Divide et impera és a Dinamikus programozás rekurzív változata között csak annyi a különbség, hogy az első egymástól függetlenül oldja meg a részfeladatokat és ezért nem veszi észre ha ismétlődnek, a második viszont valahányszor megold egy részfeladatot eltárolja a megoldását, így ha újból találkozik ugyanazzal a részfeladattal a megoldása már rendelkezésre áll.

A **Back-track** és a **Divide et impera** közös vonása az, hogy mindkettő mélységbe járja be a megoldásfát. Mivel azonban ellentétes irányba építik a megoldást ezért az egyik preorder, a másik viszont posztorder sorrendben látogatja meg a csomópontokat. Ez magyarázza meg azt is, hogy miért generál ki a Back-track több potenciális megoldást is(amelyek közül ki kell választania az optimálist), a Divide et impera pedig miért épít fel csupán egyet, az optimálist: a fa lefele terebélyesedik, felfele viszont szűkül, sok levele de egyetlen gyökere van.

Mi egy másik alapvető különbség a **Greedy** és a **Dinamikus programozás** technikák között? Az első *optimális döntések* által, a második pedig *optimális részmegoldásokból* próbálja felépíteni az optimális megoldást. Ezért a Greedy megközelítés csak akkor kielégítő ha bizonyítani tudjuk, hogy az adott feladat esetében az optimális döntések tényleg optimális megoldáshoz vezetnek (a globális optimum lokális optimumukat feltételez). A Dinamikus programozás alkalmazhatóságának feltétele pedig az, hogy a feladat optimális megoldása valamilyen módon felépíthető legyen a részfeladatok optimális megoldásaiból (érvényes legyen az optimalitás alapelve).

### A módszer kiértékelése

Természetesen a módszer alkalmazása további felülnézetek kialakítását követeli meg, mind teljesebbé és teljesebbé téve ezáltal a képet.

A módszer a legnagyobb hasznot a csúcsképzésben hozta. A marosvásárhelyi Bolyai Farkas Elméleti Líceum mindig is szépen szerepelt országos és nemzetközi informatika versenyeken. A 2002-2003-as tanévben született eredmények viszont egyedülállóak. Az informatika olimpiász megyei szakaszán, X. osztályban úgy az első mint a második, illetve a harmadik helyezett abból az osztályból került ki, amelyben először volt alkalmazva a bemutatott módszer a programozási technikák oktatásakor. Ezen diákok közül az egyik a romániai döntőn harmadik helyezést, a magyarországi Nemes Tihamér verseny budapesti fináléján pedig második helyezést ért el. Ezek az eredmények összességükben példátlanok iskolánk informatika oktatásának történetében, és meggyőződésem, hogy nagymértékben a programozási technikák felülnézetből módszer alkalmazásának köszönhetőek.



## Mit adott nekem a Logo?

### What Have I Got by Logo?

Farkas Károly CSc

Szolnoki Főiskola és Budapesti Műszaki Főiskolák,  
Neumann Informatika Kara

#### Abstract

*The Logo language, and logo-pedagogy make us modern teacher. This study is useful for me not only in teaching but it has developed my knowledge, and my way of looking. In this paper I list some examples which was interesting and heuristic for me. These matters have come mostly from the turtle geometry, but there are some more in the field of mother language, music, even gymnastics as well.*

A Logo nyelv a logo-pedagógia fő eszköze, játéktere. A Logo nyelven keresztül a logo-pedagógiával való megismerkedés jelentette számomra a nevelélmélet, az oktatástudomány teljes megtanulását. A logo-pedagógia korábbi ismereteim, tapasztalataim nagy részét igazolta, megszilárdította, kétségeimet eloszlatta, új lehetőségeket nyújtott, alkalmazása során számos tényt felfedeztem. A Logo alapján látom az informatika alkalmazását a pedagógia megújulás, a robbanásszerű fejlesztés legjobb útjának. A logo-pedagógia segítségével tudom a gyermekeket úgy tanítani, hogy ők azt mindenkor csak játéknak, szórakozásnak élik meg. Ezekről a gondolatokról értekeztem eddigi tanulmányaimban. A Logo azonban nem csak pedagógiai ismereteimet, de matematikai, mérnöki, ismeretelméleti, sőt filozófiai tudásomat is bővítette, rendszerezte. Művelése során magam is átélhettem több tudományterületen a felfedező tanulás örömét. Most arról sorolok vázlatosan példákat, hogyan bővítette a Logo használata egy mérnök-tanár ismereteit, hogyan segítette elő rendszerszemléletének erősödését.

Csak nekem voltak ismeretlenek az alábbi tézisek? Ön például tudja-e, hogy:

1. Hétszög (n-szög) rajzolásához nem kell kiszámolnunk a csúcsszög nagyságát
2. Ugyanazon algoritmussal rajzoltatható konkáv és konvex sokszög, hatszög és hatágú csillag
3. A szabályos háromszög a hatszög egy különleges fajtája
4. Kör rajzolásához többféle algoritmus található a technöggeometriában is

5. Spirál, ciklois rajzoltatható „komolyabb” matematikai ismeretek nélkül is
6. A világegyetem leggyakoribb alakzata a spirál
7. A szinusz görbe a ciklois határértéke
8. A szimmetria invariábilis a dimenzió cserélése esetén
9. A kéményfalazás algoritmus a két sorban leírható

Érti, érzi Ön, hogy:

10. A prímszámok a számrendszer „alapkövei”
11. A cardioid és a nephroid között mi az azonosság, hogyan mozognak az égitestek

Képes:

12. A vízesés zsonglőrmutatvány elsajátítására
13. A Rubik kocka helyreforgatására?

Ad 1./ A szabályos  $n$  oldalú sokszög szintónikus rajzolás algoritmus a Logo használók körében jól ismert – a csúcok elérésekor a haladási irányhoz képest  $360/n$  szöggel kell elfordulni. Ez az algoritmus a számítógéphasználat, az informatikaoktatás didaktikájának egyik alaptörvényét indukálja, nevezetesen azt, hogy

a számítógép használatkor nem számolunk, hanem számoltatunk kell tudni.

Hétszög rajzolásakor az utasítást célszerűen így írom le:

ismételd 7[megy 20 jobbra 360 / 7]

(Nem számolom ki a hányadost!) A matematika tanítása során talán még jobban igazodhatnánk az informatika korához, amikor is a géppel való számolás, sőt a gép számoltatása válik általánossá. Az általános iskolában, ma is azt tanítják a gyerekeknek: törtet törttel, úgy osztunk, hogy az osztó reciprokéval szorzunk. Lehet, hogy ez volna a fontosabb: az osztót zárójelbe tesszük.

Ad 2./ Amennyiben a konkáv sokszöget is szintónikus szemléljük - amennyivel többet fordulunk az egyik csúcsnál a csúcok között „igazságosan elosztott” értéknél, annyival forduljunk kevesebbet a következő csúcsnál, - egyazon algoritmussal rajzolható úgy konkáv, mint konvex poligon.

Ismételd :n / 2 [megy 50 jobbra :n + :d megy 50 jobbra :n - :d]

Amennyiben a  $d$  differencia értéke  $k \cdot 360 + 360/n$ , és a  $k \cdot 360 - 360/n$  nyílt intervallumba esik, ahol  $k$  természetes egész, a sokszög konvex. A  $d$

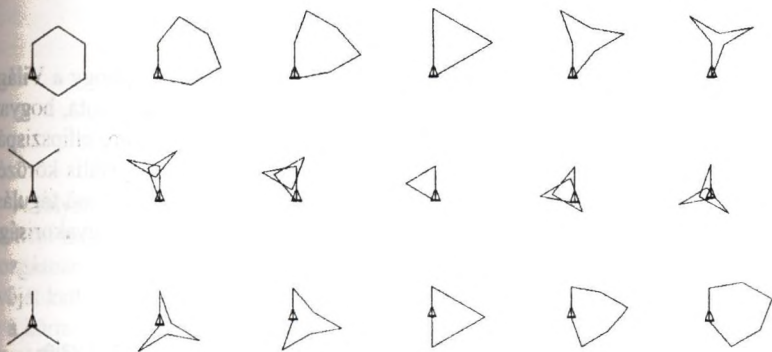
differencia értékének növelésével egy ideig erősödik a poligon sarkainak hegyesedése, majd csökken, ismétlődő intervallumokban az oldalak metszik egymást. Valamennyi sokszög egy családba tartozását szemléletesen mutatja a sokszögsorozat bemutatása

„kinematikus” geometriai szemléltetéssel.

Ehhez az eljárások a MicroWorlds programban írva ilyenek lehetnek:

```
to polygon :n :a :d
repeat :n / 2 [fd :a rt 360 / :n + :d fd :a rt 360 / :n - :d]
end
to kine :n :a :f
make "s 0
repeat 100 [polygon :n :a :s wait 10 cg make "s :s + 360 / :f / :n]
end
```

$n = 6$  esetén a  $a : d$  differencia értékét 20 fokonként változtatva a kapott sokszögek sora:



1. ábra  
6 oldalú sokszögek

Majd ez ismétlődik ciklikusan. A kinematikus poligon sorozat egyben egy pulzáló rendszernek is modellje.

Ad 3./ Az előbbi algoritmussal rajzolt poligonoknál, amikor a sokszög minden második szöge egyenesszög, az elfordulás értéke 0 vagy 180 fok, a



csúcsok (és oldalak) száma felére csökken. Először ez  $d$  egyenlő  $360/n$  értékénél következik be.

Ilyenkor válik a hatszög háromszöggé. (Az első ábra negyedik, tizedik, tizenhatodik alakzata.)

Ad 4./ A kör rajzolására a paperti algoritmusnál – ismételd sokszor, kicsit lépés, kicsit fordulsz – véleményem szerint szintónikusabb az alábbi kettő:

Ismételd sokszor [tollat\_fel megy sugárnyit tollat\_le megy kicsit tollat\_fel hátrál sugárnyit + kicsit fordul kicsit]

Két teknőcot szerepeltetek. Az egyikhez „Ádámhoz” hozzákötöm a másikat „Évát”. (Éva mindenkor Ádámmal szemben, tőle változatlan távolságban helyezkedik el.) Ádám forog. Az eredmény Éva köröz. Erről bővebben [1].

Ad 5./ Bonyolultabb matematikai görbék rajzolhatók a kisgyermek számára is érthető módon, két mozgás szuperponálásával. Ha „Éva” miközben „Ádám” forgatja, még haladó mozgást is végez, az abszolút mozgásának nyomvonala spirál. Ha az egyik teknőc egyenes, egy másik körmozgást végez, s egy harmadik mindkettőt, a harmadik nyomvonala, a mozgások eredője ciklois. [2]

Ad 6./ Feymann a „Mai fizika” című könyvében azt írja, hogy a Világegyetem leggyakoribb alakzata a spirál. Sokat gondolkodtam rajta, hogyan lehet igaza. Az égitestek közel gömbök. A bolygók körpályán, ellipszispályán mozognak. Először nem gondoltam arra, hogy a centripetális körzés csak relatív mozgás, az abszolút térben, a minden irányban történő tágulás, az ismétlődés és a gyorsulás a domináns mozgásforma. A spirál gyakorisága is a táguló világegyetem bizonyítéka.

A spirál rajzolatot eredményező alábbi algoritmus:

Ismételd sokszor [előre valamennyi, jobbra konstans, legyen a valamennyi több]

megmutatta, hogy amennyiben a fordulás értéke nem speciális (360 osztója), s miért volna kiténtetve az, sőt egyáltalán az egész szám, mindig megjelenik a spirál. Spirálokat kapunk akkor is, ha állandó elfordulás esetén a lépésmagyság növekszik (gyorsuló mozgás). A spirálokat, a teknőc rózsákat valószínűleg elsőként kezdtük el indáknak nevezni. [3]

Ad 7./ A nyújtott ciklois két mozgás eredőjeként is létre hozható. Az ötödik ponthoz írtakban megfogalmazott algoritmus MicroWorlds Logóban vagy LogoWriterben (változtatás nélkül!) ilyen lehet:

```
to vek :a :b
;;a szállító mozgás, :b relatív mozgás
;első-fekete eredő, második-piros szállító, harmadik-kék relatív görbe,
negyedik-zöld segítő
t1, if (abs xcor) > 320 [stop]
t1, if (abs ycor) > 160 [stop]
ask [t1 t2] [run :a]
ask [t1 t3] [run :b]
vek :a :b
end
```

Nyújtott cikloist kapunk a következő utasításra:

```
vek [setx xcor + .5][fd .25 rt 1]
```

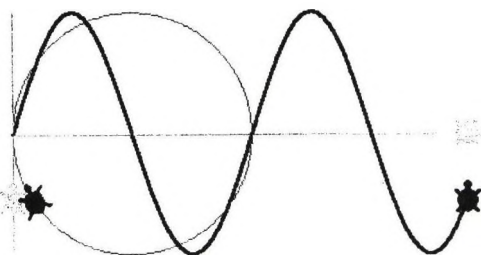


2. ábra  
Nyújtott ciklois

A folyamatosan, lépésről lépésre változtatgatott cikloisok nézegetésekor értettem meg, hogy a szinusz a cikloisok határértéke. A vízszintes szállító mozgásnak végtelen nagyra kell válnia, vagy a körmozgás vízszintes összetevőjét kell zérusra állítanunk. Ez utóbbi eset értelmezhető, megvalósítható, ha a körmozgás síkját a piros szállítómozgásra merőlegesen állítjuk. Szemléltetéshez egy negyedik zöld teknőcot hívjunk segítségül. A zöld segédteknőc (a képsíkba forgatva) köröz, és a kék teknőc a zöld kör függőleges vetületét mindenkor átvéve, annak megfelelő mértékben tér ki az x tengelytől, a vízszintes piros vonaltól. A kék teknőc tehát nem egyenletes sebességgel végzi alternáló mozgását, alternáló mozgása pulzáló sebességű. A fekete szinusz görbe: a vízszintes piros szállító, és a kék relatív mozgás eredője.

*A szinusz görbét ezek után így is előállíthatjuk:*

```
to alter
t4, fd 1.57 rt 1 ask [t1 t3] [sety ask "t4 [ycor]]
end
```



3. ábra

*Színusz görbe a ciklois határértéke. vek [setx xcor + 1][alter]*

Négyes (zöld) teknőc körözz, az egyest (fekete) és a hármast (kék) pedig arra kérem, vegye fel mindenkor a négyes y koordinátáját! Kettes (piros) a szállító, vízszintesen halad egyenletes sebességgel. [2]

Ad 8./ A téglalogo elnevezést a LEGO-Logo magyar változatára találtuk ki. A LEGO-Logo jelentése a nemzetközi szakirodalomban kettős. Egyrészt jelenti a LEGO elemekből készült modellek vezérlését, amit rendszerint valamely Logo implementációval hajtanak végre. Az LCSi mindenkor kelően foglalkozott a robotika konstruktív pedagógiai megközelítésével is. Papert LEGO professzor címet is használ. Másrészt jelenti a LEGO-Logo a LEGO vagy lego-szerű elemekből való építkezést Logo programnyelv alapján. A téglalogo alkalmazását igen sok szempontból javasoljuk. Ezzel egyszerűen, egyaránt, egymás hatását erősítve, holisztikusan, játékosan fejleszhető az algoritmizáló készség, a strukturális gondolkodás, a térszemlélet, a manipulatív készség. A rak jelenti a téglalogo lerakását, bebetonozását, a tovább „a téglalogo orra” irányában való haladást, a téglalogo négy egység hosszú, a föl a felfelé emelkedést, a téglalogo vastagsága egységnyi.

Téglalogókból építsünk lépcsőt:

Ismétel 3[rak tovább 2 föl 1]

Amennyiben a haladás dimenzióban előjelet váltunk – tovább 2 helyett hátrál 2 - az építmény az eredeti tükörképe lesz, s ugyanilyen tükörképet kapunk, ha a másik dimenzióban, a fel-le irányban váltunk előjelet – föl 1 helyett alá 1 a parancsunk.

Ezt szemléltetni (és a feladatmegoldásokat értékelni) igen egyszerű. [4]



Ad 9./ A kéményfalazás algoritmusá látványos, elegánsan rövid a téglalogóban.

Tudd első\_sor  
ismétel 4[rak tovább 4 jobbra]  
Vége

Tudd első\_emelés  
föl 1 jobbra  
Vége

Tudd második\_sor  
ismétel 4[rak tovább 4 balra]  
Vége

Tudd második\_emelés  
föl 1 balra  
Vége

Tudd kéményelem  
első\_sor első\_emelés második\_sor második\_emelés  
Vége

Ad 10. 11./ A prímszámok jelentőségét mutatja a teknőcrózsákkal való kísérletezgetés is. Ezt többször publikáltuk. [3] Ehhez hasonló didaktikai lehetőséget sejtünk a bolygómozgások modellezésekor. [2] Ennek a hipotézisünknek az igazolásán dolgozunk.

Ad 12./ A Logo nem csak matematikai ismereteimet növelte. Jelentős hatása volt a nyelvi gondolkodásmódom fejlesztésében is. Lényegesnek tartom a Logo alapszavak fordításakor a magyar megfelelők megválasztását. Ennek fontosságát tapasztaltam. A két Magyar hazában legelterjedtebb Logohoz, a Comenius Logohoz mi is elkészítettük az általunk javasolt magyar utasításszavakat. A célszerűnek tartott magyar utasításszavakat alkalmazzuk a MicroWorlds Logo magyaráításakor is. [5]

A Logo jó eszköz volt számomra a gyorsolvasást tanító programok elkészítésére. A Gyorsolvasás programlemez Logo nyelven készült gyakoroltató feladatokat, játékokat tartalmaz. [6]

A Logoval való játék fejlesztette zenei műveltségem is. Az ének-zene programcskákvaló játék, azok megírása is nyilván fejlődően hat a hallásra, a ritmusérzékre.

A vizesés zsonglőrmutatvány algoritmizálás segítségével való megtanulását Papert az Észrengés című könyvében ismerteti. Többen kipróbáltuk és igazolhatjuk a mester állítását: mozgássorozatok könnyebben megtanulha-

tók, amennyiben azokat logoszerűen algoritmizáljuk. Aki tornázott vagy más sportot gyakorolt tapasztalhatta, hogy a művészi mozgás nem csak fizikai tevékenység. [7]

Ad 13./ Ehhez hasonlóan a Rubik kocka helyreforgatás is segíthető valamely – célszerűen Logo nyelven megfogalmazott – algoritmus alkalmazásával, s ezen „mankó” használatával bejárhatjuk azt az utat, amely után már jobban átlátjuk a bűvös kocka mikrovilágát. [8]

### Hivatkozott irodalom

- [1.] Farkas K.: *Miért kaphat „Éva” időnként idegösszeomlást?* HungaroLogo 2003. Konferencia kiadvány. NJSZT. Budapest, 2004.
- [2.] Farkas K.: Logo-pedagógia. *Iskolakultúra*, 2003. október. p. 21-37.
- [3.] K. Farkas: *The World of Turtle Roses*. Ivan Kalaš ed.: *Acta Didactica Universitatis Comenianae*. Informatics, Issue. Pozsony, 1994. Comenius University Slovakia. p. 63-67.
- [4.] Farkas K.: *Játékos informatika*. Kandidátusi disszertáció. D17799 I-II, 1993. p. 169. Appendix 1996. p. 55.
- [5.] HungaroLogo 2002 CD, NJSZT titkárság vagy <http://mwlogo.freeweb.hu/>
- [6.] Gyorsolvasás CD JIO Bt, <http://gyorsolvasas.freeweb.hu>
- [7.] Papert S. *Mindstorm*. Children, Computers and Powerful Ideas. Basic Books. Second Ed. 1993. Magyarul.: *Észrengés*. A gyermeki gondolkodás titkos útjai. SZÁMALK. 1988.
- [8.] K. Farkas: *First steps in education of Informatics*. Eurologo'95, 5<sup>th</sup> European Logo Conference Birmingham, England, 1995. p. 65-70.

# Játékos Informatika – Logo és programozás

## Playful Informatics – Logo and Programming

Törtely Éva

Templom Téri Általános Iskola, Pilisvörösvár

### Abstract

*Logo happens to be great fun. The user is completely in charge of what goes on; not the computer. Logo is totally open to your imagination. It is easy to make a very simple one-shape animation. The turtle will move forward. Alternatively, you can animate your turtle by making it „wear „ a number of shapes and moving. The Logo (MicroWorlds Pro) is a great way for kids to get started on programming. In MwPro you have some methods of playing multimedia. The children can make an animated scene with graphics, a bit of text, a simple animation and some music.*

„A Logo tudása nem egy eljárás ismeretét, nem szabályok bevésését, nem jónak kikiáltott, még csak nem is az eddig jól bevált algoritmusok alkalmazásának begyakorlását, számonkérését (dolgoztatírás?) jelenti. A Logo tudása a gyermeki nyitottság, a kreativitás, az útkeresés örömeinek bemutatását és megőrzését, játékot, a kutatás örömeinek, a kísérletezőkedvnek, a kreativitásnak az élvezetét, az önbizalomnak, a fantáziának, az informatika szeretetének birtoklását, a hatékony gondolkodásra való képességet, a technika helyénvaló (nem idegenkedő és nem fetisizáló) szemléletét, saját gondolkodásunk és a világ megismerésének egy kellemes módját jelenti.<sup>1</sup>” Ezek a mondatok a szívből szólnak. Csak akkor tudjuk tudásunkat a gyerekeknek hatékonyan átadni, ha a felnőtt (a pedagógus) meg tud maradni lélekben gyermeknek.

Egy olyan témát szeretnék bemutatni, melyben tág tere van a játéknak, fantáziának, kísérletezésnek. A gyerekek – kicsik, nagyobbak egyaránt – nagy élvezettel foglalkoznak az animációkészítéssel. Az általam ismert Logo változatok közül erre a MicroWorlds Pro-t tartom a legalkalmasabbnak.

Amikor a programot elindítjuk, teljesen üres munkaterületet látunk. Még teknőcünk sincs, tehát nem vagyunk „rákényszerítve” a teknőcgrafika használatára. A rajzfilmkészítéshez azonban teknőcre van szükség, ezért először teknőcöt kell „teremtteni.

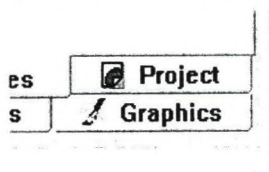
Animációt többféleképpen készíthetünk, nézzük először a legegyszerűbbet.

<sup>1</sup> Farkas Károly: Logo-pedagógia Pedagógiai informatika, tanítás a Logóval Iskolakultúra 2003/10. 26.p.



## Egy alakos animáció mozgással

A képernyő jobb oldalán kattintsunk a Graphics fülecskére.



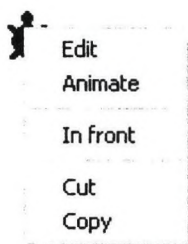
1. ábra

Válasszunk a teknőcnek áruhát.



2. ábra

Kattintsunk először a választott alakra, majd a teknőcre. Jobb egérgombbal kattintsunk a teknőcre.



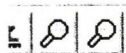
3. ábra

A megjelenő menüből válasszuk az Animate parancsot. A teknőc elindul.  
*Feladat: Készítsünk akváriumot!*

1. Rajzoljuk meg a háttérrel.  
A MwPro-ban kéznél vannak a rajzeszközök. Nincs szükségünk külön rajzóprogramra.
2. Teremtsünk teknőcöt, fordítsuk a megfelelő irányba, változtassuk hallá.

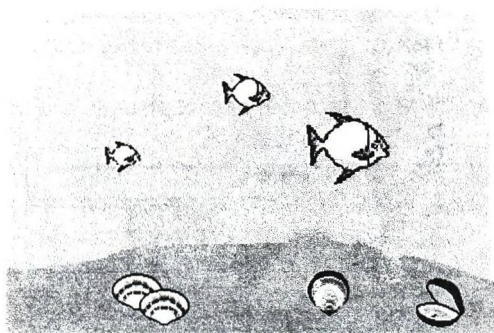
### 3. Animáljuk.

Ha több halat szeretnénk az akváriumba, ismételjük a 2- és 3. lépést. A teknőcök (halak) mérete változtatható a nagyítóval.



4. ábra

Miközben az akváriumot készítetem, kisfiam a következő javaslatokat tette: a nagy hal ne takarja el a kis halat, a kicsik ússzanak gyorsabban, legyen benne más állat is. A befejezett rajzfilm egyetlen képkockája:



5. ábra

A halak mozognak, a kagylók csak díszek.

### Mozgás és alakváltoztatás

A rajzfilmkészítés lényege az, hogy rajzolunk egy képet, majd rajzolunk egy másikat, amely nagyon kicsivel tér el az előzőtől. A második ábrán látható, hogy vannak olyan ábruhái a teknőcnek, amelyekkel igazi rajzfilmet alkothatunk (pl. a macska).

*Feladat: Sétáló cica*

Az ábruhák igen egyszerű módon összekapcsolhatók. A munka menete:

1. Válasszuk ki az első macskát.
2. Kattintsunk a teknőcre.
3. Válasszuk a következő macskát.

4. Nyomjuk le a SHIFT gombot és kattintsunk a teknőcre.

Amennyiben még több darabból áll az áruha, ismételjük a 3. és a 4. lépést. Ha animáljuk a teknőcöt, a macska végigsétál a rajzlapon.

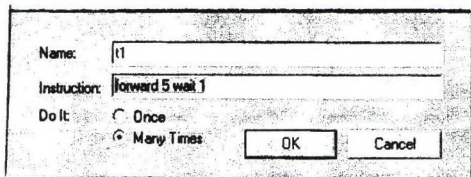
Mindeddig Logo parancsokról, programozásról szó sem esett, egy írni-olvasni nem tudó kisgyerek is sikerélményhez juthat.

### *Mitől mozog a teknőc?*

„... sok iskolában programnyelvet tanítanak, Comenius Logót, programozási szabályokat, vagy éppen a Logo versenyre készítenek fel.<sup>1</sup> Comenius Logo helyett persze írhatnánk Logowritert, MSW Logót, vagy akár MicroWorlds Pro-t is, ha rosszul használja a pedagógus. „Betanítanak, megpróbálják a gyermekek gondolkodását forgatókönyv szerint fejleszteni.” A programnyelv van az emberért és nem az ember a programnyelvért. A Logo nem CSAK Teknőc-grafika! „Több helyen a Logo iskolás lett, tanítási skolasztikus.”<sup>2</sup>

Mi eddig felhasználói programként kezeltük a MicroWorlds Pro-t. Logo parancsokról, programozásról szó sem esett. Pedig érdekes kérdés, hogy mitől mozog a Teknőc?

Jobb-klikk a teknőcre, s a menüből az Edit parancsot válasszuk ki.



6. ábra

A teknőcnek van neve és átnevezhető. Rendelkezik utasításokkal, melyeket végrehajthat egyszer, illetve sokszor. Kísérletezzünk! Milyen lesz az állat mozgása, ha az előrehaladás mértékét, s milyen, ha a várakozási időt változtatjuk meg.

### **Animáció haladóknak**

Mindazt, amit eddig játékosan hoztunk létre, programnyelven is megadhatjuk.

---

<sup>1</sup> U.o.

<sup>2</sup> U.o.



Egy lépéssel közelebb a programozáshoz

Mutassunk rá az egyik teknőc áruhára.



7. ábra

A megjelenő felirat tudatja velünk, hogy a ruha neve bird1 (ha nem tetszik az angol szó, át lehet nevezni másra), a 36-os számú fogason lóg, mérete 72x64. Nevezzük át a teknőcöt, és az utasításorba írjuk be a következőt:

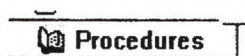
A dialog box with the following fields and options:

- Name: kolibri
- Instruction: setshape 36 wait 1 setshape 37 wait 1
- Do It:  Once,  Many Times
- Buttons: OK, Cancel

8. ábra

Az animált madárka egy helyben repül, ahogy a kolibri.

Előfordulhat, hogy parancsaink nem férnek el az utasításorban, más megoldást kell találnunk. Írjunk eljárásokat! Kattintsunk a Procedures fülre.



9. ábra

A képernyő jobb oldalán megjelenő fehér téglalapra írjuk azokat a dolgokat, melyeket a Teknőcnek meg kell tanulnia.

```
to repül
setsh 36 wait 1
setsh 37 wait 1
end
```

10. ábra

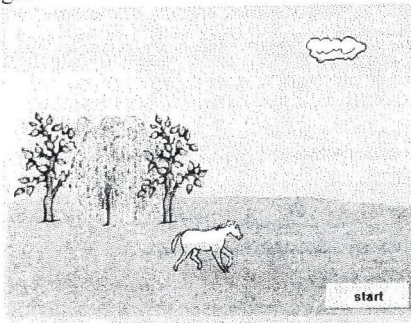
Több eljárással komolyabb programokat építhetünk. Már ebből a három eljárásból is látszik a Logo lényege: eljárásokat használunk, az eljárásokból építkezünk.

```
to start
ló üget
end
```

```
to ló
setshape [horse1 horse2 horse3
horse4]
end
```

```
to üget
fd 0 wait 2
üget
end
```

Ha több teknőcünk van, mindnek külön el kell mondanunk, hogy mi a dolga.



```
to teknőc
t1, setshape [horse1 horse2 horse3
horse4] seth 90 infront
t2, setshape "fák seth 270
t3, setshape "cloud seth 90
end

to mozog
t1, forever [fd 0 wait 1]
t2, forever [fd 2 wait 1]
t3, forever [fd 3 wait 1]
end

to start
teknőc mozog
end
```

11. ábra

Találjunk ki meséket, készítsünk a mesékhez rajzfilmeket! „Gyermekeink lelkét gazdagítja a mese, az ének és a tánc. Persze tudom, hogy számukra fontos a számítógép és a nyelvek is, de nem szabad elfelejtenünk, hogy egy lélek nélküli nép elveszti önmagát és irányítójét.”<sup>1</sup> Írja Budai Ilona. A mesékben a szereplők hatással vannak egymásra, befolyásolják egymás cselekedeteit. Most tehát nem azért beszélünk feltételes elágazásról, mert valamely külső „erő” ránk tukmálja, hanem azért, mert a cselekményhez szükségünk van rá. Először mindig az anyanyelvünkön fogalmazzunk!

Ha az egyik megérinti a másikat, akkor...

Ha találkozik egyik a másikkal, akkor...

Amikor már kellőképpen tisztáztuk, hogy mit akarunk, akkor jön csak a programnyelvi megfogalmazás.

```
if touching? „t1 „t2 [ask [t1 t2] [...]
```

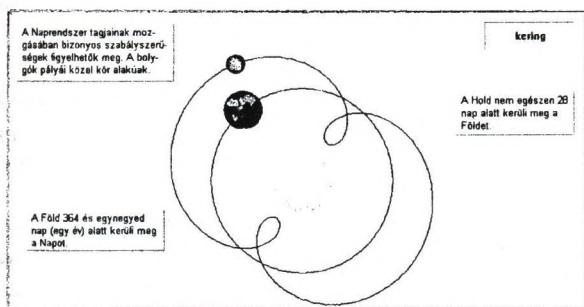
<sup>1</sup>Budai Ilona: Székely mesebeszéd – Allprint Kiadó 2003. 6.p.

## Multimédia

Eddig elkészítettük a hátteret és az animációt. Hogy teljes legyen a munka, írjunk le az animációhoz tartozó történetet, adjunk neki címet. Kiegészíthetjük még zenével is, illetve - hangfelvételt készítve – „alámondás”-sal is.

*Feladat: Naprendszer*

Farkas Károly cikkében találtam rá arra, hogyan lehet a Logóban a bolygómozgást szemléltetni. Miért ne használhatnánk fel mások eljárásait az animációkészítés közben?



12. ábra

Ami még igazán nagyszerű a MicroWorlds Pro-ban, hogy ablakot nyit a világra, hiszen munkáinkat a Világhálón is bemutathatjuk. Tanítványaim néhány alkotását két helyen is megtekinthetik: a [templomteri-bp.sulinet.hu](http://templomteri-bp.sulinet.hu), avagy a [templomteri.uw.hu](http://templomteri.uw.hu) címen.

## Irodalom

- [1.] Farkas Károly: Logo-Pedagógia Pedagógiai informatika, tanítás a Logóval, Iskolakultúra 2003/10.
- [2.] Papert, S.: Észrengés. A gyermeki gondolkodás titkos útjai SZÁMALK 1988.



## **Számítástechnika tanár = Internet pedagógus !? (vagy csak a tanár dilemmája)**

**Informatics Teacher is Equal to Internet Educator?!  
(Or it's Just Only a Teachers Dilemma)**

**Nagy Imecs Vilmos**

Székelyudvarhely

### **Abstract**

*Until the twentieth century the teachers were known as only a knowledge-communciator, and the students as receptors of a ready information. Only in the beginning of the twentieth century were these facts changed by the reform-education.*

*By Papert the computer is capable to invent a micro-world, in which the students can learn more and more active. The Internet can have the same influence on teaching, as these new computers had. The revolution of the Internet had changed also the method of teachers educating skills. At the beginning there were some myths linked to the concept of Internet, which were hiding the important contents from the users eye. This kind of myths are for example: the network democracy, the knowledge of ethical rules, the appearance of cyber-society.*

*For the Internet teachers every day is a challenge to involve the Internet more and more in education. Educators must be nowadays people, who can build the Internet in the learning-teaching process, to get a quality change for the better.*

### **Történeti áttekintés**

Visszatekintve a neveléstörténet elmúlt évszázadaira a következő alapvető tanulásfelfogás és pedagógiai elméletrendszer alakult ki:

#### **1. Az ókor és középkor tanulásszemlélete**

A könyvekből és tanítók szavai alapján történő, tekintélyelvű, merev, dogmatikus elsajátítás. Önállóság, kreativitás nem lehetséges, hiszen az ellentmondana az erre a korra jellemző, minél pontosabb reprodukálás elvének. A megismerés logikája deduktív.

## 2. Comenius didaktikája

A tanítás lényege a szemléltetés, a tanításnak az érzékszerveken keresztül kell hatnia a tanulókra. A megismerés logikája induktív. Tanárszerepek: mester (a szakember), atya (a lelki vezető), barát.

Meghatározzák a didaktikát: ógörög eredetű, didaskein (oktatni, tanítani) szóból származik. Értették alatta a tanítást, tanulást, útmutatást, mint a nevelő, a tanító tevékenységét, a tanulást, mint a növendék tevékenységét. Mivel a tanítási, tanulási folyamat iskolában zajlik, ezért a didaktika a szervezett tanítási, tanulási tevékenység eredményeként határozható meg

## 3. Reformpedagógiai irányzatok

A gyermek személyiségének, tevékenységének középpontba kerülése jellemzi. Az eddigi passzivitást aktív, szabad tevékenység váltja fel, amelyben a tanár feladata a segítségnyújtás, az önálló ismeretszerzés feltételeinek megteremtése. Az érzékelés helyett a cselekvés foglal el központi szerepet. Piaget szerint a megismerés nem más, mint a valóság viszonyainak interiorizációja (belsővé válása), ez pedig csak cselekvés által válhat valóra. A megismerés logikája itt is induktív jellegű.

A fentieket összegezve általánosan megállapíthatjuk, hogy a tanár elsősorban ismeretközlelőként, a diák pedig a készen kapott ismeretek passzív befogadjaként állt hosszú időn keresztül, és ezen csak a XX. század elején kezdődött reformpedagógiai kísérletek próbáltak változtatni

A felsorolt didaktikákhoz csatlakozhat napjaink egyik jelentős irányzata, a **konstruktív pedagógia** is, amely kísérletet tesz arra, hogy a tanulók eddigi passzív szerepét megváltoztassa, ami által a tanár feladata és funkciója is változáson megy át. A megismerés dedukcióra és a cselekvések által létrejövő konceptuális változásokra épül. "Papert szerint a számítógép képes arra, hogy ... olyan mikrovilágot teremtsen, amelyben a gyerekek ... kötetlen, elemi érdeklődésből fakadó, aktív tanulási módját valósíthatják meg."

Hasonló szerepváltoztató hatást idézet elő az **Internet széles körben** való elterjedése.

### Az Internet megjelenése - mindent megváltoztat!

Ha csak az elmúlt évtizedek oktatástechnológiájának fejlődését nézzük, azt tapasztalhatjuk, hogy időről időre egy-egy új eszköz került előtérbe, amit "csodaszereként" emlegettek, amely megváltoztatja az oktatás hatékonyságát. Ezek az eszközök:

- audiovizuális eszközök,
- filmvetítő,
- írásvetítő,
- nyelvi labor,
- személyi számítógépek

sorra beilleszkedtek a hagyományos oktatás kelléktárába.

Némelyek mára elavultak, „kikoptak”, mások jelenleg is használatban vannak. *De vajon beváltották-e a megjelenésükkor hozzáfűzött reményeket?*

Az Internetet valóságát, „valódi arcát”, csakúgy mint magát a számítástechnikát, bizonyos fajta mítoszok burkolják be, amelyek alkalmasak arra, hogy elfedjék a felhasználók elől a lényegét. Ilyen mítoszok például: az esélyegyenlőség és a hálózati demokrácia megteremtődésének ígérete, új etikai szabályok megtanulása (netikett) az új hálózati (cyber-) társadalom létrejötte, a kényelmesebb élet ígérete, vagy éppen olyan egyszerű tévhitek, mint hogy az Interneten minden megtalálható, stb. A hálózat lerombolja az emberek egymástól való távolságának korlátait, az információkhoz való jutás korlátait, megváltoztatja az egész társadalmat, és ennek részeként természetesen az oktatást is., ám érintetlenül hagyja az emberi természetet. (semmiképpen sem kényszerít bennünket valamiféle steril, digitalizált világba). Hozzásegít, hogy értelmi és érzelmi életünk kiteljesedjék, de nem változtatja meg természetünk alapvető jegyeit. Ezek a vélemények Esther Dyson-tól, a "digitális világ" egyik legbefolyásosabb emberétől hangzottak el. .

### **Számítástechnika tanár= Internet-pedagógus?**

Próbáljuk meg definiálni, mit takar a fenti fogalom.

1. Aki órán használja az Internetet? NEM.
2. Aki otthon információkat gyűjt a Netről? NEM.
3. Aki levelezőlistákon szerepel, netán elkészíti saját, vagy iskolájának weblapját? NEM
4. Aki web címet ír fel a táblára hogy a diákok használják? NEM
5. Aki felhívja a diákok figyelmét bizonyos új szoftverekre ? NEM

### **Egy mondatos válasz**

"Olyan tanár, aki képes az Internetet, mint eszközt úgy beépíteni a tanítási-tanulási folyamatba, hogy ezzel minőségi változást idézzen elő."

Két kérdés vetődik fel:

1. az Internet, mint eszköz kérdése
2. a minőségi változás kérdése



## Hol a szakadás?

A világszerte folyó iskolai Internet-programok mindegyikében az Internet, mint cél jelenik meg elsőként. Az eszközként való használat csak abban az esetben válik realitássá, ha egyrészt az eszközök megfelelő számban és minőségben állnak rendelkezésre, másrészt pedig a tanárok megfelelő tudással, hozzáértéssel és motivációval rendelkeznek.

Itt számos kérdés tevődik fel.

1. Mennyi számítógép szükséges a hatékony használathoz és honnan kerülnek ezek az iskolákba?
2. Milyen szoftvereket és tananyagot használjanak a pedagógusok?
3. Milyen társadalmi hatást gyakorol a számítógépesítési program?
4. Van-e összefüggés az iskolai számítógépesítés és a munkaerőpiaci igények között?

És ami a számunkra most a legfontosabb:

- a tanárok képzésének és továbbképzésének kérdésköre.

Alkalmasak-e a jelenlegi iskolai tanárok az új módszerek használatára?

- Ha nem, hogyan képezhetők ki erre?

- Milyen továbbképzési formák hatékonyak?

Térjünk vissza második tényezőhöz, vagyis ahhoz, miként határozható meg, milyen minőségi változásról lehet szó? Soroljunk fel néhányat ezekből:

1. Motivációs bázist épít ki a tanuló számára (ugyanis tudásszerzésre motiválhatja) Nem elhanyagolható szempont, ha figyelembe vesszük, hogy napjainkban a média és a tömegkommunikációs eszközök milyen hatást fejtenek ki a társadalomra, és különös tekintettel a fiatalokra .
2. Egyetemes szemléletet ad, kitérít az egyén életterét - mintegy ablakot nyit a világra, elérhetővé tesz számos, addig elérhetetlennek tartott dolgot. A tanulók aktivizálásának egyik legérdekesebb, leghasznosabb eszköze az ún. *online-project-ek létrehozása*????, illetve az ilyenekben való részvétel.
3. Értékeket közvetít, változtat, értékrendet befolyásoló szerepet kap. Ebben döntő fontosságú a *tanári orientáció és hatás!* Ezt a változást ugyanis NEM a számítógép biztosítja, hanem azok az információk, és személyek, személyes kapcsolatok, amelyek a hálózat igénybevételével alakulnak ki.

4. Áthidalhat társadalmi (szociokulturális) különbségeket. Ez azonban csak akkor lehetséges, ha a hátrányosabb helyzetben levő diák is hozzáféréssel rendelkezik. Ez ellen hat az a tény, hogy a jelenlegi állapotok szerint bizonytalannak látszik az általános iskolák gépekkel való ellátása.

Nézzük ezek után, milyen tulajdonságokkal kell(ene) rendelkeznie annak a tanárnak, akit Internet-pedagógusnak nevezhetünk.

1. kritikai érzék
2. jó angol nyelvtudás
3. magas szintű számítástechnikai ismeretek
4. elemzőképesség, információ rendszerezési képesség
5. magas általános műveltség
6. etikusság, erkölcsösség, felelősségérzet
7. tolerancia
8. igényesség (nyelvhasználatban, tartalomban, képi világban - belső igényesség)
9. képesség arra, hogy az információt olyan tanulási környezetet ("learning environment"), amely tevékenységre, aktivitásra és kreativitásra ösztönzi a tanulókat (motivációs képesség)

Adjuk hozzá a felsoroláshoz azokat a követelményeket, amelyeket Pólya György állít a jó tanárral szemben:

1. érdekeljen a szaktárgyad.
2. ismerd a szaktárgyadat.
3. a tanulás legjobb útja az, amelyet magad fedezel fel.
4. ne pusztán tárgyi tudást adj tanítványaidnak, hanem fejlesszed gondolkodási készségüket is, szoktassad megfelelő értelmi megatartásra, rendszeres munkára őket.

Hogyan segíthetünk ezeken a gondokon ?

### **Külföldi példák**

Az **Egyesült Államokban** az elemi iskolai tanárok 75%-a, a középiskolai oktatók több mint 90%-ának van otthon is számítógépe, ami jelentősen megkönnyíti, hogy a gépet eszközként is használhassák munkájukban

**Magyarországon** nagyon hasznos lépés volt az 1999-ben elindított pedagógus számítógépes program, amelynek keretében több mint tízezer tanár jutott hozzá korszerűen felszerelt hardvereszközökhöz és alapszoftverekhez. Ennek ellenére valószínűsíthető, hogy a tanárok otthoni számítógépes ellátása

tottsága még mindig 50% körüli vagy annál alacsonyabb. Ez az arány a tapasztalatok alapján magasabb a középiskolai tanárok körében, és jóval alacsonyabb az általános iskola alsó tagozatán tanítók, illetve óvodapedagógusok esetében.

Lássuk ezek után, hogyan változik meg a tradicionális tanár-diák viszony, pontosabban milyen lesz az Internet-pedagógus - diák viszony. A tanár az eddigi előadóból, új szerepkörbe lép át, irányító, tanácsadó, konzultáns lesz.

### **Változó iskola, változó diák – változó tanár?**

Az iskolarendszer világszerte lassan végbemenő folyamat, ami nem kis részben az iskolai munkát meghatározó pedagógusokon múlik.

Marx György már a '60-as évek végén „gyorsuló idő”-ről beszélt.

A személyi számítógép, internet, mobiltelefon, DVD ekkoriban még szinte csak az álmok kategóriájába tartozott. Harminc év elteltével azonban ezek az eszközök mindennapos használati cikké, sok esetben nélkülözhetetlen munkafeltétellé váltak. Ez alatt az idő alatt azonban kedvezőtlen változások mentek végbe magában a pedagógus társadalomban:

#### **Említsünk meg néhányat ezek közül:**

1. a folyamatos és valószínűleg megállíthatatlan presztízsvesztés, ami összefüggésben lehet a pedagógus fizetések alacsony szintjével;
2. az iskolában oktató tanárok átlagéletkorának emelkedése, ami a folyamatos megújulás gátjává válik.

### **E-generáció (net-generációnak)**

A civilizáció történetében először fordul elő, hogy a gyermekek tanítják a felnőtteket. A gyerekek ügyesebbek a számítógép kezelésében. A szülők, a tanárok és más felnőttek a gyerekekhez fordulnak információért, segítségért és más, számítógépes dolgokkal kapcsolatban.

Ezek a gyerekek beleszülettek az új technológia világába, környezetük természetes részeként veszik birtokukba, s helyezik mindennapi tevékenységük és kultúrájuk középpontjába, szüleik azonban általában nehezebben sajátítják el azt.

A digitális korszakban az interaktivitás a jellemző a korábbi passzív szereppel szemben. A gyerekek saját maguknak testre szabhatják saját tudásu-



kat, mi akkor vagyunk jó pedagógusok, ha ebben nem megakadályozzuk, hanem segítjük - persze ennek vannak veszélyei is.

A mai felnövő diáksereg jártasabb az Interneten mint a mai számítástechnika tanár.

Magyarázat: egyelőre több ideje van (igaz az egyéni tanulás rovására) mint a tanárnak és fogékonyabb azokra a mozdulatokra és klikkelésekre amely nélkülözhetetlen egy böngésző programnál.

Milyen szolgáltatást vesz igénybe a diák az Interneten?

- levelezés
- ingyen SMS küldése
- chat
- magas színvonalú termékek megtekintése (autó, számítógép)
- játékok és filmek letöltése
- vírus ellenőrző programok letöltése

A békéscsabai INFOERA 2002 találkozóról kiemelném Csepeli György előadását amelyet egyszerűen így nevezete a mai diákokat: "eGeneráció".

**A tanár-diák viszonyát** kooperáció és testreszabottság kell, hogy jellemezze. Az interneten szerzett tudás hypertextes világában értelmetlen tananyagokról beszélni, a diszciplínák elkülönítése értelmetlenné válik.

A fény mellett ott az árnyék, a hatalmas lehetőség és a démonok megjelenése. Ez az eszköz olyan mint a szike, pusztításra és csodálatos gyógyításra is alkalmas.

Nagy a tanárok felelőssége, ahelyett, hogy őket (a diákokat) öregítsék, magunknak kell megfiatalodni - mondta zárszavában az államtitkár.

Az e-generáció diákjai nem olvasnak, egy irodalmi mű elemzését vagy tartalmát letöltik az Internetről. Nagyon gyakori hogy különböző órákon számon kért dolgozatok írása egy bizonyos témában egyetlen vagy esetleg két weboldal összefűtéséből jön létre. Saját véleményüket nem írják be a fogalmazásban. A diákokat mindig a legújabb szolgáltatás érdekli az Interneté. Egy-két évvel ezelőtt az ingyenes SMS volt a téma. Gyakran kérdeztek olyan web címeteket ahonnan ingyen SMS küldhető.

**Milyen eszközök jelentek meg a levelezés folytán?**

Kiterjedt, népszerű, igen változatos, rendkívül ötletes az ún. „smiley”-k serege (smiley, tsz. smilies = mosolygó, mosolygós, vigyori). Ezek a betűkből, számokból, matematikai műveleti jelekből és írásjelekből „összeeszkábált” arcfélék

elsősorban érzelmi állapotot, hangulatot fejeznek ki, de utalhatnak egy személy külsejére, szokásaira vagy más jellegzetességeire is. A csetes beszélgetés közben simán, könnyedén beüthetők. ( A képek 90 fokkal elfordítva nézendők !):

:-) = mosoly,	:~)) = nevetés,	:~)))) = hahotázás,	:- ( = sírás,
8-) = szemüveges,	(:-) = kopasz,	:- { } = bajszos,	{:-) = parókás,
:-) = punk,	d:-) = baseball sapkás,	<-) = kínai.	

A smileykkal cselekvést, tevékenységet is jelezhetünk:

:-* = csókol,	;-) = kacsint,	:-x = hallgat,	:-q = cigarettázik,
l-O = ásít,	:-V = kiabál,	>:0 = dühöng,	
:-D = teli szájjal nevet,		:-? = pipázik.	

Ezek a jelek kitalálói ötletességét, kreativitását dicsérik. Használatuk kedvességét, vidámságot visz a csevegésbe. Jelentésük „megfejtése” élvezetes szellemi fejtörő a résztvevők számára. A már meglévő smileyk tárháza is igen nagy, és naponta újabbakat és újabbakat szül az emberi fantázia. Persze az átlag-csetek során csak néhányat használnak rendszeresen a beszélgetők, leggyakrabban a mosoly- és nevetésjelző ábrákat – mutatván, hogy az előtük leírt szöveg csak vicc, nem kell nagyon komolyan venni.

Szinte természetes, hogy alkalmazzák a közismert rövidítéseket is: vki (valaki), vmi (valami), kvtár (könyvtár), aztán a szóbeli szlengben elterjedt rövidített szóalakokat is: pasi, cigi, tuti, furi (furcsa), jogsi, flekó, fizu, uncsi, gatyó, mini, vaccsi, uzsi, skizó (skizofrén), tom (tudom), spéci. Ha ezeknek van – a rövidítésen belül – hosszabb és rövidebb változata is, írásban szinte mindig a rövidebbel találkozunk: caj-bicaj, szit-szitu, csá-csáo.

A helyesírással is elég önkényesen bánik a **net-generáció**. Részint gyakran követnek el valós hibákat, tévesztéseket, betűkihagyásokat – főként a gyorsaság diktálta kapkodásból, figyelmetlenségből –, részint pedig sokat és felszabadultan játszanak a szándékosan, viccből eltorzított „hejesírás” poén-lehetőségeivel. Leggyakoribb játéku: a szóelemzés elvének semmibevételével mindent kiejtés szerint írni. Pl.: Jöhecc jáccani.; Kéccer vettem: eccer észre, eccer el.; Még eszt se tuggya!; Tekincsük át a weblapot!; Majnem mekhaltam.

A hálózaton a legkülönfélébb virtuális közösségeket alakítják ki a gyerekek. Társalgásuk szinkron vagy aszinkron. Az előbbi beszélgető szobákban, az utóbbi fórumokon, hirdetőtáblákon zajlik – moderátor (szülő, tanár, szociális munkás) megfigyelésével vagy a nélkül – szinte teljesen szabadon, de a netikett szabályainak szigorú betartásával. Bárki bármikor beléphet vagy távozhat, közzéteheti a véleményét, kezdeményezhet újabb témát. A FreeZone például információcserét, csevegőszobákat és különböző valós idejű programokat kínál a gyerekeknek. Közben baráti kapcsolatok szövődnek, gyakran egész kibercsaládok jönnek létre, amelyek éppúgy megjeleni-



tik a csoportképződés és -kapcsolatok sajátosságait, mint a valódi életben. Mégpedig az írott szóra, a szövegre alapozva, mintegy rehabilitálva a verbális nyelv varázsát. Metanyelvi funkcióban szinte új írásrendszer keletkezett, rövidítésekkel, betűszókkal, hangulatjelekkel (smile) stb.

Magukra lehetnek, megválaszthatják a csoportot, személyeket, akikkel együtt akarnak lenni a virtuális térben. Ezek a virtuális kapcsolatok egy gombbal megsemmisíthetők. Erős tendencia van a játékos identitásra, hipertrofizált én (istenné válás, kiszakadás a mindennapi létből), kábítás amelyre, ha ráérez néhez kiszakadni belőle. Ezekből a diákokból lesznek az Internet rabjai.

Az új generáció gondolkodására nagy hatást gyakorol(hat) az úgynevezett hipertext, amelyet még a 60-as években Douglas Englebart fejlesztett ki, aki az egeret is feltalálta. A hipertext ugyanis az információfeldolgozás soros, lineáris formáit párhuzamos, nem szekvenciális műveletek folyamatává alakítja, s ezáltal komplex kognitív struktúrák kialakítását teszi lehetővé. Englebart zseniális újítása a World Wide Web előfutárának tekinthető. Sokan vallják, hogy ez a technika alapvetően megváltoztatta a gondolkodásmódjunkt.

### Erdélyi körséta

2002 végén új hírportál jelent meg az erdélyi piacon. [www.hirek.ro](http://www.hirek.ro). Barátok segítségével tervezték meg az oldalt, tudósítókat alkalmaztak, megszerezték a szükséges engedélyeket, és 2002 szeptember 15-én elindították az új hírportált. „Eredetileg azt a célt tűzték ki maguk elé, hogy kizárólag saját anyagokat közöljenek, amelyek Erdély eseményeiről szólnak. Az oldal beindulása óta sikerült jó kapcsolatot kiépíteni néhány erdélyi magyar rádióval, újsággal, így a szerkesztők esetenként a hírcsere lehetőségével is élnek. Infrastruktúrájuk szegényes ugyan, de: a híreket a tudósítók akár egy netklubból is elküldhetik, az oldalt pedig bárholnan lehet frissíteni. Itt működik az általuk létrehozott, interneten hallgatható Erdély Rádió stúdiója.

Látogatottsági statisztikája azt mutatja, hogy egyre nagyobb iránta az érdeklődés. Decemberben közel százezer helyről néztek be a portálra. „Érdekes, a sajtó részéről Magyarországról nagyobb az érdeklődés, mint itthonról.” – tűnődik el a tulajdonos, aki rendszeresen követi a látogatottsági mutatót. A hírovtat után a legnépszerűbb a vitafórum, ezt követik az apróhirdetések és a társkereső. Az oldalon SMS-küldő programot is elhelyeztek, amelynek működése egy másik szervertől függ, így egyelőre nem biztosított, hogy a telefonra küldött szöveges üzenet meg is érkezik.

Erdély legnagyobb hírportálja, a Transindex 1999-ben indult, és másfél évig hárman szerkesztették egy tömbházlakás nappalijában. Az indulás utáni időszakban napi 2-3 látogatót tartottak nyilván, akkor az internet elterjedési mutatója Romániában még 0,8 százalék körül volt.



A Transindexnek, „a napos oldalnak” publikus a látogatottsági statisztikája, naponta 2100-2300-an olvassák. Emellett fontos a hivatkozottság-indexük is: 5-6 napilap és ugyanennyi rádió folyamatosan felhasználja hír-blokkjában az internetes újság híreit. Népszerű vitafóruma, a Disputa már a napi 500 látogató felé közeleg, szűk egy év alatt megduplázódott a forgalma

### **Van-e Erdélyben befogadó közönség?**

Cs. Zs. szociológus megjelent felméréséből kiderül, a belső-erdélyi és partiumi fiatalok közel egyharmada, ezen belül a számítógép-használók hatvan százaléka használ internetet. E fiatalok, ahogy a néhány megkérdezett idősebb internetező közül is kevesen követik az on-line médiát.

G. A. rendszergazda, az egyik kolozsvári internetklub felvigyázója úgy véli, klienseinek mintegy nyolcvan százaléka csak levelezésre, illetve szórakozásra használja az internetet. Néhányan, főleg egyetemisták, szintén a világháló használják dokumentációs forrásnak egy-egy dolgot elkészítéséhez, és csak elenyésző azok száma, akik újságok, tévék honlapjain, portálokon böngésznek. Az elektronikus médiában dolgozó F. S. szerint az erdélyi híroldalak nem találnak megfelelő fogadtatásra, mivel az olvasóközönség még nem ismerte fel az on-line média jelentőségét: „Amíg a megyei lapok ugyanúgy irnak, mint 13 évvel ezelőtt, az olvasóközönség pedig ezt el is fogadja, addig a portáloknak nem lesz közönsége.”

Az erdélyi ifjúsági szervezetek folyamatosan pályáznak számítógépekre, a multimédiás infrastruktúra fejlesztésére, teleházak létrehozására..

### **Veszélyes internet?**

Egy orosz pedagógus szerint a gyermekek többsége az Internet rabságában él. Leonyid Klein a számítógép-függőséggel magyarázza a gyermeki agresszió felerősödését, valamint a gyermekek által elkövetett gyilkosságok és erőszakos bűncselekmények számának növekedését. A moszkvai Európai Gimnázium tanára cáfolja azt, hogy a Harry Potter könyvek megjelenése óta az iskolások többet olvasnának. Többségük, egy főlmérés tanúsága szerint nem is olvasta magukat a könyveket, csupán a Harry Potter féle számítógépes játékokat ismeri.

### **A tanulási eredmények mérése.**

Ilyen és ehhez hasonló feltétek mellett a számítástechnika tanár kell értékelje a tanulók ismereteit és osztályozzon vagyis jegyet adjon.

Az információgyűjtést gyakran az ellenőrzés fogalmával jelölik, s első részmozzanatként kezelik a különböző oktatásmódszerekben. Az iskolákban gyakori a számonkérés kifejezés használata. Az értékelés szintje, funkciója, tárgya befolyásolja, hogy milyen módszerekkel lehet élni az információgyűjtés során. Alkalmazni lehet a megfigyelést, a szóbeli, az írásos kikérdezést, a különböző, teljesítményt feltáró módszereket. A tanulók tudásának ellenőrzésekor használhatjuk a szóbeli feleleteket, beszámolókat, röpdolgozatokat. Az ellenőrzés leggyakoribb színhelye az számítógépterem, a tanítási óra, de történhet különböző vizsgahelyzetekben, tanulmányi versenyeken, stb.

A tanulási eredmények mérésének elvi és gyakorlati kérdései: a minőségi és mennyiségi értékelés közötti átmenetet képviseli a megítélés. Ebben az esetben a viselkedéseket, a teljesítményeket két csoportba soroljuk, elfogadhatóak vagy elfogadhatatlanok. A mennyiségi értelmezés előtt becslést végzünk. Mérés esetén sokkal pontosabb az értékelés. A tudásszint mérő tesztek jellemzői, a tesztkészítés technikája. Vizsga, vizsga-rendszer: A tesztekben feladathelyzetbe hozzuk a tanulót és e feladatok megoldása alapján értékeljük a megvizsgálandó tulajdonságot. Mércét állapítunk meg, amelyhez aztán viszonyítani lehet a tanuló teljesítményét. A tesztek sokféle típusa jött létre, vannak tudásmérő tesztek, intelligenciát, kognitív kompetenciát, különböző képességeket felmérő tesztek.

És végül hogy mindez miért történik

*„...ha jobb az iskola, jobb a társadalom.”*

(Ferge Zsuzsa)

### **Források**

- [1.] Fehér Péter, Milyenek az internet-korszak pedagógusai?, [www.sulinet.hu](http://www.sulinet.hu)
- [2.] Vass László, Az internet generáció felemelkedése
- [3.] Jóna Dávid, Elmélkedés a mediapedagógiáról
- [4.] Fehér Péter, Milyen legyen egy Internet-pedagógus?  
Internet, Az informatika oktatásmódszertanának fogalma, tárgya, feladatai, kapcsolatrendszere
- [5.] Tasnádiné Rónaky Edit, Az „internet-szlang”
- [6.] Csepeli György, eGeneráció az INFO ÉRA 2002 konferencián elhangzott előadása.

# Mesterséges intelligencia társasjátékokban

## Artificial Intelligence in Board-games

Máthé Zsolt<sup>1</sup>, Görög Levente<sup>2</sup>, Komáromi Lóránd,  
Szilágyi Sándor Miklós<sup>2</sup>

<sup>1</sup>Kolozsvári Műszaki Egyetem, <sup>2</sup>Sapientia EMTE

### Abstract

*A specific method for developing intelligent decision-making agents by using different combinations of AI (Artificial Intelligence) algorithms is presented in the current work. These agents are designed to play in most logical games for two, namely a small change in their structure makes them suitable to adapt to different circumstances and rules. They have been tested with one of the famous board game of Ravensburger. The obtained results are promising and encouraging in order to start with future work regarding the new theories phrased in the time. The present study goes with explaining how to prone optimally a search tree, how to validate this procedure, and how to construct a general genetic algorithm.*

### Bevezetés

Alaposabb szakismereteket igénylő, jól elszigetelt, és algoritmikus megoldásokat (egyelőre) nem ismerő feladatok számítógépes implementációja immár követelményként jelentkezik – egyre sürgetőbb módon – a XXI. századi számítástechnikában. Ahogyan az automatizálásra váró problémák komplexitása növekszik, úgy válik egyre nyilvánvalóbbá a tény, hogy a klasszikus értelemben vett racionalizáló algoritmusok nem alkalmasak az ilyen jellegű feladatok pszeudo-leírására, illetve megoldására. A jól specifikált problémákra megírt egyszerű reflexszerű ágensok, akár csak az ezeket támogató explicit keresőtáblák funkcionális használata egy lassú erodálási folyamat során jut majd el a morális kiöregedéshez [7]. Egyáltalán – vetődik fel a kérdés – lehet-e egy megfelelően bonyolult feladatot a hagyományos értelemben jól specifikálni, például olyan feladatokat, amelyekkel az intelligens entitások nap mint nap szembesülnek? Kérdéses, hogy megérthetjük-e, milyen mechanizmusok alapján érzékelünk, tanulunk, emlékezünk és következtetünk, illetve, amennyiben igen, meg tudjuk-e valósítani ennek a megértésnek egy adekvát leírását és ezáltal valamiféle automata rendszerbe történő integrálását: „...annak tanulmányozása, hogy hogyan lehet a számítógéppel olyan dolgokat művelni, amiben pillanatnyilag az emberek a job-



bak” (Rich és Knight, 1991). Dolgozatunkban ezekre a kérdésekre szeretnénk (lehetséges) válaszokat adni, illetve további kérdéseket megfogalmazni, mindezt egy konkrét, még fejlesztésben lévő alkalmazásunk tükrében.

A fentiekre reflektálva nyilvánvaló, hogy olyan módszereket keresünk, amelyek plauzibilis távlatokat nyitnak meg intelligens viselkedés modellezésére. Először azonban megemlítjük a már létező ilyen jellegű, validált eljárásokat:

1. **Evolúciós algoritmusok (EA):** a genetika és természetes szelekció fejlődéstani vetületére alapozott adaptív algoritmusok [2].
2. **Neurális hálózatok (NN):** Az agy információ-feldolgozási mechanizmusának analógiájára létrehozott rendszerek, amely felépítésükben gráfra hasonlítanak, neuronokat csomópontokkal, míg kapcsolatokat élekkel reprezentálva [5].

A fent bemutatott módszerek nem tekinthetők homogén eljárásoknak. Reprézenciától függően lényeges eltérés mutatkozhat implementálásukban, olyannyira, hogy csupán az alapelv marad közös. A módszerek *gyűjtőneve* a **mesterséges intelligencia (AI)**, azaz összefoglalásként:

1. A nem létező algoritmus helyett a probléma megoldását heurisztikus kereséssel\* támogatjuk
2. Létrehozuk a heurisztikus keresésnek megfelelő rendszerarchitektúrát (intelligens ágensek)
3. Gyors, empirikus prototípust készítünk – ezzel kísérletezve javítjuk a feladat specifikációját

## Koncept

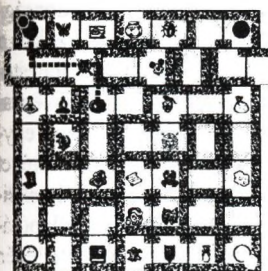
A tanulmány alapjául szolgáló labirintus-probléma megoldásában a fent bemutatott MI módszerek, valamint az általános játékelméletek egy optimális kombinációját célozzuk.

A labirintus-problémát a Ravensburger cég *Das verrückte Labyrinth* nevű társasjátéka szolgáltatta. A játéktábla egy 7x7-es mátrixként fogható fel, melynek elemei háromféle útszakaszt tartalmazhatnak (cső, elosztó és sarok), valamint egyes elemeken képrajz is található (1. ábra).

A képrajzok számával (24) megegyező számú kártya kerül a játékosok (max. 4) között kiosztásra.

---

\* A heurisztikus keresés lényege az exponenciális bonyolultságú és kimerítéses keresés ágainak megnyesése a lényegtelen keresési irányok figyelmen kívül való hagyásával, az exponenciális bonyolultság polinomiálisra való visszavezetésére.



1. ábra

A kártyákon a képrajzokkal meg-  
 egyező figurák vannak. Kezdetben a  
 játékosokat reprezentáló különböző  
 színű bábuk a tábla sarokpozícióiban  
 helyezkednek el, innen próbál ki-ki  
 eljutni ahhoz a táblán lévő figurához,  
 amelyet kártyapaklijának legfelső kár-  
 tyája jelez. Amint ez sikerült, a játékos  
 megmutatja kártyáját a többieknek,  
 félreteszi, majd veszi a következő kár-  
 tyát, és ismétli ugyanezt.

Az a nyertes, akinek leghamarabb elfognak a kártyái. Mozogni (lépni)  
 csak út mentén szabad, új utat pedig tolással (sor, oszlop) lehet létrehozni. A  
 mátrix páros sorai és oszlopai tologathatóak mindkét irányba (fel-le, jobbra-  
 balra) az addicionális tolóelemmel, amelyet tetszőlegesen pozícióban (forgatás)  
 lehet beilleszteni. A tolás során megváltozik a tábla struktúrája és ezáltal a  
 lehetséges útvonalak is. Az 1. ábrán látható állás: a bal felső sarokban lévő  
 játékos a második sor balról jobbra történő eltolásával utat alkot kezdőpozíci-  
 ójától a (3,3) pozícióban lévő képrajzig. A játékos rendelkezésére álló (2,  
 0→1) tolóelem – akárcsak a tolás után (2, 7→8) kieső elem – sarok.

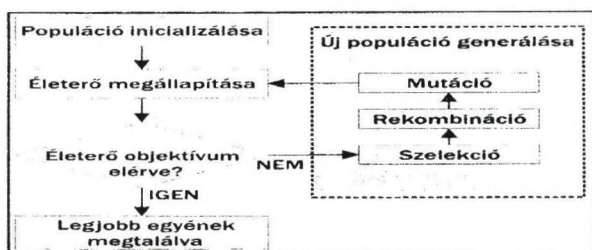
### Módszerek és matematikai leírás

A játék elágazási tényezője  $\sim 38.8235^*$ . Durva közelítéssel tekinthetjük  
 39-nek – ami pontosan 4-gyel nagyobb a sakk elágazási tényezőjénél (35).  
 Ezzel az elágazási tényezővel az 5-ös mélység 16-os nagyságrendű ( $7.7 * 10^4$   
 ( $10^4 \cdot 15$ )), a 10-es mélység viszont ennek a nagyságrendnek mintegy két-  
 szerese ( $6 * (10^4 \cdot 31)$ ). Nyilvánvaló, hogy exponenciális komplexitás eseté-  
 ben a teljes keresési fa felépítése túlságosan idő- és memóriaigényes [3].  
 Noha a játék állását könnyű reprezentálni és a mindenkori állások állapotai  
 hozzáférhetőek a program számára, megjelenhetnek véletlen, diszkrét ele-  
 mek ebben a tudásreprezentációban: a kártyák véletlenszerű kiosztása, az  
 időkorlát és ebből következően a mélységkorlát, valamint az ellenfél jelen-  
 léte. A véletlen elemeket tartalmazó tudásreprezentáció, az idő- illetve  
 mélységkorlát kizárja a tökéletes döntést, vagyis a keresési fán alfabéta nye-  
 sést [7] kell végrehajtani. Az alfabéta nyesést az adott szint csomópontjai-  
 nak hasznosságértékei szerint végezzük. Egy csomópont hasznosságát a

\* 13 cső, 15 sarok és 6 elosztó, ezek elhelyezési pozíciója szerinti átlag:  $(13 \cdot 2 + 15 \cdot 4 + 6 \cdot 4) / (13 + 15 + 6) \sim 3.2352$ . Ezt szorozva a toható sorok és oszlopok számával (12), megkapjuk a játék átlagos elágazási tényezőjét, amely  $\sim 38.8235$ .



következésképpen határozzuk meg: kezdetben definiáltunk egy tulajdonság-szettet, amely – első megközelítésben – kielégítően jellemezte azokat a szempontokat, amelyekre egy lépés során figyelni kellett (a játékos lépés-terének nagysága, a kitolt elem milyensége, a cél és az aktuális pozíció lépés-terei közti minimális távolság stb.). A szett mindenik tulajdonságához egy-egy függvényt rendeltünk, amelyek az adott táblaállapotokban számértéket feleltettek meg a megfelelő tulajdonságoknak. Így a tulajdonságok értékei- nek összegeként mérhetővé vált a hasznosság egy adott szint adott állapotá- ban (csomópontban). A tulajdonságokat azonban súlyozni is kellett, lévén ez teljesen empirikus, egy genetikus módszert alkalmaztunk, amelynek fo- lyamatábrája a lenti ábrán található.



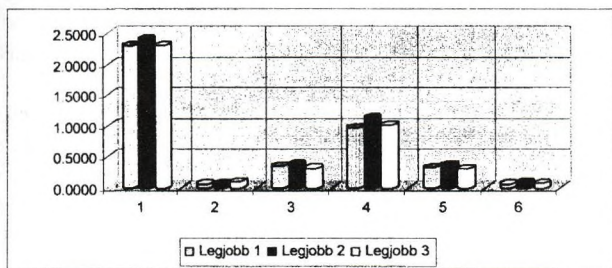
2. ábra

Generáltunk egy 100 tagú populációt, melynek tagjai a megfelelő tulajdonságok súlyait (gének) véletlenszerűen vették fel általunk definiált hal- mazokból (inicializálás). A populáció minden tagját elláttuk egy ugyanak- ora energianívóval, majd elkezdjük véletlenszerűen játszani őket. A nyer- tesek energianívóját előnyük függvényében növeltük, a veszteséket ugyan- így csökkentettük, mígnem a természetes kiválasztódás során egyesek el- pusztultak, mások dominánssá váltak egy  $x$  állapotban. Megvizsgáltuk, hogy teljesült-e az életerő kritérium a legdominánsabb egyedek esetében, amennyiben nem, az elpusztult egyedek helyett újakat hoztunk létre a fennmaradt dominánsakból mutáció (+5 %-os változtatás a génekben) és rekombináció (két, véletlenszerűen kiválasztott egyed néhány, ugyancsak véletlenszerűen kiválasztott tulajdonságainak kicserélése). Amint teljesült az életerő-kritérium (elérkeztünk az  $xx$  végállapotba), megkerestük a leg- dominánsabb egyedeket, és ezeknek a génjeit feleltettük meg az optimális tulajdonságsúlyoknak – melyeket a továbbiakban konstansként kezeltünk (az 1. grafikonon a három leoptimalisabban teljesítő egyed tényezőinek súlyai láthatóak). Hasonlóan jártunk el a keresési fa mélységkorlátjának meghatározásakor az idő-hatékonyság tulajdonságok súlyozása esetében is.



A genetikus algoritmussal nyert mindkét eredményt (mélységkorlát, tulajdonságok súlya) felhasználjuk a program nehézségi szintjeinek állításában. Mindezekből látható, hogy az általunk használt genetikus algoritmus nem feladat-specifikus.

## Elemzések



Legjobb 3 egyed	T1	T2	T3	T4	T5	T6
1	2.30	0.061	0.35	0.97	0.33	0.065
2	2.40	0.074	0.38	1.13	0.38	0.086
3	2.30	0.099	0.32	1.01	0.31	0.072

1. grafikon

Az 1. grafikonon az első 3 legsikeresebb egyed tényezőink súly-értékei láthatóak, 6 paraméter, melyek közül az első esik a legnagyobb súllyal, a második nagysága fordítottan arányos a hatékonysággal, a többiek közötti összefüggés pedig határozatlan.

## A keresési fa megnyesése

A játék következő  $n$  állapotát egy keresési fában szeretnénk tárolni, azonban, hacsak az  $n$  nem egy végzetesen kis érték, nincsen erre lehetőség: a játék elágazási tényezője 39, és minden egyes levélcsomópontban útkereséssel szembesülünk. A legrövidebb út-kereső algoritmus külön fontosságot rejteget a kiértékelő függvényben. Mivel ez a leggyakrabban használt módszer az új irányok keresésére és az állapotok heurisztikus kiértékelésére, hangsúlyt kellett fektetnünk arra, hogy komplexitása a lehető legkisebb legyen, és a lehető legkevesebb, de ugyanakkor a leggyorsabban [1] elérhető memóriát használja. A legrövidebb út-keresés implementálható a legegyszerűbb mélységi bejárásos próbál-

gátásokkal (backtracking vagy rekurzió), mi azonban a **Depth-First**-tel szemben a **Breath-First** bejárást preferáltuk (az előbbi exponenciális komplexitása végett), illetve egészen pontosan az ezen az eljárásan alapuló **Lee** algoritmust [6], amely a szakirodalomban még **Fill** néven is ismert. **Breath-First**: próbáljuk meg a vektorelemű vektort (azaz mátrixot) úgy elképzelni, mint egy irányítatlan gráfot, ahol akkor van el két (csomó)pont között, ha azok szomszédok. Ezt az algoritmust alkalmazva tulajdonképpen egy szabályozott, feltételes hullámlázmást [4] végzünk a vektortömbön, melynek eredménye éppen a legrövidebb út a kezdőpont és a többi pontok között. Így nyilvánvalóan javítunk az alprogram futási idején, lévén ennek az eljárásnak a komplexitása  $O(m+n)$ , ahol  $m$  az élek száma, illetve  $n$  a csomópontok száma.

A keresési fán alfábéta nyesést [7] kell végeznünk az állapottér szűkítése végett. Ezt a hagyományos Greedy módszerrel valósítjuk meg\*, de lévén ez mohó algoritmus, heurisztikus keresés esetében kérdéses a hatékonysága, azaz validálni kellett (1. táblázat). A nyesés validációját 3-as és 4-es mélységben végeztük: a programot önmaga ellen játsztattuk teljes, illetve megnyesett keresési részfával.

Mélység	Továbbvitt állapotok	Optimális nyerési aránya
3	10%	82%
	1%	92%
	1‰	99%
4	10%	69%
	1%	77%
	1‰	85%

1. táblázat

A táblázat első oszlopa a bejárt mélységet tartalmazza, a másodikban azt tároljuk, hogy a nyesés után hány százaléka (ezreléke) maradt meg a teljes keresési fának, míg az utolsó oszlop a teljes keresési részfával dolgozó algoritmusnak a megnyesett keresési részfával operálóhoz viszonyított nyerési arányát fejezi ki szintén százalékosan. 3-as mélységben a táblázat minden egyes sorában szereplő szám adatok 100 játék átlagaként értelmezendők, míg 4-es mélységben 10 játék után számítottunk átlagokat. Azon hipotézisünket, hogy nyesési módszerünk hatékony, igazolni látszanak ezek az adatok, hiszen 4-es mélységben a keresési részfa csupán 10%-át megtartva a program 31%-osan teljesített a megnyesetlen keresési részfával operáló gépi ellenfele ellen. A keresés mély-

\* Adott szint (mélység) levélcsofópontjain azok hasznossága szerint csökkenő sorrendben megyünk végig, majd egy bizonyos értéknél kisebb hasznosságú csomópontok alatti ágakat lenyessük. A levélcsofópontok hasznosságát a [3]-ban bemutatott módszerrel állapítjuk meg.



ségkorlátjának megállapítását (3 illetve 4) nem explicit módon végeztük, hanem szintén genetikus algoritmus segítségével. Adott mélységig keresve, mérhetjük a keresés relatív hatékonyságát, illetve mérhetjük az adott mélységhez szükséges keresési időt. A hatékonyság ( $E$ ) és az idő ( $t$ ) közös nevezőre hozásával (a két tényező súlyozása genetikus algoritmus segítségével) mérhetővé válik a hasznosság ( $U$ ). Ezután csupán az  $f(E, t) = U$  függvény maximumát kell megállapítani. A genetikus algoritmus megállási fitness-kritériuma (életerő objektívum) a konvergáló sorok mintájára a következőképpen van definiálva:  $|X_n - X_{n+1}| < \epsilon$ . Ez annyit tesz, hogy a populáció két legdominánsabb tagjának fitness-mutatója megközelíti egymást (különbségük abszolút értéke kisebb, mint az  $\epsilon$  hibakorlát), vagyis nincsen már lehetőség számottevő fejlődésre.

### Következtetések, célok

Az AI módszerek egy lehetséges jövőképet vetítenek elő a számítástechnikában. Megfelelően nagy állapotér esetében a lehető leghatékonyabbak az ismert algoritmusok közül-, a kisebb terek esetében ez még függőben lévő kérdés. Céljaink közé tartozik az ezzel kapcsolatban felmerülő problémáink során megfogalmazott hipotéziseink tesztelése, egy feladat-specifikációtól független, általános genetikus algoritmus megfogalmazása, valamint pszeudo-nyelven vezérelt interkommunikációs ágensek működtetése a megoldandó feladat specifikálására.

### Hivatkozások

- [1.] Todd M. Austin, Dionisios Pnevmatikatos, Gurindar Sohi: **Streamlining data cache access with fast address calculation**, May 1995, ACM SIGARCH Computer Architecture News, Proceedings of the 22nd annual international symposium on Computer architecture, Volume 23, Issue 2
- [2.] Thomas Bäck: **Evolutionary Algorithms in Theory and Practice**, New York – Oxford, Oxford University Press, 1996
- [3.] Andrew C.-C. Yao, **On the Complexity of Comparison Problems Using Linear Functions**, Proc. 16th Annual Symp. on Foundations of Computer Science, Berkeley, Calif. 1975, 85-89.
- [4.] Xiong Ji-Guang: **An Algorithm for Path Connections Using Asynchronous Diffracted Waves**, JACM(2004)
- [5.] Simon Haykin: **Neural Networks. A Comprehensive Foundation**, Prentice Hall PTR, Upper Saddle River, NJ, 1994
- [6.] C.Y.Lee: **An algorithm for path connections and its applications**, IRE Trans. EC-10, P. 346 (Sept. 1961).
- [7.] Stuard J. Russell and Peter Norvig: **Mesterséges Intelligencia modern megközelítésben**. Panem – Prentice Hall, Inc. 2000



# Optimális vezérlési stratégia megvalósítása mesterséges neuronhálók segítségével

Comparison Between LTI LQ Optimal Controller with Hierarchical Numerical Solution vs ANN Solution

Dr. Dávid László

## Abstract

*In recent year, enormous progress has been made on the application of the neural networks as neuro- adaptive controllers. Tracking control described in this paper's was solved by a discrete time digital repetitive controller as a quadratic programming problem with inequality constraints and parameter estimation. Numerical examples are presented on standard Riccati equation solution vs. real time neural solution by Hopfield network.*

## Bevezetés

Századunk derekán világszerte több kutatócsoport kezdett interdiszciplináris tudományos témakörrel foglalkozni. Ez teremtette meg a feltételeit egy új tudományág a mesterséges intelligenciák megjelenésének és fejlődésének is. Ma e kutatások több ágazatra szakadtak és csakis egy új módszertani szemlélet bevezetése jelent áttörést, aminek az egyik lényeges jellemvonása az, hogy az addig általánosan elfogadott soros feldolgozást, egy sajátos párhuzamos feldolgozásra alapozó mesterséges struktúra váltotta fel, amely azt eredményezte, hogy a számítógépes programok több évtizede egyetemesen elfogadott Neumann elvét egy párhuzamos sajátos algoritmussal helyettesítjük, melyet az agyi idegsejtek mintájára mesterséges neuronhálónak neveztek el. Ebben a témakörben az általunk megvizsgált módszer arra próbál választ kapni, hogy miként lehet az időigényes optimális vezérlési stratégiát neuronhálók segítségével megoldani ezzel párhuzamosítva az algoritmust.

## A feladat megfogalmazása

Legyen a következő sztochasztikus diszkrét időben változó dinamikus rendszer.

$$\begin{aligned} \underline{x}_{k+1} &= \Phi(t_{k+1}, t_k) \underline{x}_k + \Gamma(t_{k+1}, t_k) \underline{u}_k + \underline{w}_k = \Phi_k \cdot \underline{x}_k + \Gamma_k \cdot \underline{u}_k + \underline{w}_k \\ \underline{y}_k &= \mathbf{H}(t_{k+1}, t_k) \cdot \underline{x}_k + \underline{r}_k = \mathbf{H}_k \cdot \underline{x}_k + \underline{r}_k \end{aligned} \quad (1)$$

ahol,  $\underline{z}_k, k = \overline{1, N}$  az előírt  $t_k$  mintavételekben N lépésben előírt pálya. Tételezzük fel, hogy keressük azt a vezérlési stratégiát amely biztosítja a következő célfüggvény minimumát.

$$\min_{\underline{u} \in U} \{J(\underline{u})\} = \min_{\underline{u} \in U} \left[ \underline{e}_N^T \cdot F \cdot \underline{e}_N + \sum_{k=0}^{N-1} \underline{e}_k^T \cdot Q_k \cdot \underline{e}_k + \delta \underline{u}_k^T \cdot R_k \cdot \delta \underline{u}_k \right], \quad (2)$$

ahol  $Q_k = \frac{1}{k^2} \cdot I_n$ ,  $F = \frac{1}{N^2} \cdot I_n$  és  $R_k = r$  A feladathoz rendelt diszkrét

Riccatti egyenlet a következő:

$$P_k = \mathbf{H}_k^T \cdot Q_k \cdot \mathbf{H}_k - \Phi_k^T \cdot P_{k+1} \cdot \Gamma_k \cdot [\mathbf{R}_k + \Gamma_k^T \cdot P_{k+1} \cdot \Gamma_k]^{-1} \cdot \Gamma_k^T \cdot P_{k+1} \cdot \Phi_k + \Phi_k^T \cdot P_{k+1} \cdot \Phi_k \quad (3)$$

$$q_k^T = -2 \cdot \underline{z}_k^T \cdot Q_k \cdot \mathbf{H}_k + 2 \cdot \underline{u}_k^T \cdot R_k \cdot K_k + q_{k+1}^T \cdot [\Phi_k - \Gamma_k \cdot K_k] \quad (4)$$

ahol,  $K_k = (\mathbf{R}_k + \Gamma_k^T \cdot P_{k+1} \cdot \Gamma_k)^{-1} \cdot \Gamma_k^T \cdot P_{k+1} \cdot \Phi_k$  és a megfelelő optimális vezérlést megkaphatjuk mint:

$$\underline{u}_k^* = -K_k \cdot \underline{x}_k + (\mathbf{R}_k + \Gamma_k^T \cdot P_{k+1} \cdot \Gamma_k)^{-1} \left[ \mathbf{R}_k \underline{u}_k + \frac{1}{2} \Gamma_k^T \cdot q_{k+1} \right] \quad (5)$$

ahol az N időpontban meghatározott végfeltétel akövetkező:

$$P_N = \mathbf{H}_N^T \cdot F \cdot \mathbf{H}_N, \quad q_N^T = -2 \cdot \underline{z}_N^T \cdot F \cdot \mathbf{H}_N \quad (6)$$

Az általánosan elfogadott módszer e feladat megoldására az 1 ábrán bemutatott ún. „ismételt vezérlés”, ahol a z előírt pályapontokat úgy határozzuk meg, hogy a legelső pont az aktuális mért y kimenetnek a pályára eső vetülete legyen.



1. ábra  
Az előírt pálya és az aktuális kimenet

Természetesen a véges  $\delta$  számítási idő feltételezi a becült  $\underline{x}_\delta$  állapot kiszámítását ahonnan a következő képlet alapján kaphatjuk az  $\underline{u}_\delta^*$  vezérlést:

$$\underline{u}_\delta^* = \mathbf{R}_k^{-1} \cdot \mathbf{\Gamma}_k^T \cdot (\mathbf{P}_1 \cdot \underline{x}_{2\delta} + \underline{q}_1) \quad (7)$$

$$\underline{x}_{2\delta} = (\mathbf{I} - \mathbf{\Gamma}_k \cdot \mathbf{R}_k^{-1} \cdot \mathbf{\Gamma}_k^T \cdot \mathbf{P}_1) \cdot (\mathbf{\Phi}_k \cdot \underline{x}_\delta + \mathbf{\Gamma}_k \cdot \mathbf{R}_k^{-1} \cdot \mathbf{\Gamma}_k^T \cdot \underline{q}_1) \quad (8)$$

Gyakorlati megfontolások azt eredményezték, hogy a szögletes pályapontokban az előírt pályapontok besűrűsödnek mindaddig, amíg a vezérelt kimenet el nem érte ezt a pontot. Ezután az előírt pálya megint felveheti a kiterített állapotát.

### Az optimális vezérlés számításának hierarchikus megközelítése

Legyen az (1) egyenlettel megadott dinamikus rendszer, amelyet a (2) célfüggvény által megadott optimális pályán óhajtunk vezérelni. A Boltianski által megadott klasszikus algoritmus felhasználja a feladathoz rendelt Lagrange és Hamilton függvényeket:

$$\mathbf{L}(\underline{x}, \underline{u}, \underline{p}) = J(\underline{u}) + \sum_{k=0}^{N-1} \langle \underline{p}_k, \mathbf{\Phi}_k \underline{x}_k + \mathbf{\Gamma}_k \underline{u}_k - \underline{x}_{k+1} \rangle \quad (9)$$

$$\mathbf{H}_k(\underline{x}_k, \underline{u}_k, \underline{p}_k) = \frac{1}{2} \langle \underline{x}_k, \mathbf{Q}_k \underline{x}_k \rangle + \frac{1}{2} \langle \underline{u}_k, \mathbf{R}_k \underline{u}_k \rangle + \langle \underline{p}_k, \mathbf{\Phi}_k \underline{x}_k + \mathbf{\Gamma}_k \underline{u}_k \rangle \quad (10)$$

Így a feladatunkat átírhatjuk mint:

$$\mathbf{L}(\underline{x}, \underline{u}, \underline{p}) = \frac{1}{2} \langle \underline{u}_0, \mathbf{R}_0 \underline{u}_0 \rangle + \langle \underline{p}_0, \mathbf{\Phi}_0 \underline{x}_0 + \mathbf{\Gamma}_0 \underline{u}_0 \rangle + \sum_{k=1}^{N-1} \{ \mathbf{H}_k(\underline{x}_k, \underline{u}_k, \underline{p}_k) - \langle \underline{p}_{k-1}, \underline{x}_k \rangle \} + \dots \quad (11)$$

$$+ \frac{1}{2} \langle \underline{x}_N, \mathbf{F} \underline{x}_N \rangle - \langle \underline{p}_{N-1}, \underline{x}_N \rangle$$

És megválasztva a  $\underline{p}_1, \underline{p}_2, \dots, \underline{p}_N$  állapotokat a feladat párhuzamosítása N párhuzamos illetve egy koordináló folyamatra a következő:

$$\underline{u}_0^* = \min_{\underline{u}_0 \in U} \left\{ \frac{1}{2} \langle \underline{u}_0, \mathbf{R}_0 \underline{u}_0 \rangle + \langle \underline{p}_0, \mathbf{\Phi}_0 \underline{x}_0 + \mathbf{\Gamma}_0 \underline{u}_0 \rangle \right\} \Rightarrow$$

$$\Rightarrow \underline{u}_0^* = -\mathbf{R}_0^{-1} \mathbf{\Gamma}_0^T \underline{p}_0 \quad (12)$$



Az első lépésre illetve a köztes,  $k = \overline{2, N-1}$  lépésekre érvényes egyenletek:

$$\begin{aligned} \underline{u}_k^* &= -\mathbf{R}_k^{-1} \mathbf{\Gamma}_k^T \underline{p}_k \\ \underline{x}_k^* &= -\mathbf{Q}_k^{-1} (\mathbf{\Phi}_k^T \underline{p}_k - \underline{p}_{k-1}) \end{aligned} \quad (14)$$

Végül a  $k = N$  utolsó lépése a következő egyenletet kapjuk:

$$\underline{x}_N^* = \min_{\underline{x}_N} \left\{ \frac{1}{2} \langle \underline{x}_N, \mathbf{F} \underline{x}_N \rangle - \langle \underline{p}_{N-1}, \underline{x}_N \rangle \right\} \Rightarrow \underline{x}_N^* = \mathbf{F}^{-1} \underline{p}_{N-1} \quad (15)$$

A koordinációs feladatot a következő egyenletek biztosítják:

$$\underline{p}_k^* = \max_{\underline{p}_k} \{ \mathbf{L}(\underline{x}, \underline{u}, \underline{p}) \} \quad (16)$$

$$\underline{e}_k = \mathbf{\Phi}_k \underline{x}_k + \mathbf{\Gamma}_k \underline{u}_k - \underline{x}_{k+1}, \quad k = \overline{0, N-1},$$

$$\underline{p}_k^{(n+1)} = \underline{p}_k^{(n)} + \gamma_k \frac{\partial \langle \underline{e}_k, \underline{e}_k \rangle}{\partial \underline{p}_k} \quad (17)$$

### A mesterséges neuronháló, mint megoldási algoritmus

A Kennedy és Chua által vizsgált a 2 ábrán bemutatott háló és VLSI változata bizonyítottan használható egyenlőtlenség feltételek melletti optimumkeresési feladatokra. Így a

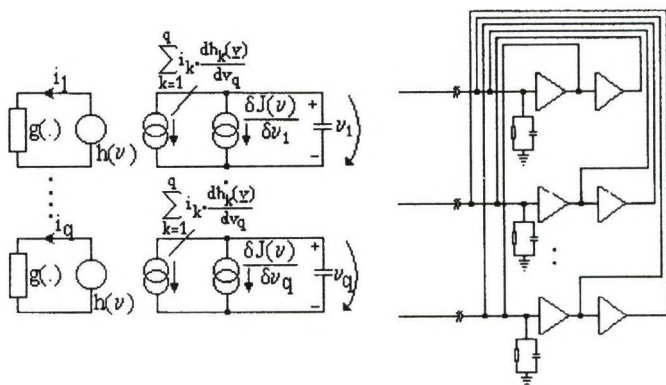
$$\underline{v}^* = \min_{\underline{v} \in \mathcal{D}^m} \{ J(\underline{v}) / h_i(\underline{v}) \geq 0, i = \overline{1, q} \} \quad (18)$$

feladatot megoldhatjuk mint:

$$\frac{dv_j}{dt} = -\lambda_j(\underline{v}) \cdot \frac{\partial \mathbf{L}}{\partial v_j}(\underline{\mu}, \underline{v}) = -\lambda_j(\underline{v}) \cdot \left( \frac{\partial J}{\partial v_j}(\underline{v}) - \sum_{i=1}^q \mu_i \cdot \frac{\partial h_i(\underline{v})}{\partial v_j} \right) \quad (21)$$

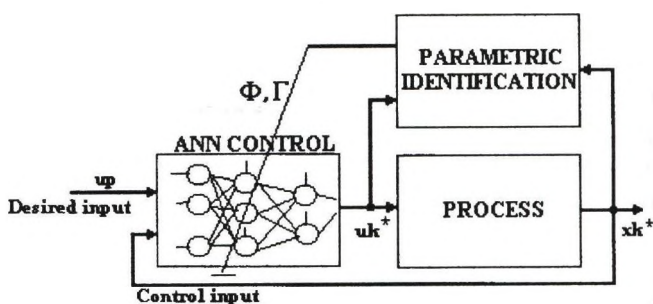
Ahol a  $\mu_i = g_i(\underline{v}) = g_i(h_i(\underline{v}))$  függvény egy passzív monoton nem csökkenő függvény.

$$C_j \cdot \frac{dv_j}{dt} = -\frac{\partial J}{\partial v_j} - \sum_{i=1}^p g_i(h_i(\underline{v})) \cdot \frac{\partial h_i(\underline{v})}{\partial v_j} \quad (22)$$



2. ábra

Természetesen ez a megoldás használható az optimális vezérlésre is a 3. ábra alapján, ha átfogalmazzuk ezt egyenlőtlenség feltételekre.



3. ábra

A tanulmányozott vezérlési struktúra

Mivel a célunk a célfüggvény korlátos minimumának keresése ezért felhasználjuk a

$$\underline{v} = \left( \underline{x}_0^T, \dots, \underline{x}_N^T, \underline{u}_0^T, \dots, \underline{u}_{N-1}^T \right)^T \in \mathcal{R}^{(N+1)n + N \cdot m},$$

állapotokat, az LQ feladat megoldásához.

$$\underline{u}^* = \min_{\underline{u}_k \in U, \underline{x}_k \in X} \{ J(\underline{u}) \} \quad (23)$$

Így az időpontokra lebontott korlátok:

$$A \cdot \underline{v} = \begin{bmatrix} -\Phi_0 & I & 0 & \dots & -\Gamma_0 & 0 & \dots & 0 & 0 \\ 0 & -\Phi_1 & I & \dots & 0 & -\Gamma_1 & \dots & 0 & 0 \\ 0 & 0 & -\Phi_2 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & -\Gamma_{N-2} & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & -\Gamma_{N-1} \end{bmatrix} \cdot \begin{bmatrix} \underline{x}_0 \\ \vdots \\ \underline{x}_N \\ \underline{u}_0 \\ \vdots \\ \underline{u}_{N-1} \end{bmatrix} = \underline{0} \quad (24)$$

Felhasználva Taha tételét (1982) miszerint m egyenlőség feltételt

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad i = \overline{1, m} \quad (25)$$

átírhatunk m+1 egyenlőtlenység feltételre kapjuk:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad i = \overline{1, m}; \quad \sum_{j=1}^n \left( \sum_{i=1}^m a_{ij} \right) x_j \geq \sum_{i=1}^m b_i \quad (26)$$

Így az LQ vezérlés feladata megfogalmazható mint:

$$\underline{u}^* = \min_{\substack{\underline{u}_k \in U, \\ \underline{x}_k \in X}} \left\{ J(\underline{u}) / h_i(\underline{v}) = \sum_{j=1}^{(N+1)n+m} \Lambda_{ij} \cdot \underline{v}_j \geq 0, i = \overline{1, N \cdot n + 1} \right\} \quad (27)$$

ahol a  $\Lambda_{ij}$  elemek a következő mátrixból számíthatóak ki:

$$A = \begin{bmatrix} -\Phi_0 & -I & 0 & \dots & -\Gamma_0 & 0 & \dots & 0 & 0 \\ 0 & -\Phi_1 & -I & \dots & 0 & -\Gamma_1 & \dots & 0 & 0 \\ 0 & 0 & -\Phi_2 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & -\Gamma_{N-2} & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & -\Gamma_{N-1} \\ \Omega^T & \Omega^T - \underline{I} & \Omega^T - \underline{I} & \dots & \Psi^T & \Psi^T & \dots & \Psi^T & \Psi^T \end{bmatrix}$$

$$\Omega = \left[ \sum_{i=1}^n \Phi_{i1} \quad \sum_{i=1}^n \Phi_{i2} \quad \dots \quad \sum_{i=1}^n \Phi_{in} \right]^T$$

$$\Psi = \left[ \sum_{i=1}^n \Gamma_{i1} \quad \sum_{i=1}^n \Gamma_{i2} \quad \dots \quad \sum_{i=1}^n \Gamma_{im} \right]^T$$

$$\underline{I} = [1 \quad 1 \quad \dots \quad 1]^T$$



## Vizsgált példák

A fenti módszert alkalmaztuk a következő feladatra.

$$J(v_1, v_2) = v_1^2 + v_2^2 + v_1 \cdot v_2 + 2 \cdot v_1$$

$$h_1(v_1, v_2) = v_1^2 + v_2^2 - 1,5 \leq 0$$

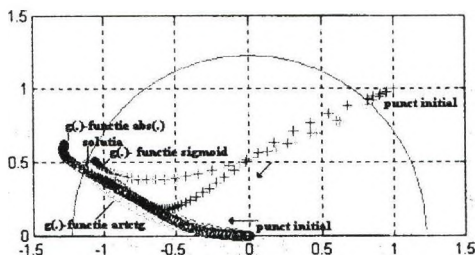
$$h_2(v_1, v_2) = v_1 \leq 0$$

$$h_3(v_1, v_2) = -v_2 \leq 0$$

Ahol a Hopfield hálót leíró egyenletek a következők:

$$C_1 \cdot \frac{dv_1}{dt} = -2 \cdot v_1 - v_2 - 2 - 2 \cdot g(v_1^2 + v_2^2 - 1,5) \cdot v_1 - g(v_1)$$

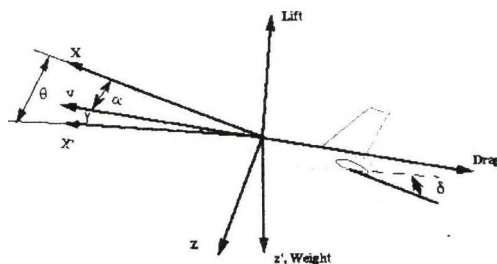
$$C_2 \cdot \frac{dv_2}{dt} = -2 \cdot v_2 - v_1 - 2 \cdot g(v_1^2 + v_2^2 - 1,5) \cdot v_2 + g(-v_2)$$



4. ábra

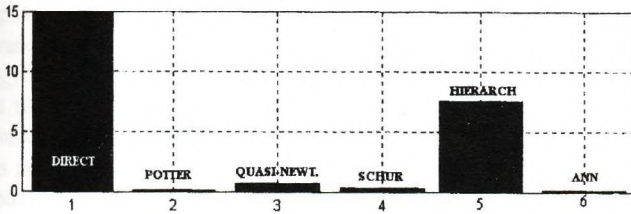
*Az optimumkeresés megoldásának szimulálása Hopfield háló esetében*

Dinamikus vezérlés megoldására az 5 ábrán bemutatott repülőgép pálya vezérlése esetében tanulmányoztuk az algoritmus hatékonyságát.



5. ábra

És a kapott eredmények számítási idő szerinti rangsorolása látható a 6 ábrán.



6. ábra

Végezetül megállapíthatjuk, hogy hierarchikus párhuzamos algoritmus konvergenciáját nagymértékben befolyásolja a  $g(\cdot)$  függvény megválasztása esetenként elfogadhatatlanul megnövelve a számítási időt, míg a neuronháló alkalmazható jó megoldást eredményez.

### Könyvészet

- [1.] B.D.O.Anderson, Optimal Control, Linera Quadratic Methods, Prentice-Hall, 1989.
- [2.] A.J.Laub *Schur method for solving algebraic Riccati equations*, IEEE Transaction on Automatic. Control 24,913-921,1979.
- [3.] C.T.Leondes, Control and Dynamics Systems, Advances in Theory and Applications, Academic Press 1996 .
- [4.] S.G.Tzafestas, *Applied Digital Control*, North,Holland Systems and Control,1985
- [5.] L.William,Ph.D.Brogan, Modern Control Theory, Prentice-Hall, 1989.
- [6.] A. J. N. VAN BREEMAN. *Neural Adaptive Control*. Master's thesis, University of Twente, 1997;
- [7.] CHEN F.C, KHALIL H.K, *Adaptive control of nonlinear systems using neural networks*, Int. J. Control, 55(6):1299-1317, 1992;

# Fejlett neuronmodellek szimulációja és megvalósítása

Simulation and Implementation  
of Advanced Neural Models

Bakó László, Brassai Sándor Tihamér

SAPIENTIA EMTE – Marosvásárhely, Románia

## Abstract

*Neurobiological experiments have lead to the development of the third neural networks generation, the neuromorphic artificial neural networks. One of the potential advanced neural models is that based on the "spiking" neural behavior observed in the biological brain. In this paper the biological background, the mathematical modeling and computer simulation of spiking neural networks are discussed, with application to a letter recognition device. A FPGA implementation of the modeled neuron is also presented.*

## Bevezetés

A neurobiológia kísérleti eredményei vezettek el a fejlett (harmadik generációs) mesterséges neuronális hálózatok megjelenéséhez, melyek igyekeznek természetazonos neuronmodelleket kiépíteni. Az alapul vett lehetséges neuronmodellek egyike impulzussorozatokra építi működését, a biológiai agyban észlelhető úgynevezett "neurontüzelési" jelenséget kihasználva. Ezen neuronok vizsgálata, modellezése, szimulációja illetve mesterséges rendszerekben való minél jobb megközelítéssel való megépítése képezi kutatásunk célját.

## 1. A kutatás biológiai alapjai

### *A természetes tüzelő neuron*

Egy tipikus természetes neuron három különálló funkcionális részre osztható: dendrit (bemenet), sejttest (szoma) és axon (kimenet). A sejttest a központi feldolgozó egység, amely egy fontos nemlineáris feldolgozási lépést hajt végre: ha az össz-kimenet meghalad egy bizonyos küszöbértéket,



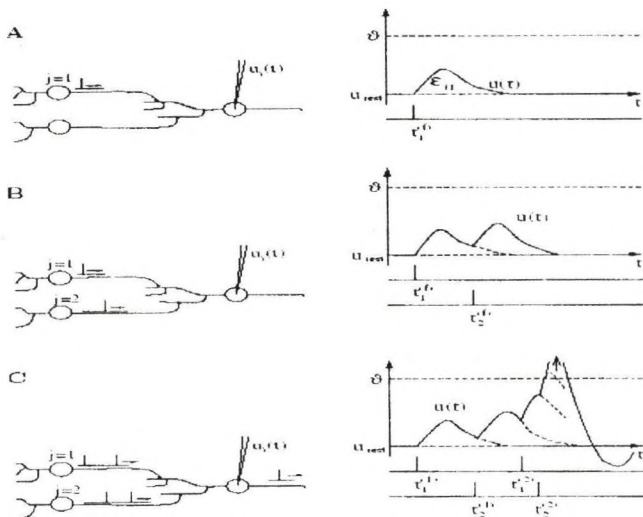
egy kimeneti jel – akciós potenciál – generálódik [1,2]. Két neuron közötti csatlakozást szinapszisnak nevezünk. Egy jelet küldő neuronra, mint preszinaptikus neuronra, míg a fogadó neuronra, mint posztszinaptikus neuronra hivatkozunk.

Az agyban észlelhető akciós potenciálok rövid elektromos impulzusok, melyeket tüzelésnek nevezünk, az amplitúdójuk körülbelül 100 mV és az időtartamuk jellegzetesen 1-2 ms. Az impulzus formája nem változik, amint az akciós potenciál terjed az axon mentén. Egy neuron által kibocsátott akciós potenciál-láncot tüzelési impulzus sorozatnak nevezünk, egy sztereotipikus eseménysorozat, amely szabályos vagy szabálytalan időintervallumonként következik be. Maga az akciós potenciál formája nem hordoz semmiféle információt, a hangsúly az impulzusok számán és időzítésén van. Két tüzelés közötti minimális távolság meghatározza a neuron abszolút ellenállási periódusát. Az abszolút ellenállási periódust követi a relatív ellenállási fázis, ahol nehéz de nem lehetetlen egy akciós potenciált kiváltani.

### *Szinapszisok. A neuronális dinamizmus*

Azt a pontot, ahol egy preszinaptikus neuron axonja egy poszt-szinaptikus neuron receptorához (dendritjéhez) kapcsolódik szinapszisnak nevezzük. Egy tüzelés hatása a posztszinaptikus neuron egy intracelluláris elektróddal észlelhető, amely méri a potenciál különbséget  $u(t)$  a neuron belseje és közvetlen környezete között. Ezt a potenciál különbséget nevezzük membrán potenciálnak. Bemeneti tüzelés hiányában, a neuron nyugalmi állapotban van, és a nyugalmi állapotnak megfelel egy membrán potenciál. Egy tüzelés érkezése után, a potenciál megváltozik és végül visszaesik a nyugalmi potenciálra. (1. ábra). Ha a változás pozitív, a szinapszis serkentő, ha a változás negatív a szinapszis gátló. Nyugalmi állapotban, a cella membránja negatívan polarizált, értéke körülbelül  $-65$  mV. Egy serkentő szinapszison megjelenő bemenet a membrán negatív polarizációját csökkentő hatását depolarizációnak, míg az ezt növelő hatást hiperpolarizációnak nevezzük.

Feltételezzünk két preszinaptikus neuront  $j = 1, 2$ , melyek közül mindkettő küld egy impulzust az  $i$ -ik posztszinaptikus neuronhoz. A  $j = 1$  neuron tüzel a  $t_1^{(1)}, t_1^{(2)}, \dots$ , időpontokban, hasonlóan a  $j = 2$  neuron tüzel a  $t_2^{(1)}, t_2^{(2)}, \dots$  időpontokban. Mindkét impulzus előidéző egy-egy posztszinaptikus potenciált  $\epsilon_{i1}$  vagy  $\epsilon_{i2}$ , külön-külön. Mindaddig, amíg csak néhány bemeneti impulzus van, a potenciál változása közel megegyezik az egyedi posztszinaptikus potenciálok összegével, a membrán potenciál lineárisan válaszol a bemeneti impulzusokra (1.B ábra):



1. ábra

Az  $i$ -ik posztiszinaptikus neuron bemeneteket fogad két preszinaptikus neurontól  $j=1,2$ . Mindkét preszinaptikus impulzus előidéz egy serkentő posztiszinaptikus potenciált

$$u_i(t) = \sum_j \sum_f \epsilon_{ij} (t - t_j^{(f)}) + u_{\text{rest}}$$

Ha sok bemeneti impulzus érkezik egy rövid időintervallumon belül a linearitás érvényét veszti. Amint a membrán potenciál eléri egy kritikus értéket, a potenciál görbéje impulzusszerű viselkedést mutat melynek amplitúdója körülbelül 100 mV. Ez az akciós potenciál terjed majd az  $i$  neuron axonján a többi neuron szinapszisai fele. Tüzelés után a membrán potenciál nem tér vissza közvetlenül a nyugalmi potenciálra, hanem egy az alatti, hiperpolarizációs fázison megy keresztül. Az egyedi serkentő posztiszinaptikus potenciálok amplitúdója az egy milivolt tartományban van. A tüzelés eléréséhez szükséges kritikus érték körülbelül 20÷30 mV-al a pihenési potenciál fölött helyezkedik el. Amint az a 1.C ábrán is látható, négy posztiszinaptikus potenciál nem elegendő egy tüzelési impulzus kiváltására. Ennek érdekében legalább 20-50 preszinaptikus potenciál, rövid idő alatt való érkezése szükséges.

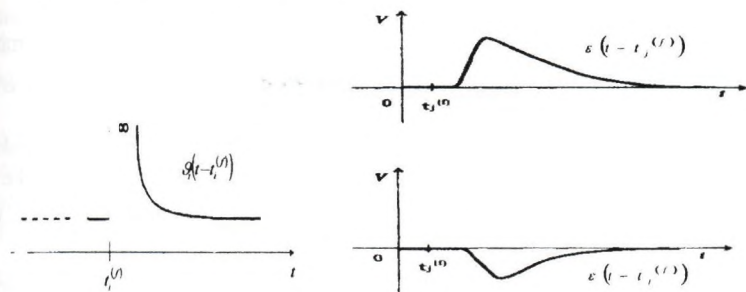
## 2. Neuronális modellek kidolgozása

Szinte bármilyen számítógépről legyen szó, sokszor ahhoz ragaszkodunk, hogy az egyes számítási lépések időzítése egy globális idősemába

illeszkedjen be. Például, egy *feedforward* neuronális hálózat  $x$ -ik rétegének valamely kapujától azt várjuk el, hogy kimenete a számítás  $x$ -ik időpillanatban aktiválódjon a hálózat bemenén fellépő jelektől függetlenül. Ezzel ellentétben egy biológiai neuronális hálózatban a neuronok tüzelési idejét a rendszer bemeneti befolyásolják. A tüzelő neuronokból felépített hálózatok, ezért, képesek az időt, mint kódolási és számítási erőforrást – a legtöbb létező számítási modellt fölülmúlva – kihasználni.

### Tüzelő neuronokból felépített formális neuronális hálózat modellje

Tekintsünk egy tüzelő neuronális hálózatot (TNH) [3,4] melyet egy véges, tüzelő neuronokból (TN) álló  $V$  halmaz és egy  $E \subseteq V \times V$  szinapszisokból álló halmaz alkot. Minden  $i \in V$  neuronhoz tartozik egy  $\vartheta_i : R^+ \rightarrow R^+$  küszöbérték függvény (2A. ábra) és minden  $\langle i, j \rangle \in E$  szinapszishoz egy  $\varepsilon_{i,j} : R^+ \rightarrow R$  válaszfüggvény (ahol  $R^+ = \{x \in R : x \geq 0\}$ ) (2B. ábra).



2. ábra

A) Természetes tüzelő neuron küszöbérték függvényének alakja (balra)

B) Természetes tüzelő neuron válaszfüggvényének alakja - SPSP és GPSP, (jobbra)

Egy  $j$  tüzelő neuron bizonyos  $t_j^{(1)}, t_j^{(2)}, \dots$ , időpillanatokban bocsát ki TI-t azaz egy sztereotipikus impulzust közvetít az axonjának elágazó struktúráján keresztül. Ennek következtében pozitív (serkentő posztszinaptikus potenciálok: SPSP) vagy negatív (gátló posztszinaptikus potenciálok: GPSP) impulzusokat generálódnak melyek változtatni fogják a posztszinaptikus neuron membránpotenciálját.

A  $j$  neuron tüzelése által kiváltott SPSP-k és GPSP-k időbeni karakteristikája a következőképpen írható le:  $\omega_{ij} \varepsilon(t - t_j^{(i)})$ , ahol az



$\varepsilon(t - t_j^{(j)})$  válaszfüggvény értéke 0, ha  $(t - t_j^{(j)})$  kisebb, mint a  $\Delta_{ij}$  közvetítési idő, majd szinte lineárisan növekszik és végül exponenciálisan csökken vissza 0-ra. A  $\omega_{ij}$  faktor (súlyzási tényező) pozitív az SPSP-ek esetében és negatív a GPSP-k esetében. Matematikai egyszerűsége miatt az *impulzus-reakciós modell* az egyik legmegfelelőbb a digitális alkalmazásokban a posztszinaptikus tüzelések időpillanatainak megbecsülésére.

#### *Az impulzus-reakciós modell*

Az impulzus-reakciós modell (IRM) esetében valamely  $i$  neuron sejttestének (szoma)  $u_i^{(0)}$  membránpotenciálja, egy adott  $t$  időpillanatban, a következőképpen fejezhető ki  $\sum_{j,f} \omega_{ij} \varepsilon(t - t_j^{(f)})$ , szumma, melyet a  $j$  preszinaptikus neuronok által egy  $t$  előtti,  $t_j^{(f)}$  időpillanatban kibocsátott SPSP-ok és GPSP-ok alkotják. Mindemellett az említett modell neuronjainak *fojtottsági tényezője*,  $\eta(t - \hat{t}_i)$ , a  $\hat{t}_i$  függvénye, mely megadja azt a  $t$  előtti időpillanatot, amikor az  $i$  neuron utoljára tüzelt. Ha pozitív, de nagyon kis értéke van (kevesebb mint 2 ms), akkor  $\eta(t - \hat{t}_i)$  erősen negatív értéket ad, ugyanakkor, ha  $(t - \hat{t}_i)$  értéke nagyobb, a *fojtottsági tényező* értéke újra nulla felé tart. Az impulzus-reakciós modell szerint tehát egy  $i$  neuron akkor tüzel, ha az eredő szumma értéke elér egy bizonyos  $\mathcal{G}$  küszöbértéket és e kifejezés elsőrendű deriváltja pozitív, azaz a küszöbértéket alulról közelíti meg.

A modell számítástechnikai alkalmazásokra való megfelelőségét javíthatja ha feltételezzük, hogy neuronok membránpotenciálja nyugalmi helyzetben (posztszinaptikus impulzusok hiányában) nulla és a küszöbérték mindig nagyobb mint nulla  $\mathcal{G} > 0$ . A *Hodkin-Huxley modellhez* [2,4] viszonyítva ez a modell sokkal könnyebben alkalmazható digitális számítógépeken, mivel meghatározásában nem szerepelnek differenciálegyenletek. Másrészt, ha megfelelően választjuk a  $\eta$ ,  $\mathcal{G}$  és az  $\varepsilon$  függvényeket az *impulzus-reakciós modell* képes jól megközelíteni a természetes, biológiai neuronok dinamikáját.

#### **4. Tüzelő neuronális hálózatok szimulációja**

Szimulációs programunk kiindulási pontja a fennebb bemutatott impulzus-reakciós modell, melyet a neurononkénti egy impulzusos kódolással alkalmaztunk.

## *A szimuláció algoritmusa*

A szimulációs eljárások fő célja azon hálózati paraméterek kiszámítása melyek szükségesek minden TI generálásához a folyamatos hálózatfeldolgozás egy adott időpillanatában. A TI-ok a neuronok membrán potenciáljaiból és a küszöbérték függvények segítségével generálódnak. Szükséges tehát, hogy a szimuláció során feldolgozzuk a megfelelő pre-és posztszinaptikus impulzusokat érvényes membránpotenciálakká minden olyan neuron számára, mely a következő szimuláció-időlépésben tüzelni fog. Ugyanakkor érvényes szinaptikus súlyzásokat és késleltetéseket kell kiszámítani. Mindezen paramétereket a számítógép memóriájában lefoglalt struktúrában tároljuk, struktúra mely a hálózat állapotát adja meg.

Az időbeni következetesség elérésére az egyik módszer a szimulációs folyamat minden időlépésének két fázisra osztása: az elsőben kiszámítjuk az összes neuron TI-ait majd ezeket felhasználjuk a második fázisban, amikor a TI-k közvetítéséhez szükséges műveleteket végezzük el a következőképpen:

1. Lépés: a neuron memóriából kiolvasott értékekkel membránpotenciálok és küszöbértékeket számítunk ki. Ha valamely membránpotenciál küszöbérték fölött helyezkedik el akkor TI képződik (listába mentés neuronazonosítóval társítva)

2. Lépés: a TI listából kiolvasott értékeket súlyozzuk ( $w_{ij}$ ) és a felépített hálózat topológiáját felhasználva továbbítjuk a posztszinaptikus neuronok felé (tanítási algoritmus (Hebb) – a szinapszisok hatékonyságának és időzítések hangolása). [2]

Általános megfogalmazásban a Hebb-féle tanulás három összetevővel fejezhető ki:

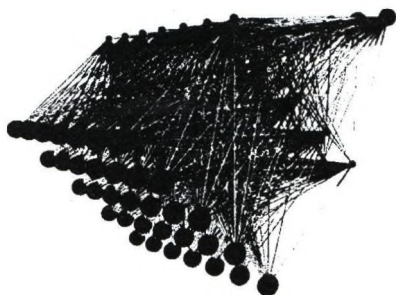
1. Szinapszis hatékonyság nő, ha a pre- és posztszinaptikus neuronok egyidőben tüzelnek.
  2. Szinapszis hatékonyságának csökken, ha csak a pre- vagy a posztszinaptikus neuron aktív.
  3. A neuronális kapcsolat nem változik, ha egyik neuron sem aktív.
- A Hebb-féle tanítás egy nem felügyelt tanítási módszer.

## *A szimulációs program*

A feladat komplexitására és a pontos időzítések szükségességére való tekintettel igen nagy figyelmet kellett fordítani a szimulációs programok környezetének a kiválasztására. Hosszabb kísérletezés után RedHat Linux operációs rendszer alatti C++ fordító mellett döntöttünk. Lényeges előnyökhöz jutottunk, mint például a megépített neuronális hálózatok könnyebb háromdimenziós ábrázolása (Povray, Gnuplot grafikus csomagok), valamint a modern és időszerű XML formátumban való neuron hálózati struktúramentések.

Szoftverfejlesztési munkánkat a korábban kidolgozott és az előző fejezetekben leírt *impulzus-reakciós modell* és a *küszöbérték-alapú tanulási módszerre* alapozott szimulációs környezet kidolgozására fordítottuk. Ide tartozik, az egyedi tüzelő neuronok modellének és a szinapszisoknak a programozása, a rétegépítési eljárások és a rétegek közötti összeköttetések megvalósításához szükséges függvények megírása. Bármilyen 3D elhelyezkedésben elképzelt neuronális hálózat kiépíthető neurononként vagy pedig rétegenként adva meg a hálózat architektúráját. A program három típusú neuronnal dolgozik: bemeneti réteg neuronok, köztes vagy rejtett réteg neuronok és kimeneti réteg neuronok.

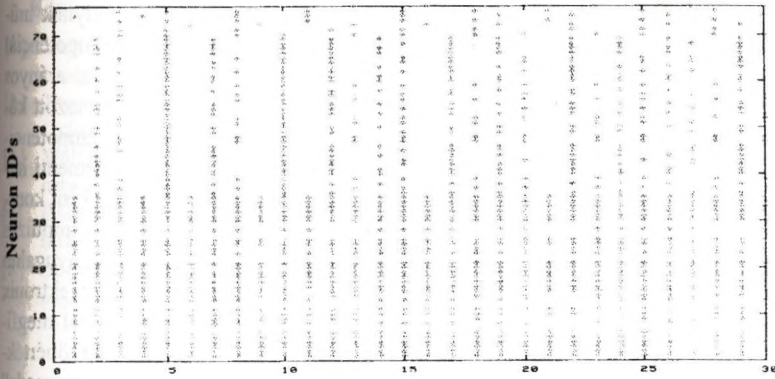
Ebben a környezetben egy  $7 \times 5$ -ös mátrixból való karakterfelismerést végző hálózatot építettünk. (3. ábra – kék: bementi neuronok, barna: rejtett réteg neuronok, piros: kimeneti réteg neuronok). A 7 kimenő neuron megfelel egy 7-szegmenses kijelző részeinek. A hálózat a bemenetén megjelenő (logikai 0 vagy 1 értékeként) zajos karaktereket képes osztályozni, amint az a 4-es ábrán is követhető. (az Y tengelyen a neuron-azonosítókat (ID) ábrázoltuk, az X tengelyen pedig az időlépés számát). Neuron ID-k: 0÷34 bemeneti réteg, 34÷69 rejtett réteg, 70÷76 kimeneti réteg.



3. ábra  
A megépített neuronhálózat

A szimuláció minden lépésében egy újabb bementi vektor jelenik meg a hálózat bementi rétegén, majd a szükséges szinaptikus számítások hajtódnak végre. Megfelelő számú lépés után bizonyos kimeneti neuron csoportok együttes tüzelése ugyanarra a bementi karakterre, konvergens eredményhez vezet. A 4-es ábra egy ezer lépésben futtatott szimuláció első harminc lépésében mutatja a neuronok tüzelését.

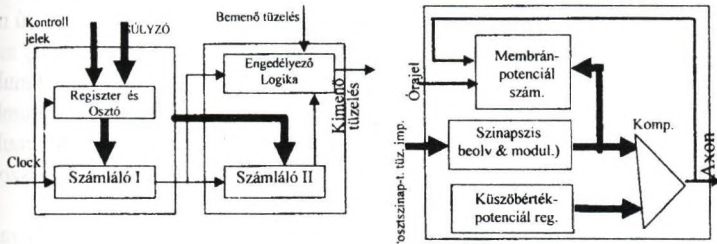




4. ábra  
Mérési eredmény, a szimuláció első 30 lépése

### 5. A tüzelő neuronmodell FPGA implementációja

A kidolgozott elméleti neuronmodellre, a megvalósított szimulációs program eredményeire valamint az [5]-re támaszkodva kezdtük el az FPGA környezetben való gyakorlati megvalósítást. A neuron két fő részét, a szinapszis és a szomat (sejttest) külön-külön terveztük és valósítottuk meg. A hardver egy XSA100 fejlesztőkártya, XC2S100 Spartan II FPGA-val, (100ezer logikai kapu). Az FPGA-ban megvalósított szinapszis egy impulzus-sokszorozó áramkör, amely ha impulzust észlel a bemenén, akkor a rendszer órajel frekvenciájának, a felprogramozott súlyzó értékével való egész osztása révén kapott számú kimenő impulzust (tüzelést) továbbít a szoma fele (5. ábra).



5. ábra  
Balra a szinapszis, jobbra a szoma elvi felépítésének rajza

A szomat ugyancsak VHDL-ben megírt áramkör alkotja melynek működése röviden a következőképpen foglalható össze: a membránpotenciál értékét tároló számláló növekszik a bementi impulzusok számával arányos módon. Ha az így kiszámolt érték meghaladja az előre felprogramozott küszöbérték potenciált, akkor a neuron tüzelni fog. Ezután a membránpotenciál hiperpolarizációs fázisba kerül, értéke nullázódik. Ha nincs bementi impulzus, akkor a membránpotenciál fokozatosan csökken, órajelenként konstans, állítható értékkel. Ha nem érkezik megfelelő számú bemenő impulzus egy adott időintervallumon belül, a membránpotenciál visszatér a nyugalmi értékre. Ez a folyamat végigkövethető a 6. Ábrán, melyen egy Tektronix logikai analizátorral mintavételezett mérési eredmény látható, ahol megfigyelhetjük, hogy az ennél a kísérletnél 143-ra beállított küszöbérték-potenciál elérése után jelentkezik egy axon-tüzelés. A teljes neuronmodell egy szoma és nyolc szinapszis egybeépítésével áll össze.



6. ábra

Az FPGA fejlesztőkértyáról mintavételezett kísérleti eredmények egy része

## 6. Következtetések, tervek

Kutatásunk 2002 utolsó negyedében indult és 2003-ban részben a Sapiaentia-KPI támogatásával zajlott. A bemutatott, jelenlegi állapot mér-földkő szerepet tölt be tevékenységünkben, az első szimulált és hardverben megvalósított, fejlett neuronmodellre épülő, számos kísérletben tesztelt, működőképes mesterséges neuron létrehozása által. Az utolsó verziójú modell hiányossága a relatív magas hardverigény (~15000 logikai kapu) mely egyértelműen optimalizációt igényel. A jelenleg beépített Hebb-féle tanulási módszer csupán a szinapszisok hangolását alkalmazza, melyre az áramkör megfelelőképpen elő van készítve. Terveink szerint a fejlettebb tanulási algoritmusokat szeretnénk majd kidolgozni, melyek kiaknázzák a szoma paraméterhangolási lehetőségeit is.

Ilyen irányvonalat követve tervezzük a mesterséges neuronális hálózatok hardver-kiépítését is. Olyan *genetikus algoritmussal* működő rendszert szeretnénk kidolgozni, amely topológiai önhangolásra képes akárcsak a természetes agy.

## Könyvészet

- [1.] Wulfram Gerstner and Werner M. Kistler; Spiking Neuron Models. Single Neurons, Populations, Plasticity; Cambridge University Press, 2002.
- [2.] Thomas P. Trappenberg; Fundamentals of Computational Neuroscience; Oxford University Press, 2002.
- [3.] Wolfgang Maas, Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 10(9):1659-1671, 1997.
- [4.] (Maas and Bishop, 1999) Wolfgang Maas and Christopher M. Bishop, Pulsed Neural networks, MIT Press, 1999.
- [5.] (Aghdzen, 1990) Peter J. Ashenden, The VHDL Cookbook (First Edition), *Dept. Computer Science, University of Adelaide, South Australia* (July, 1990)



# A dinamikus öröklődés

## The Dynamic Inheritance

Kovács Lehel István

Babeş-Bolyai Tudományegyetem

### Abstract

*Concerning the object-oriented programming many open questions have been left unanswered. One issue is what happens when some properties are changing considerably in time? Example: the frogs and the butterflies are experiencing many changes during their lifetime. Like the spawn – tadpole – frog or the egg – larva – chrysalis – butterfly transition. Can we model these transitions? Another example is when we represent the vehicles as containers with their components. What happens in the case of a crush? Some components will “fly out” from the container, becoming independent, by having they own orbit. How can we model dynamically their behavior? How can we design dynamically new methods? We must pay attention to the major role of the random, chaos, and entropy...*

*Dynamic inheritance means that the inheritance hierarchy of classes is changing at run-time. Dynamic inheritance allows objects to change and evolve over time. While parent classes provide for objects variables and methods, changing the parent classes, the variables and methods of a child class are changing too. More specifically, dynamic inheritance refers to the ability to add, delete, or change parents from classes at run-time. In this case the egg – larva – chrysalis –butterfly transformations can be realized in the following way: we define a class for each transitional state of the butterfly, the creature during its life-cycle will be an instance of a class, which inherits dynamically one after another from each class.*

**Key Words and Phrases:** *object-oriented programming, dynamic inheritance.*

### 1. Bevezetés

A klasszikus objektumorientált paradigma – amely a valóság megközelítésének, modellezésének, ábrázolásának egy módszere, amint azt a [6.]-ban is kifejtettük – a modellezés során a valós tárgyakkól objektumokat absztrahál, amelyeket állapotával (tartalmával, adataival) és metódusaival jellemez. Az objektumokat a számunkra lényeges tulajdonságaik, viselkedési módjuk alapján megkülönböztetjük és kategóriákba, osztályokba soroljuk őket, oly

módon, hogy a hasonló tulajdonságokkal rendelkező objektumok egy osztályba, a különböző vagy eltérő tulajdonságokkal rendelkező objektumok pedig külön osztályokba kerülnek. Az objektum-osztályok hordozzák a hozzá tartozó objektumok jellemzőit. Minden *objektum* valamilyen osztály *példánya* (*instance*), rendelkezik osztályának sajátosságaival, átveszi annak tulajdonságait az adatszerkezetre és a műveletekre vonatkoztatva egyaránt.

A klasszikus objektumorientált paradigma jól leosztott szerepkörökkel és jól meghatározott tulajdonságokkal ruházza fel elemeit.

Az *egységbezárás* (*encapsulation*) azt jelenti, hogy az adatstruktúrákat és az adott struktúrájú adatokat kezelő metódusokat kombináljuk; azokat egy egységként kezeljük, és elzárjuk őket a külvilág elől. Az objektumok állapota és a viselkedése, a feladatok elvégzésének a *hogyan*-ja az objektum belülye. Az aktuális belső implementáció (művelet) el van rejtve a rendszer többi részétől. Az objektum belseje sérthetetlen. Az védi adatait, nem enged, hogy ahhoz idegen objektumok hozzáférjenek, hanem csak saját eljárásai dolgozhatnak velük.

*Öröklődésről* (*inheritance*) akkor beszélünk, ha már meglévő osztályokat használunk fel új osztályok definiálására, azzal a céllal, hogy a már meglévő kódot újra fel tudjuk használni, illetve, hogy működésében kibővítsük, testre szabjuk a már meglévő osztályt.

A *polimorfizmus* (*polymorphism*) azt jelenti, hogy ugyanarra az üzenetre különböző objektumok különbözőképpen reagálhatnak, minden objektum a saját (az üzenetnek megfelelő) metódusával.

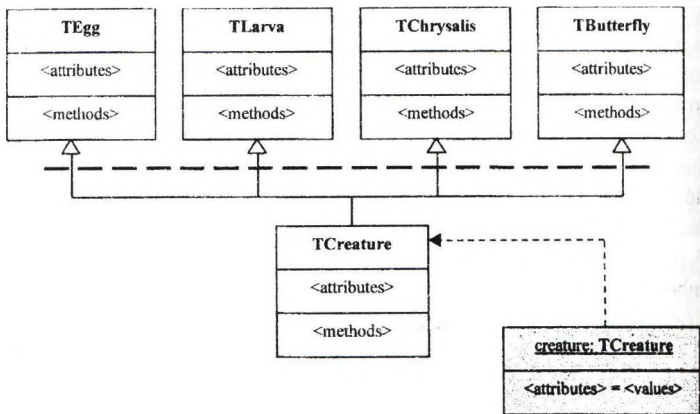
A futás alatti kötés (*late binding*) az objektumorientált paradigma egyik legszebb, de talán legnehezebben érthető fogalma. Megoldást szolgáltat azonban a teljes polimorfizmus megvalósítására. A futás alatti kötés azt jelenti, hogy bizonyos metódusok nem fordítási időben, hanem később, futáskor kötődnek a hívás helyéhez. Talán ez a tulajdonság jelenthetné az első áttörést az objektumok dinamikus építése felé.

Egy objektumorientált program pedig nem más, mint egymással kommunikáló objektumok összessége, melyben minden objektumnak megvan a jól meghatározott feladata.

Jelen cikk célja annak vizsgálata, hogy miként lehet a klasszikus objektumorientált paradigmát kiterjeszteni úgy, hogy az objektumok felépítése dinamikussá váljon. Maradjanak meg a fent említett tulajdonságok, ezek működjenek hatékonyan, de az objektumok struktúrája, a különböző osztályok és objektumok közötti hierarchiák, az objektumok és osztályok közötti viszonyok dinamikusán épüljenek fel a program futása során.

## 2. Dinamikus öröklődés

A dinamikus öröklődés azt jelenti, hogy az osztályok öröklődési hierarchiája futás közben változik. A dinamikus öröklődés biztosítja az objektumoknak az időbeli változást és fejlődést. Az őszosztályok adatait és metódusait biztosítanak az objektumoknak, ha lecseréljük dinamikusan az őszosztályokat, megváltoztathatjuk az objektumok adatait, metódusait. Ebben az esetben a pete (egg) – lárva (larva) – báb (chrysalis) – pillangó (butterfly) átmenet úgy valósulna meg, hogy a pillangó minden köztes állapotának megfeleljék egy osztály, az élőlény (creature) pedig az életciklusa során egy olyan osztály példánya lenne, amely futás időben sorra mindegyik osztály leszármazottja.



– Dinamikus öröklődés –

A dinamikus öröklődés értelmezhető többszörös öröklődés (multiple inheritance) esetén is. Ebben az esetben azt jelenti, hogy dinamikusan lehet őszosztályt kitorölni vagy újat hozzáadni a meglévőkhöz. Többszörös öröklődésről akkor beszélünk, ha egy osztály egyidőben egyszerre kettő vagy több osztálytól örökli adatait, metódusait. A dinamikus öröklődés nem küszöböli ki a többszörös öröklődés anomáliáit, de a dinamikus öröklődés hozzájárulhat ahhoz, hogy az osztályok száma csökkenjen, és a rendszer tervezése és kezelése könnyebbé váljon. Az objektumorientált szoftverfejlesztést bővebben a [3.] és [6.] tárgyalja, az öröklődés válfajairól és az alábbi jelölésről bővebben a [7.]-ben írtunk.

Formálisan így írhatjuk le a dinamikus öröklődést:



Legyenek  $P_1, P_2, \dots, P_n$  osztályok,

$V_{P_1} = \{v^1_{P_1}, v^2_{P_1}, \dots, v^{k_1}_{P_1}\}$  a  $P_1$  osztály adattagjai,

$M_{P_1} = \{m^1_{P_1}, m^2_{P_1}, \dots, m^{k_2}_{P_1}\}$  a  $P_1$  osztály metódusai,

$$P_1 = M_{P_1} \cup V_{P_1},$$

$V_{P_2} = \{v^1_{P_2}, v^2_{P_2}, \dots, v^{k_3}_{P_2}\}$  a  $P_2$  osztály adattagjai,

$M_{P_2} = \{m^1_{P_2}, m^2_{P_2}, \dots, m^{k_4}_{P_2}\}$  a  $P_2$  osztály metódusai,

$$P_2 = M_{P_2} \cup V_{P_2},$$

...

$V_{P_n} = \{v^1_{P_n}, v^2_{P_n}, \dots, v^{k_{2n-1}}_{P_n}\}$  a  $P_n$  osztály adattagjai,

$M_{P_n} = \{m^1_{P_n}, m^2_{P_n}, \dots, m^{k_{2n}}_{P_n}\}$  a  $P_n$  osztály metódusai,

$$P_n = M_{P_n} \cup V_{P_n},$$

legyen  $C$  osztály úgy, hogy az  $c$ :  $C$  objektum életciklusának tetszőleges  $t$  időpillanatban (futási idő):

$$C(t) \leq P_i \mid C(t) \triangleleft P_i, \quad i \in [1, n]$$

többszörös öröklődés esetén pedig:

$$C(t) \leq \{P_{i_1}, \dots, P_{i_2}\} \mid C(t) \triangleleft \{P_{i_1}, \dots, P_{i_2}\}, \quad i_1, i_2 \in [1, n]$$

a  $t$  időpillanatban  $\gamma_t \in \Gamma$  kiválasztási feltétel alapján

választjuk ki az  $i$ -edik őst,

vagy az  $i_1 \dots i_2$  ősoket.

(1.)

A  $P_i$  őosztályok, a  $C$  leszármazott osztály, és  $\Gamma$  egy feltétel rendszer.

A  $\leq$  és  $\triangleleft$  szimbólumokat a következőképpen értelmezzük:

**2.1. Definíció:** (A helyettesíthetőség fogalma) A származtatott osztály objektumai bármilyen körülmények között helyettesíteni tudják az őosztály objektumait. Ha  $C$  a származtatott osztály,  $P$  az őosztály,  $C = \text{subst}(P)$  azt jelenti, hogy a  $C$  bármely példánya használható ott, ahol a  $P$  bármely példánya előfordul.

**2.2. Definíció:** (Altípus, sub-type) Az altípus egy olyan osztály, amely kielégíti a helyettesíthetőség fogalmát ( $C = \text{subst}(P)$ ). Formálisan:  $C \leq P$ .

**2.3. Definíció:** (Alosztály, sub-class) Az alosztály egy olyan öröklődéssel létrehozott tetszőleges osztály, amely nem elégíti ki a helyettesíthetőség fogalmát ( $C \neq \text{subst}(P)$ ). Formálisan:  $C \triangleleft P$ .

A dinamikus öröklődés előnyei:

- Dinamikusan, futási időben le tudjuk cserélni egy objektum osztályát.
- A leszármazott osztályok dinamikusan hozzáadhatók a szülő által ismert osztályok listájához.
- Könnyen, a kód megváltoztatása nélkül lehet új osztályokat definiálni.
- Csökkenti az osztályok számát.
- Könnyen megfeleltethető a tervezési mintáknak és a keretprogramoknak [1.], [5.].

Természetesen csak akkor tudjuk a dinamikus öröklődést implementálni különféle programozási nyelvekben, ha az illető nyelv is fel van készítve erre. Sajnos a programozási nyelvek nagy többsége nem támogatja a dinamikus öröklődést, mint ahogyan nem támogatja a többszörös öröklődést sem. Számos programozási nyelv a dinamikus öröklődés helyett inkább a polimorfizmust és a futás alatti kötetést ajánlja.

A tisztán objektum-orientált paradigmára épülő nyelvek, mint például a Smalltalk, Self, CLOS, ilyen vagy olyan formában támogatnak egyfajta dinamikus öröklődést. C++-ban is megvalósítható egyfajta dinamikus öröklődés, template osztályok vagy különféle osztálygyárak használatával, amint ezt a [4.]-ben is láthatjuk, ugyanez megvalósítható Delphi-ben is vagy a COM, DCOM modellre épülő programozási nyelvek esetében is. Itt azonban inkább csak dinamikus példányosításról beszélhetünk, az objektumokat köthetjük futás idő alatt különböző osztályokhoz, természetesen így átvéve az adott osztály felmenő ágán az ősöket is.

### 3. Dinamikus öröklődés és az objektumorientált adatmodell

Az objektumorientált szemlélet hamar teret hódított az adatbázisok területén is, az objektumorientált adatmodell számára azonban nem alakultak ki egyértelmű szabványok.

Az adatmodell tulajdonságai a következők:

- objektumok és objektumok közötti relációkból épül fel
- valamilyen objektumorientált alapsnyelvre épül
- perszisztenciát használva az objektumokat adatbázisban tárolja
- lekérdezőnyelvvvel rendelkezik

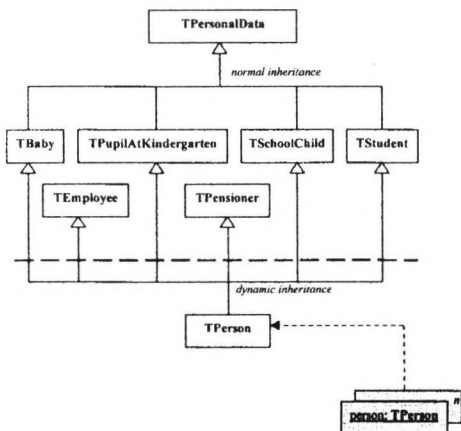
Az adatmodell objektumorientált aspektusokkal rendelkezik: osztályok, attribútumok, metódusok, integritás feltételek, objektum-azonosítók, egybezártág, öröklődés, polimorfizmus, perszisztencia és kiterjeszhetőség absztrakt adattípusok felé a főbb jellemvonásai. Tulajdonképpen kombinálja az OOP elemeket az adatbázisok jellemző tulajdonságaival úgy, hogy kibővíti egy, már meglévő, OOP nyelvet. Az objektumorientált adatbázisok hatékonyan tudnak tárolni bonyolultabb adatstruktúrákat is (fák, összetett objektumok, listák, kollekciónk stb.), és hatékonyan kezelik a dinamikus struktúrákat.

A dinamikus öröklődés kérdésköre is szorosan összefügg az objektumorientált adatmodellel, hisz olyan mechanizmusokat kínál, amelyek megkönnyítik az adatok (objektumok) adatbázisban való ábrázolását.

Gondoljunk csak arra az egyszerű példára, hogy személyekről szóló adatokat szeretnénk tárolni egy nyilvántartó rendszerben.

De a személyek is lényeges változásokon mennek át életükön keresztül. Először *csecsemők (baby)*, aztán *óvodások (pupil at kindergarten)*, majd *tanulók (school child)*, *diákok (student)*, *alkalmazottak (employee)*, végül *nyugdíjasok (pensioner)*.

Dinamikus öröklődést alkalmazva egyszerű és átlátható a fenti adatbázis megtervezése: létrehozzuk a megfelelő osztályokat és a *személyek* osztályt úgy építjük fel dinamikusan, hogy a példányai életciklusai során rendre a fenti osztályok valamelyikéből öröklődjön. Először *csecsemő* lesz, majd *óvodás*, aztán *tanuló*, *diák*, *alkalmazott*, végül *nyugdíjas*. A háttértárolón ábrázolt adatok közül is rendre a megfelelő osztályhoz tartozókat olvassa be, dolgozza fel, tárolja. Így egyetlen *személy* objektummal – ennek teljes életciklusán keresztül – bármilyen lekérdezést, vagy adatmanipulációt el tudunk végezni.



– Dinamikus öröklődés és az objektumorientált adatmodell –



A *TPersonalData* osztályt azért vezettük be, hogy tartalmazza az összes olyan adatot és metódust, amely normális módon átöröklődik minden egyes leszármazottra. Így a közös adatokat csak egy helyen deklaráljuk.

Az ODL (*Object Definition Language*) jelölési módját alkalmazva – ezt kibővítve a dinamikus öröklődés általunk bevezetett jelölési módjával – álljon itt formálisan is a fenti adatbázisnak egy egyszerű leírása:

```

interface TPersonalData
    (key id) {
        attribute string id;
        attribute integer age;
        attribute string name;
        attribute Struct Address
            (string zip_code, string city, string
             street) addr;
    };

interface TBaby: TPersonalData {
        attribute integer nappy_nr;
    };

interface TPupilAtKindergarten: TPersonalData {
        attribute string kindergarten;
    };

interface TSchoolChild: TPersonalData {
        attribute string school;
    };
interface TStudent: TPersonalData {
        attribute string university;
        attribute string faculty;
        attribute string department;
    };

interface TEmployee: TPersonalData {
        attribute string workplace;
        attribute string duty;
        attribute integer salary;
    };

interface TPensioner: TPersonalData {
        attribute integer pension;
    };

interface TPerson:
(TBaby if age in [0..2], TPupilAtKindergarten if age in
 [3..6],
TSchoolChild if age in [7..18], TStudent if age in [19..25],
TEmployee if age in [26..60], TPensioner if age > 60)
    (extent persons) {
        attribute List<TPerson> friends;
    };

```

## Könyvészet

- [1.] D. S. Batory, A. P. Buchman: *Molecular Objects, Abstract Data Types and Data Models: A Framework*, in: Proc. 10th VLDB, Singapore, 1984, pp. 172-184.
- [2.] H. Schönig and A. Sikeler: *Cluster Mechanisms Supporting the Dynamic Construction of Complex Objects*, in: Proc. 3rd Int. Conf. on Foundations of Data Organization and Algorithms FODO'89, LNCS 367, Paris, France, 1989, pp. 31-46.
- [3.] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorenson: *Object Oriented Modeling and Design*, Prentice Hall, Englewood Cliffs, NJ, 1991.
- [4.] James O. Coplien: *Advanced C++ Programming Styles and Idioms*, Addison Wesley, 1992.
- [5.] E. Gamma, R. Helm, R. Johnson, and J. Vlissides: *Design Patterns, Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading, MA, 1995.
- [6.] Kovács D. Lehel István: *Az objektumorientált paradigma*, Egyetemi Kiadó, BBTE, Kolozsvár, 2000.
- [7.] Lehel Kovács, Gábor Légrádi, Zoltán Csörnyei: *Purely Functional Programming and the Object-Oriented Inheritance and Polymorphism*, in: *Studia Universitatis Babeş-Bolyai – Informatica*, Vol. XLVI. 2001/1., BBTE, Kolozsvár, 2001, pp. 101.
- [8.] David E. Goldberg: *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley, 1989.

# **A mikroprocesszoros rendszerek megjelenésének hatása a vasúti irányítórendszerekre**

## **Effects of Microprocessor Based Systems on the Railway Signaling Technology**

**Dr. Héray Tibor**

Széchenyi István Egyetem, Győr

### **Abstract**

*The paper gives a short outline about the evolution of the railway signalling methods and practices starting with the fully human controlled systems and ending with the electronics ones, it also presents some probable solutions of the future. Because of the large spatial range of the railway technology both centralisation and decentralisation have the same reality in the developing work depending on the type of the traffic. The paper deals with both developing trends. Centralised fully electronic signalling systems will be probably the right solution in case of main railway lines with heavy traffic in spite of the decentralised control methods at the secondary lines.*

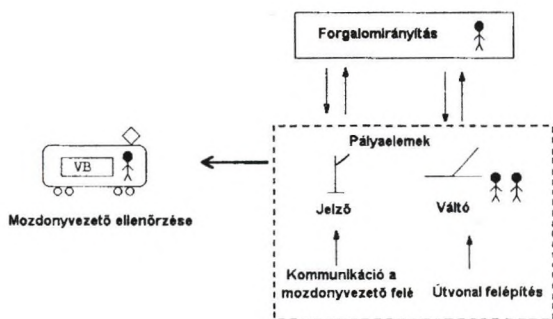
### **1. Vasúti automatikák fejlődése a mikroprocesszorok megjelenéséig**

Az elektronika, a számítógépek alkalmazásának rohamos terjedése a vasúti irányítás területét sem hagyja érintetlenül. Az elektronikus biztosítóberendezések megjelenése óta eltelt több mint egy évtized alaposan megváltoztatta a vasút automatizálásával kapcsolatos alapvető biztonságfilozófiai nézeteket. A technikai fejlődés itt is együtt jár a különböző szakmai területek között korábban érzett, és esetleg a valóságban is fennálló válaszfalak leomlásával, magával hozva a korábban viszonylag élesen elválasztható szakterületek egymáshoz közeledését, „konvergenciáját”. E divatos kifejezést először a távközlés és informatika szakterületére alkalmazták, de ma már látszik, hogy ugyanez a konvergencia jelenség figyelhető meg az egyes szakterületeken belül (1. távközlésen belül a vezetékes és vezeték nélküli technika megoldásait), s a szakterületek köre is bővül. A villamosmérnöki területen nemcsak a távközlés és informatika, hanem az automatizálás is csatlakozik ezekhez az egymáshoz konvergáló tudományos és mérnöki területekhez.



A vasúti technológia megvalósítása bonyolult ember-gép rendszerben történik, melynek gépi komponenseit a pálya-jármű-automatika hármass rendszer szoros kapcsolata jellemez. Ezen ember-gép rendszerben a fejlődés folyamán egyre növekszik a gépi komponensek száma, főleg a biztonsági feladatok megoldásában. A vasúti technika fejlődését kiváltó okok közül alapvetőnek az utasok helyváltoztatással kapcsolatos minőségi elvárásait, a nagy sebességet és rövid elérési időt tekinthetjük. Mindkét igény kielégíthetőségét a vonatbefolyásoló berendezések rohamos fejlődése, és európai szintű egységesítése (ETCS) teszi lehetővé. A vasúttársaságok ugyanakkor a folyamat megvalósítását minél kisebb ráfordításokkal kívánják megoldani, a vasút-irányítás fokozatos centralizációjával, illetve az irányításban résztvevő személyzet számának fokozatos csökkentésével. A fejlődés alapvető jellemzője tehát az ember-gép rendszeren belül a gépi komponensek számának fokozatos növekedése, melynek határa ma még nehezen körvonalazható.

Az elektronikus biztosítóberendezések megjelenése együtt jár a funkciók bővítésének lehetőségével. Az alapfeladatok a mechanikus biztosítóberendezések kialakulása óta változatlanok, de a feladatok megoldásának részletei bővülnek. Érdekes a vasúti folyamat klasszikus alapmodelljéből kiindulni, amelyből a fejlődés irányai világosan levezethetők (1. ábra).

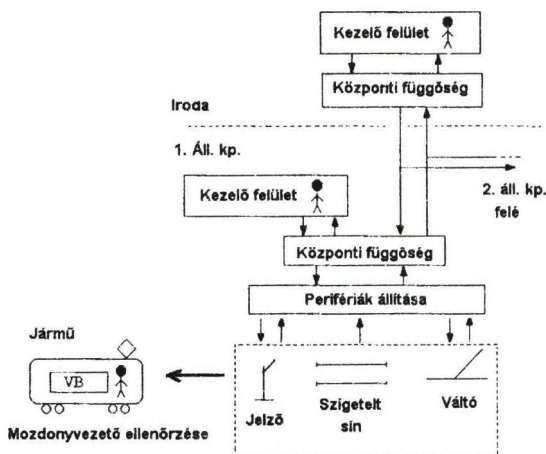


1. ábra

*Kulcsos berendezések struktúrája*

A kezdeti vasútirányítás szinte teljesen emberek által végrehajtott tevékenységek útján valósult meg: kézzel, a helyszínen állították be a váltókat a kívánt vágányútnak megfelelő állásba. A vágányút felépítése után szabadra állított (ekkor még elsősorban mechanikus működtetésű, ún. karos) jelzők útján értesítették a mozdonyvezetőt a továbbhaladás lehetőségéről. A mozdonyokon már ebben a kezdeti időben is jelen volt egy egyszerű készülék

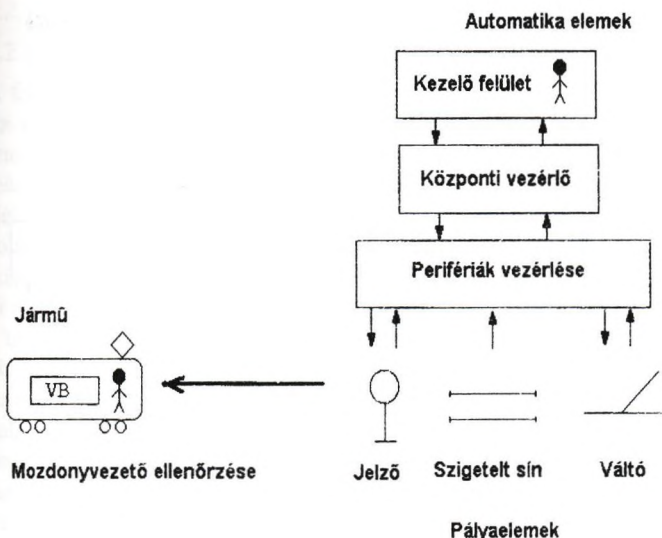
(az ún. „holt-ember” berendezés), ami a mozdonyvezető éberségét ellenőrizte. A fejlődés első lépéseként a helyszíni váltóállítás helyett bizonyos mértékig centralizált, a váltóközvet súlypontjába elhelyezett állító központból történő megoldást alkalmaztak, az állítás azonban még mindig az emberi energia felhasználásával valósult meg (2. ábra).



2. ábra  
Központi váltóállítás megjelenése  
(mechanikus biztosítóberendezések)

Megjelent a folyamatirányító berendezések szokásos struktúráját követő megoldás, amelynél a kezelői felület még speciális formában, több központba elosztott módon valósult meg, további jellegzetessége volt, hogy a váltóállító szerelvények egyben a vezérelt külsőtéri elem állapotát is jelezték (az állítódob, vonóvezeték és állító emeltyű kalibrált láncon keresztül megvalósított csúszásmentes mechanikus kényszerkapcsolata útján).

Következő lépésként a periférikus berendezéseknél fokozatosan villamos megoldásokat alkalmaztak: megjelentek a villamos váltóhajtóművek és fényjelzők, s ezek vezérlésére már jelzőgós biztosítóberendezéseket alkalmaztak. Ezzel megszűnt a kezelő szervek (kapcsoló-gomb vagy nyomógomb és vezérelt berendezések) mechanikus kényszerkapcsolata. Ennek következtében a kezelő kezelői felületen már szükségessé vált a külsőtéri elemek állapotának külön visszajelentése is (3. ábra).

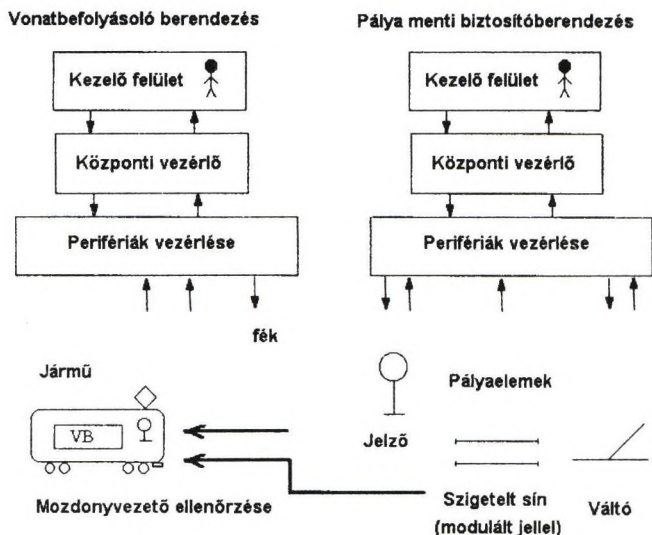


3. ábra

*Jelfogós biztosítóberendezések megjelenése*

Ebben a fejlődési fázisban rendkívül fontos feladat volt a vágányszakaszok foglaltságának gépi úton történő ellenőrzése - sínáramkörök vagy tengelyszámláló berendezések útján. A sínáramkör fontos szerepet játszott a pálya-jármű között szükséges információs kapcsolat kialakításában is. A mozdonyra történő jelátvitel megvalósíthatósága érdekében modulált sínáramköröket alkalmaznak. Ezzel a szokásos sebesség-jelzési rendszerek által megkívánt sebesség-információkat megfelelő módon fel lehet juttatni a mozdony vezető-állására. Tengelyszámláló berendezések alkalmazása vagy nagyobb mennyiségű információ feljuttatásának igénye esetén ilyen információs kapcsolatról külön kábelhurok lefektetésével kellett gondoskodni. Ezzel a klasszikus pálya mellé telepített állomási biztosítóberendezések mellett egyre nagyobb szerepet kaptak a járműre telepített ún. vonatbefolyásoló berendezések, amelyek fő célja a mozdonyvezető tévedéséből eredő veszélyeztetések minél teljesebb kizárása volt (4. ábra).





4. ábra  
 Vonatbefolyásoló berendezések megjelenése

## 2. A számítógépes megoldások hatása a vasútirányítás fejlődésére

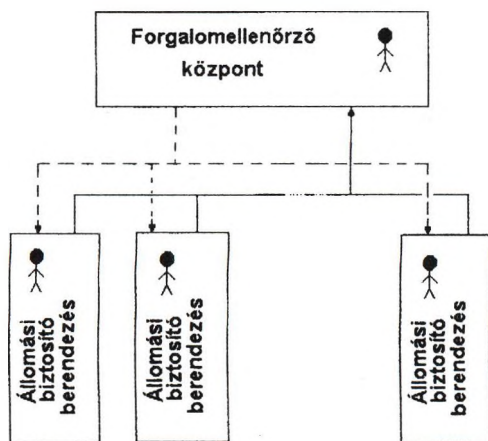
### 2.1. Általános megjegyzések

A 4. ábra mutatja a vasútirányításnak azt a modelljét, ami jól mutatja annak speciális jellegét: a teljes folyamat automatizálása a pálya-jármű komponensekkel kapcsolatos megosztott folyamatirányító rendszerekkel valósítható meg. A számítógépek alkalmazásának területét e két viszonylag független irányító rendszer központi egysége jelenti. Az elektronika, mikroelektronika rendkívüli fejlődése hozta magával a pálya menti telepítésre szánt elektronikus biztosítóberendezések, és a mozdonyra telepített ún. fedélzeti számítógépek megjelenését. S innen két szálon követhető a fejlődés: az elektronikus biztosítóberendezések vonalán az egyre inkább centralizált forgalomirányítás felé mutat a fejlődés, míg a fedélzeti számítógépek az egyre inkább decentralizált megoldást teszik lehetővé.

## 2.2. Fejlődés a centralizált irányítás felé

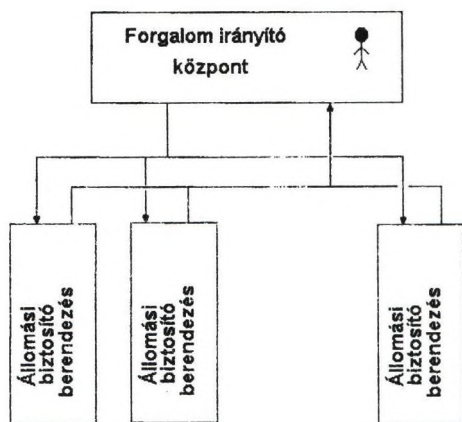
### 2.2.1. A centralizált forgalomirányítás hagyományos megoldási módja

A forgalomirányítás területén a centralizálási törekvések is hosszabb időszakra nyúlhatnak vissza. A vasúti közlekedés hálózati jellegéből adódóan az automatizálás szigorúan kötött térbeli alrendszerek kialakításához vezetett. Az állomási és állomásközi berendezések jelentették e térbeli hierarchia alsó szintjét. Már igen hamar kialakult az állomási forgalomirányító helyek összehangolásának igénye. Hosszabb vonalszakaszokon fekvő állomások forgalmi munkájának összehangolására már a kezdeti időszakban kialakult a menetirányítói rendszer, ami az állomásokkal parancs és visszajelentés irányban egyaránt távközlési kapcsolatokon keresztül tartotta a kapcsolatot. A forgalomirányítást kézzel vezetett segédeszközök támogatták (forgalmi napló). Ezek feladata kettős volt: a vonatok követése útján egyrészt lehetővé tette a forgalmi dolgozó számára a tényleges forgalmi helyzet megjelenítését, másrészt dokumentációul is szolgált a későbbi elemzések, kiértékelések számára. A mechanikus biztosítóberendezésekről jelfogósakra való áttérés megteremtette a távellenőrzés és távvezérlés lehetőségét. Első lépésben a visszajelentési irány gépi kapcsolata alakult ki, amivel a forgalom-ellenőrzés minősége vált javíthatóvá (központi forgalomellenőrző, ún. KÖFE rendszerek az 5. ábra szerint), majd sor került a parancs-irány gépesítésére is, és kialakultak a központi forgalomirányítók, az ún. KÖFI rendszerek (6. ábra).



5. ábra

A központi forgalomellenőrzés



6. ábra  
Központi forgalomirányító központ

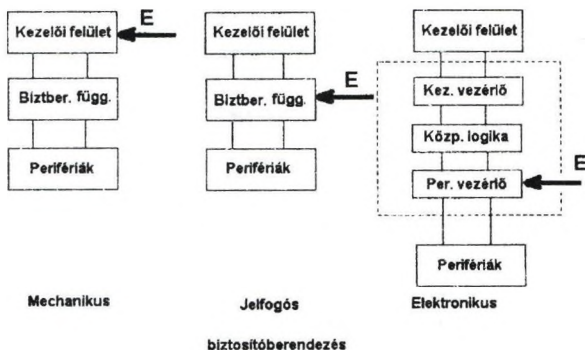
Az a KÖFI rendszer az állítási távolság növelését a biztosítóberendezésekre telepített felülvezérlő automatikával oldotta meg. A biztonsági feladatokat normál üzemi esetben a biztosítóberendezések teljesítették, a biztonsági feladatokkal nem rendelkező automatika kialakítására hamar megszülettek az elektronikus megoldások. Ezzel lényeges személyzet megtakarítás volt elérhető: megtörténhetett az állomási forgalmi szolgálattevők munkájának gépi helyettesítése. A menetirányítói rendszernél a forgalmi dolgozók közötti kommunikáció megvalósítására „külön-célú” távközlési berendezéseket alkalmaztak. A központi forgalom-ellenőrző központokba már gépi úton érkeztek az információk a vonal állomásairól, a menetirányítók tehermentesítésére pedig hamarosan megjelentek a forgalmi napló vezetését gépi úton támogató kiegészítő gépi berendezések: a vonatszám-jelentő, menetrendíró készülékek. A vonatszám-jelentő berendezések működtetéséhez általában szükséges volt a forgalomirányító személyzet egyszeri adatbeviteli tevékenysége, majd utána a biztosítóberendezésekből származó információk felhasználásával történt meg a vonatszámok továbbítása. A menetrendírók szintén biztosítóberendezési adatok felhasználásával végezték a tényleges út-idő diagramok gépi felvételét.

A forgalomirányítás e módjának feltétlenül gyengéje, hogy a központ működőképessége erősen függ az állomási alaptervezések jó működésétől. Kedvezőbb lenne az olyan megoldás, ami a pálya menti biztosítóberendezésektől függetlenül tudná begyűjteni a járművek helyére vonatkozó adatokat.



## 2.2.2. Az elektronikus biztosítóberendezések hatása a centralizálásra

A centralizált forgalomirányítás előzőekben ismertetett felülvezérlő automatikákkal történő klasszikus megoldásának létjogosultságát, szükségességét az adta meg, hogy az alapberendezések felépítése nem tette lehetővé a periféria-berendezések állítási távolságának közvetlen növelését. Ebből a szempontból a mechanikus biztosítóberendezéseknél maga a perifériális berendezések (a váltók és jelzők) állítása, jelfogó berendezéseknél pedig azok üzemszerű állapotának ellenőrzési lehetősége volt a kritikus tényező. Áttekintve a berendezések architektúrájának alakulását az állítási energia hozzávezetésének helye a következőképpen alakult (7. ábra).



7. ábra

*Állítási energia hozzávezetésének alakulása*

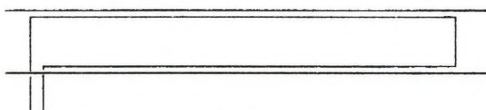
A mechanikus biztosítóberendezéseknél az energia hozzávezetés a kezelőfelületen keresztül történt, az állítási távolságot az ember fizikai terhelhetősége igen alacsony értéken korlátozta. Ezért ez a berendezés még többközpontos jellegű, az állító központnak a váltókörzet súlypontjában kellett lennie, s nagy váltókörzet esetén több állító központ között kellett megosztani az állítási feladatokat. A jelfogós biztosítóberendezések már egyközpontos jellegűek, ez volt a centralizáció első lépése. Mivel a korrekt végállás ellenőrzését végző áramkör a váltó felvágását a biztosíték kiolvadásával tudja maradandóan jelezni, ezért a lehetséges állítási távolságot a maximális kábel hurokellenállás szabja meg. A nagy rendszertechnikai változást az elektronikus biztosítóberendezéseknél a központi vezérlő logikának több mikroprocesszoros vezérlő síkra történő felbontása jelentette. Itt az egyes vezérlési síkok között az információs kapcsolat optikai kábelen valósítható

meg. A hajtóművekhez szükséges energia hozzávezetés rézkábeleire csak a periféria vezérlő sík és a perifériák között van szükség. Így a periféria vezérlő egység gyakorlatilag tetszőleges távolságra kerülhet magától a központi biztosítóberendezési logikát megvalósító számítógéptől, így az elektronikus biztosítóberendezések kezelési körzete műszakilag tetszőleges nagyságú lehet. A centralizálásnak a kezelők pszichológiai terhelhetősége fog csak határt szabni, figyelembe véve a nem automatizálható biztonsági felelősséggel végrehajtandó ún. „különleges kezeléseket”.

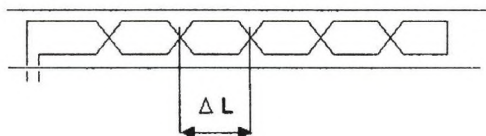
### 2.3. Fejlődés a decentralizálás irányába

A számítógépek megjelenése a jármű-vezérlés és -biztosítás területén szintén nagy változásokat hozott magával, ami a biztosítóberendezések ma megskott architektúrájának alapvető módosulási lehetőségét hordozza magában.

#### Sugárzó kábel keresztezés nélkül



#### Sugárzó kábel keresztezéssel



8. ábra

*Információs kapcsolat a mozdonyvezető felé*

A pálya-jármű közötti információs kapcsolat sugárzó kábeles kialakításával mód nyílt a jármű helymeghatározásának pontosítására, s ezzel lehetővé vált a pályajelzők nélküli közlekedés (8. ábra). A sugárzó kábel keresztezései révén igen sűrűn elhelyezett információs pontok realizálására nyílt lehetőség, ami a vonatok fékútnyi távolságban való közlekedése felé nyitotta meg az utat (az ún. villamos láttávolságra való közlekedés). A vonatbefolyásoló berendezések fejlődése és egységesítése (ETCS) a pálya-jármű közötti kapcsolat területén új technikai eszközök megjelenésével és meghonosodásával jár: a digitális mobil rádiós rendszerek fokozatos elterjedésé-



vel, és speciális vasúti alkalmazásával (GSM-R) kialakult a rádiós forgalomirányítás lehetősége. Ez a vasúti forgalomirányítás számára teljesen új műszaki lehetőséget teremt. Ennél a forgalomirányítási módszernél a járművek maguk végzik a helymeghatározás feladatát pályára orientált helymeghatározó elemek segítségével, de szóba jöhet műholdas helymeghatározás is (GPS). A pontos helyadatokat a jármű rádiós úton közli egy központba, melyek alapján a központ átveszi a vonatirányítás feladatát. A hagyományos térközi közlekedés helyett a vonatok utolérését ún. virtuális- vagy ún. mozgó térközi rendszer zárja ki [6]. A hagyományos pályaberendezések komponenseinek kiiktatásával megszűnik a pályához kötött (sináramkörös vagy tengelyszámláló) jármű-helymeghatározás. Figyelembe kell venni, hogy ezek a foglaltság-érzékelő elemek a járművek helymeghatározásán túl a vonatok integritás vizsgálatát is elvégezték - a vonatszakadást is foglaltságként tudták jelezni. Ezek elhagyásakor ezt a feladatot más úton kell megoldani a biztonságos közlekedés érdekében.

A rádiós technika vasúti alkalmazásával a vágányút-elemek rádiós állításának a lehetősége is megteremtődik - s ezzel a helyhez kötött biztosítóberendezések funkcióit, a vágányút-vezérlési feladatokat is a fedélzeti számítógép segítségével lehet megoldani. A rádiós forgalomirányítás esetén a vágányút-vezérlés központi megoldása helyett decentralizált módon történik a vágányutak felépítése, és a vágányúti elemek biztosítása. A rádiós forgalomirányító központból a vonat számára küldött menetengedéllyel együtt a vonatok hozzáférési jogosultságot kapnak az állomási vágányútban fekvő elemek vezérléséhez. A fedélzeti számítógépben tárolt vonal-atlaszban a vágányútban fekvő váltók helyzetére vonatkozó információ rendelkezésre áll. A váltókat egy decentralizált állító- és biztosítóberendezés segítségével lehet állítani, a mozdonyról megfelelő időben küldött rádió-jel alapján. A vezérlő berendezés állapotinformációt küld a váltókról a közeledő vonat számára (váltó-állás ill. lezárt helyzet). A rádiós forgalomirányítás esetén minden váltó egy egyéni veszélyeztetési pontot jelent, és így a menetengedély közvetlen célpontja lehet.

#### *2.4. A két fejlődési irány összevetése*

Az előzőekben vázolt két fejlesztési elképzelés természetesen nem egymást helyettesítő, csupán egymás mellett jelentkező lehetőséget jelent. Az elektronikus biztosítóberendezések által kínált centralizálási lehetőség ezeket a berendezéseket elsősorban a nagy forgalmú fővonalakon való használatra teszi alkalmassá. A kezelői körzet méretének rugalmas változtathatósága az az előnnyel jár, hogy ezek a berendezések akár egyetlen nagy állomás, akár egy hosszabb vonalszakasz kisebb állomásainak vezérlésére egyaránt



használhatók. A decentralizált forgalomirányítási megoldás elsősorban a kisforgalmú mellékvonalak automatizálására használható, s talán ez lesz a megoldás e vonalak forgalomirányításának gazdaságos megoldására.

### 3. Összefoglalás

Az előadás röviden felvázolja a vasútirányítás során alkalmazott módszerek, megoldások fejlődését a kezdeti teljesen emberi úton végrehajtott irányítástól az elektronikus biztosítóberendezések megjelenésével kialakult helyzetig, s kitekintést ad a következőkben várható, ma még csak körvonalazódó megoldások felé is. A vasút nagy térbeli kiterjedésű technológiai folyamat lévén a centralizáció és decentralizáció felé történő fejlesztésnek egyaránt meg van a realitása, a vasúti forgalom jellegétől függően lehet az egyik vagy másik irány a követendő. Az előadás foglalkozik mindkét fejlesztési irány célszerű alkalmazási területével. A nagy forgalmú fővonalakon az elektronikus biztosítóberendezések által nyújtott egyre nagyobb centralizáció, a mellékvonalakon inkább a decentralizált vezérlési megoldások alkalmazására kerülhet sor.

### 4. Irodalomjegyzék

- [1.] Fricke, H. - Pierick, K.: Verkehrssicherung, B.G. Teubner, Stuttgart - Leipzig, 1990.
- [2.] Héray Tibor dr.- Rózsa Gábor dr.: Vasúti biztonsági automatika-rendszerek funkcionalitásának fejlődése, SZIF Akadémiai Nap, 1998. szeptember
- [3.] Héray Tibor dr.: Korszerű vasúti automatikák biztonságelméleti kérdései, Kandó Kálmán Villamosipari Műszaki Főiskola Tudományos Ülésszak, 1995. június
- [4.] Héray Tibor: Elektronikus biztonsági rendszerek kialakításának rendszertехnikai kérdései, Kandó Kálmán Villamosipari Műszaki Főiskola Tudományos Ülésszak, 1998. május
- [5.] Pachl, J.: Anforderungen an die sicherheitsgerechte Visualisierung der Betriebslage; Signal und Draht (90)1998. 1+2; 5 pp.
- [6.] Pachl, J.: Systemtechnik des Schienenverkehr, B.G. Teubner, Stuttgart - Leipzig, 1999.

## Prediktor – korrektor és konzervatív integrátorok az égi mechanikában

### A korlátozott háromtest-probléma integrálása prediktor- korrektor és konzervatív integrátorral

Predictor – Corrector and Conservative Integration Methods  
in Celestial Mechanics

The Numerical Integration of the Restricted Three Body  
Problem with Predictor – Corrector and Conservative  
Integrators

**Kovács Barna**

Marosvásárhely, Papiu Ilarian Nemzeti Kollégium

#### Abstract

*The restricted three body problem (RTBP) is a classical, and partially resolved, problem of the celestial mechanics. The RTBP can be integrated only numerically. The conservativ integration method elaborated by B.A. Shadwick, W.F. Buell, and J.C. Bowman is applicable to the numerical integration of the RTBP. The accuracy of the conservative integration, applied to RTBP, opens a totally new vision in the later searching of the RTBP.*

A korlátozott háromtest probléma az égi mechanika egy klasszikus, és csak részben megoldott, feladata: nem integrálható, csak numerikus módszerekkel történő pályakövetés a megvalósítható. Az egy - és többlépéses módszerek mellett egy új numerikus integrálási módszer, a konzervatív integrátor használata – is alkalmazható. A B.A. Shadwick, W.F. Buell, és J.C. Bowman által kidolgozott integrálási módszer sikerrel alkalmazható a korlátozott háromtest-probléma (KHTP) numerikus integrálására. *Az integrálás pontossága maximális, így egy teljesen új szemléletet valósíthat meg a KHTP tanulmányozásában.* A dolgozat egy prediktor-korrektor integrátor eredményeit veti egybe a konzervatív integrátor eredményeivel.

**kulcsszavak** : *konzervatív integrálás, Adams módszerek, korlátozott háromtest - probléma*

#### 1. Prediktor – Korrektor módszerek

A korlátozott háromtest probléma modellezése differenciálegyenletekből álló rendszer numerikus megoldását feltételezi: köztudott, hogy a háromtest probléma, szemben a kéttest problémával, nem integrálható. A modellezés

eszközei a különböző hatákonyságú numerikus integrálási módszerek, illetve az ezekből kifejlesztett integrátorok. Általánosan két numerikus integrálási módszer használatos a korlátozott háromtest probléma numerikus integrálására, aszerint, hogy egy új csomópont kiszámítására milyen információk állnak a rendelkezésre:

- a) *egylépéses* módszerek, ide tartoznak a különböző rendű Runge - Kutta módszerek. Ebben az esetben az új csomópont kiszámítására csak egy csomópontot, az előzőt, használjuk fel, és
- b) *többlépéses* módszerek, ide tartoznak a különböző rendű Adams módszerek, de ismertek és használtak Nyström, Simpson, Milne többlépéses módszerei is. Ezeknél a módszereknél az új csomópont kiszámításánál figyelembe vesszünk bizonyos számú korábbi csomópontot is.

Adott egy  $n$  elsőrendű differenciálegyenletből álló rendszer, és adottak a kezdeti feltételek:

$$\begin{cases} \mathbf{y}' = f(x, \mathbf{y}) & \text{ahol } x \in [x_0, \beta] \\ \mathbf{y}(x_0) = \mathbf{y}_0 \end{cases} \quad (1)$$

amelyek együtt a *kezdetiérték feladat*ot adják. Feladatunk az, hogy egy ekvidisztáns rácson az  $\mathbf{y}(x_i)$ -t egy számított  $\mathbf{Y}_i$ -vel közelítsük:

$$\mathbf{Y}(x_i) = \mathbf{Y}_i \approx \mathbf{y}_i \quad (2)$$

A továbbiakban a többlépéses módszerek egy speciális változatát, a *prediktor - korrektor* módszert ismertetjük. A többlépéses módszerek adott  $s$  mellett  $s+1$  előző értéket, az  $\mathbf{Y}_{i-s}, \mathbf{Y}_{i-s+1}, \dots, \mathbf{Y}_{i-1}, \mathbf{Y}_i$  értékeket használnak fel az új, közelítet  $\mathbf{Y}_{i+1}$  kiszámítására

Az :

$$y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x)) dx \quad (4)$$

$$i = 0, \dots, N-1$$

egyenletben az  $f$ -et egy  $\Phi_s$   $s$ -ed rendű *interpolációs polinommal* helyettesítjük. Az így kapott egyenlet jobb oldala ha nem tartalmazza az  $x_{i+1}$  csomópontot, úgy a kapott számítási képlet *explicit számítási képlet*, ha az egyenlet jobb oldala tartalmazza  $x_{i+1}$ -et, akkor *implicit számítási képlet*ről beszélünk. Ha a két képlet egyszerre használjuk a közelítés kiszámítására, akkor a keletkező módszer egy *prediktor - korrektor* módszer: a *prediktor* az *explicit képlet* adja míg a *korrektort* az *implicit képlet* adja.

$s = 4$  Adams - Bashforth képlet - prediktor



$$Y_{i+1} = Y_i + \frac{h}{720} (1901f_i - 2774f_{i-1} + 2616f_{i-2} - 1274f_{i-3} + 251f_{i-4}) \quad (4)$$

$s = 4$  Adams - Moulton képlet - korrektor

$$Y_{i+1}^{v} = Y_i + \frac{h}{1440} (475f_{i+1}^v + 1427f_i - 798f_{i-1} + 482f_{i-2} - 173f_{i-3} + 27f_{i-4}) \quad (5)$$

## 2. A konzervatív integrálás alapjai

A Naprendszer mozgását leíró egyenletek konzervatív rendszert képeznek, mivel a súrlódás hiányában a rendszer *összeenergiája* megmarad. Egy pontos integrálási algoritmus ezt konzerválja a számítások során. A konzervatív integrálási módszer lényege, hogy az egymástól függő változókat egy olyan térbe transzformáljuk át, amelyben az energia lineáris függvénye a transzformált változóknak. A transzformáció után egy hagyományos numerikus módszerrel integráljuk a transzformált egyenleteket. Ezt az eljárást a következő lemma motiválja :

**1. Lemma** Legyen  $\mathbf{x}$  és  $\mathbf{c}$  az  $n$  két vektora. Ha az  $\mathbf{f} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$  függvény értékei ortogonálisak  $\mathbf{c}$ -re, úgy hogy  $\mathbf{I} = \mathbf{c} \cdot \mathbf{x}$  egy lineáris invariánsa a

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t) \quad (6)$$

differenciál - egyenletnek akkor a (6) differenciál- egyenlet

$$\mathbf{x}_j = \mathbf{x}_0 + \tau \sum_{k=0}^{j-1} b_{j,k} \mathbf{f}(\mathbf{x}_k, t + a_j \tau) \quad (7)$$

$m$  - edik diszkretizációja, ahol  $\tau$  az időnek megfelelő lépés és  $b_{j,k}$  szintén konzerválja  $\mathbf{I} - t$ .

$$\mathbf{c} \cdot \mathbf{x}_j = \mathbf{c} \cdot \mathbf{x}_0 + \tau \sum_{k=0}^{j-1} b_{j,k} \mathbf{c} \cdot \mathbf{f}(\mathbf{x}_k, t + a_j \tau) = \mathbf{c} \cdot \mathbf{x}_0 \quad (8)$$

Egy konzervatív integrálási algoritmus megírásához a (8) összefüggést használjuk fel. Legyen például a másodrendű prediktor - korrektor integrálási algoritmus :

$$\tilde{\mathbf{x}} = \mathbf{x}_0 + \tau \mathbf{f}(\mathbf{x}_0, t) \quad (9a)$$

$$\mathbf{x}(t + \tau) = \mathbf{x}_0 + \frac{\tau}{2} \left( \mathbf{f}(\mathbf{x}_0, t) + \mathbf{f}(\tilde{\mathbf{x}}, t + \tau) \right) \quad (9b)$$

A konzervatív prediktor – korrektor integrálási algoritmus megépítése során az  $\mathbf{x}$  változónak egy olyan  $\mathbf{T}(\mathbf{x})$  transzformációját keressük, amelynek segítségével az integrálás során állandó mennyiségek lineáris függvények formájában fejezhetőek ki az új  $\xi_i, i=1, \dots, n$ , változók által. A továbbiakban, megtartjuk a (9a)-t mint prediktort, a transzformált térben alkalmazzuk a:

$$\xi(t + \tau) = \xi_0 + \frac{\tau}{2} \left( \mathbf{T}'(\mathbf{x}_0) \mathbf{f}(\mathbf{x}_0, t) + \mathbf{T}'(\tilde{\mathbf{x}}) \mathbf{f}(\tilde{\mathbf{x}}, t + \tau) \right) \quad (10)$$

összefüggést mint korrektort, ahol  $\xi_0 = \mathbf{T}(\mathbf{x}_0)$ , és  $\mathbf{T}'$  a  $\mathbf{T}$  deriváltja.

Az új értéket az inverz transzformációval kapjuk meg:

$$\mathbf{x}(t + \tau) = \mathbf{T}^{-1}(\xi(t + \tau)) \quad (11)$$

Egy más értekezés témája lehetne a konzervatív és szimplektikus integrálási módszerek egybevetése, összehasonlítása. *Ge és Marsden(1988)* szerint egy integrátor pontos, ha szimplektikus és konzervatív is egyben. A két módszer általános hátránya az, hogy a *konzervatív integrálási módszerek esetében a fázistér térfogata nem szükségszerűen konzerválódik*, akárcsak az összenergia egy szimplektikus integrálás esetében. Az alkalmazott módszer kiválasztása az integrált rendszer struktúrájától függ.

### 3. A korlátozott háromtest –probléma (KHTP) integrálása konzervatív integrálási módszerrel

#### 3.1 Alapok

A korlátozott háromtest-problémáról számos kitűnő értekezésben olvashatunk.

A feladat:

*Adottak  $P_1$  és  $P_2$   $m_1$  és  $m_2$  tömegű testek, amelyek a közös tömegközéppont körül körmozgást végeznek. A mozgás síkjában az  $m_1$  illetve  $m_2$ -hez képest elhanyagolható tömegű  $P_3$  test található. A három test által meghatározott rendszerben csak a Newton féle kölcsönös gravitációs vonzóerők hatnak. Határozzuk meg a  $P_3$  által befutott pályája  $P_1$  és  $P_2$  síkjában.*

Az  $m_3$  mozgásegyenletei :

$$\begin{cases} \ddot{x} - 2\dot{y} = \frac{\partial \Omega}{\partial x}, \\ \dot{y} + 2\dot{x} = \frac{\partial \Omega}{\partial y}, \end{cases} \quad (12)$$

ahol :

$$\Omega = \frac{1}{2} \left[ (1-\mu)r_1^2 + \mu r_2^2 \right] + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \quad (13)$$

illetve

$$\begin{aligned} \mu &= \frac{m_1}{m_1 + m_2} \\ r_1 &= \sqrt{(x - \mu)^2 + y^2} \\ r_2 &= \sqrt{(x + 1 - \mu)^2 + y^2} \end{aligned} \quad (14)$$

Az integrálás pontosságát minden pillanatban a Jacobi - integrállal ellenőrizzük:

$$\dot{x}^2 + \dot{y}^2 = 2\Omega - C \quad (15)$$

ahol  $C$  a kezdeti feltételek mellé rendet állandó ( $x_0, y_0$  - az  $m_3$  kezdeti pozíciója a síkban, a 15-ben szerepel továbbá  $x$ , és  $y$  idő szerinti deriváltja). Az integrálás során a  $C$  értékétől való eltérés adja az integrálás pontosságának vagy pontatlanságának a mértékét. A  $C$  a továbbiakban a *Jacobi - konstans*.

Ha

$$\begin{aligned} q_1 &= x \\ q_2 &= y \\ p_1 &= \dot{x} \\ p_2 &= \dot{y} \end{aligned} \quad (16)$$

Akkor a Hamilton- függvény a következőképpen írható fel :

$$H = \frac{1}{2} (\dot{q}_1^2 + \dot{q}_2^2) - \frac{1}{2} (q_1^2 + q_2^2) - \left( \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \right) \quad (17)$$



### 3.2 A KHTP konzervatív integrálási algoritmus

A (16) rendszer prediktora :

$$\begin{aligned}\tilde{q}_i &= q_i + \dot{q}_i \tau, & i=1,2 \\ \tilde{p}_i &= p_i + \dot{p}_i \tau,\end{aligned}\quad (18)$$

Legyen :

$$\xi_1 = \frac{1}{2} q_1^2 \quad (19. a)$$

$$\xi_2 = \frac{1}{2} q_2^2 \quad (19. b)$$

$$\xi_3 = \frac{1}{2} \dot{q}_1^2 - \frac{1-\mu}{r_1} - \frac{\mu}{r_2} \quad (19. c)$$

$$\xi_4 = \frac{1}{2} \dot{q}_2^2 \quad (19. d)$$

A (19 a, b, c, d) összefüggéseket figyelembe véve kapjuk, hogy :

$$H = -\xi_1 - \xi_2 + \xi_3 + \xi_4 \quad (19.e)$$

Megállapítjuk, hogy (19.e) a  $\xi_i$ -nek lineáris függvénye. A (19-a) – (19-d) egyenleteket az idő szerint differenciálva kapjuk, hogy:

$$\dot{\xi}_1 = q_1 \dot{q}_1 \quad (20. a)$$

$$\dot{\xi}_2 = q_2 \dot{q}_2 \quad (20. b)$$

$$\dot{\xi}_4 = \dot{q}_2 \ddot{q}_2 = \dot{q}_2 (\dot{p}_2 - \dot{q}_1) \quad (20. c)$$

$$\dot{\xi}_3 = \dot{\xi}_1 + \dot{\xi}_2 - \dot{\xi}_4 \quad (20. d)$$

A korrektor :

$$\xi_i(t + \tau) = \xi_i + \frac{\tau}{2} \left( \dot{\xi}_i + \ddot{\xi}_i \right) \quad (21)$$

$i = 1, \dots, 4.$

A  $q_1, q_2, p_1$  és  $p_2$  értékei a következőképpen számítjuk ki :

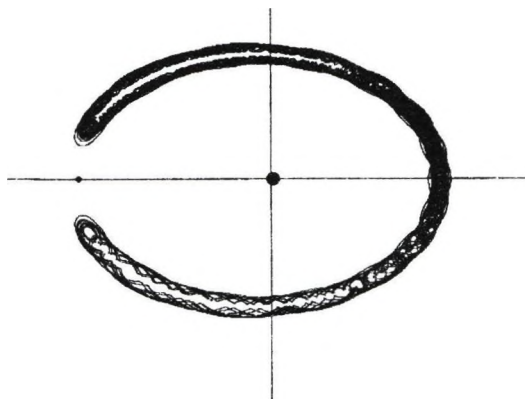
$$\begin{aligned}
 q_1 &= \operatorname{sgn}(\tilde{q}_1) \sqrt{2\xi_1} \\
 q_2 &= \operatorname{sgn}(\tilde{q}_2) \sqrt{2\xi_2} \\
 p_1 &= -q_2 + \operatorname{sgn}(\tilde{p}_1 + \tilde{q}_2) \sqrt{2\xi + \frac{2(1-\mu)}{r_1} + \frac{2\mu}{r_2}} \\
 p_2 &= q_1 + \operatorname{sgn}(\tilde{p}_2 - \tilde{q}_1) \sqrt{2\xi_4}
 \end{aligned}
 \tag{22}$$

### 3.3 Eredmények – Következtetések

A bemutatott két algoritmus alapján két számítógépes program készült, a C++ programozási nyelvet felhasználva. Az első RODOSZ konferencián bemutatott integrátor [5.] amely egylépéses integrációs algoritmussal határozta meg, és rajzolta ki, az  $m_3$  test  $m_1$  és  $m_2$  körüli pályáját, kevesebb ideig tette lehetővé a nyomkövetést. A prediktor-korrektor módszerek nagyságrendekkel növelték a számított pálya pontosságát:  $10^{-5}$ -ről  $10^{-12}$ -re, így hosszabb ideig lehetett egy-egy pályát megfigyelni.

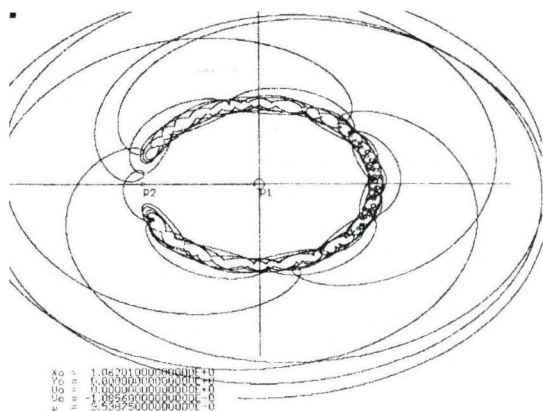
Igen nagy várakozásokkal tekintetem az új, konzervatív, integrátor által nyújtott eredmények elé. A (2.9) - (2.14) -gyel leírt algoritmussal elvégzett numerikus integrálás során a *Jacobi-konstans végig állandó maradt!* A *konzervatív integrátorral elért eredményeket figyelembe véve, megállapítható, hogy a konzervatív integrálási algoritmus egy új minőséget biztosít a KHTP további tanulmányozásában illetve, hogy a KHTP pályák további kutatását indokolt a konzervatív integrátorral folytatni.*

A mellékelt két ábra betekintést nyújt a *konzervatív integrátor* hatékonyságába: míg a *prediktor-korrektor* módszeren alapuló integrátor már kb. ezer év után nem bizonyul használhatónak, a konzervatív integrátor *9000 év után is pályán tartja* a kis testet. Figyelemre méltó, hogy a prediktor-korrektor integrálás tényleges pontossága minden „hibája mellett” is  $10^{-12}$  rendű...



1. ábra

*Nap, – Jupiter rendszerben mozgó elhanyagolható tömegű test pályája: a Jacobi-konstans értéke  $C = 3.00107500556513510$  az integrálás időtartama 9550 földi év. Az integrálás a konzervatív integrátorral történt*



2. ábra

*Nap, – Jupiter rendszerben mozgó elhanyagolható tömegű test pályája: a Jacobi-konstans kezdeti értéke  $C = 3.0010750055651351$ , végső értéke, az integrálás megszakításának pillanatában  $C = 3.00107500556530708$ . Az integrálás időtartama 1461 földi év. Az integrálás az Adams-Bashfort Adam Moulton prediktor - korrektor integrátorral történt*



#### 4. Könyvészet

- [1.] J.C. BOWMAN, B.A. SHADWICK AND P.J MORRISON, *Exactly conservative integrators*, in 15<sup>th</sup> IMACS World Congress on Scientific Computation Modelling and Applied Mathematics, Berlin, August 1997 .
- [2.] B.A. SHADWICK, J.C. BOWMAN AND P.J MORRISON, *Exactly conservative integrators*, SIAM J. Appl. Math, 59 (1999).
- [3.] V.G. SZEBEHELY, *Theory of Orbits: the Restricted Problem of Three Bodies*, Academic Press , 1967
- [4.] V. VÁLCOVICI, ȘT. BĂLAN, R. VOINEA :*Mecanică teoretică*. Editura Tehnică, București 1997
- [5.] KOVÁCS BARNA : Numerikus integrálási módszerek az égi mechanikában -- A korlátozott háromtest probléma numerikus integrálása - in RODOSZ -- tanulmányok II. Kriterion 2001

# Multimédiás adatátvitel teljesítményelemzése processz algebra segítségével

## Performance Modelling of the Multimedia Stream Controller Using Stochastic Process Algebra

Haller Piroska

Petru Maior Egyetem Marosvásárhely

### Abstract

*To develop a collaborative multimedia application we should create a middleware platform, that assures the group communication in heterogeneous system where the member's of the group have different quality requirements. Using stochastic process algebra we compare the different control strategies, and the effects of the temporal storage. The TIPP tools represent an approach to integrate qualitative analysis and performance evaluation for hierarchically connected distributed system.*

### Bevezető

Forgalom modellek és a forgalom mérések lehetővé teszik a forgalom előrejelzést illetve szabályzást. Osztott rendszerek esetén nem beszélünk folytonos adatfolyamokról. Adatfolyam aktív és szünet időszakok váltakozásából áll. Leírására szükségünk van a csúcs sebességre, vagyis aktív állapotban a forrás adási sebességére, az átlagsebességre, és az aktív állapot átlagos időtartamára.

Az on-off modell a forrást egy aktív és szünet periódusok váltakozó sorozatával jellemzi, ahol a különböző időszakok függetlenek egymástól és exponenciális eloszlásúak. Markov modellt használva a forrás folyamatos idejű m állapotú Markov lánc exponenciális eloszlású tartózkodási időikkel.

Fogadó oldalról nézve az adatok csomagok formájában érkeznek, melyek előre megadott időbeli megszorításoknak kell megfeleljenek. Ezek a megszorítások vonatkozhatnak egy adatfolyam csomagjaira, vagy különböző adatfolyamok szinkronizációjára. Ezen relációkat bizonyos intervallumra korlátozhatjuk, vagy bizonyos csomagokra, és lehetnek szabályosak vagy egyediek, illetve más eseményekhez kötöttek.

A megszorítások változtatása mindig a felhasználó oldalról történik és a platform szerepe, hogy ezeket lebontsa minden szintre (forrás, átviteli közeg, elosztó szabályzó objektumok). Ezen algoritmusok meghatározásában

fontos szerepe volt a modelleknek, mely lehetővé tette a különböző paraméterek hatásának a vizsgálatát. A rendszer paramétereinek időbeli változását egy központi szabályzó rendszer gyűjti össze az osztott monitorizáló objektumoktól.

Osztott multimédiás rendszerek esetén a formális ellenőrzésnek minőségi és mennyiségi szempontokat is figyelembe kell vennie. Ugyanakkor az időbeli megszorítások a rendszer bármely pontján dinamikusan változhatnak (újrakonfigurálható modulok), tehát a leírásoknak ezt is figyelembe kell venniük.

A felállított modell alapján elvégezzük a rendszer működésének a vizsgálatát beleértve a szinkronizációs feltételek ellenőrzését és a teljesítményének az elemzését.

Multimédiás rendszerek esetén több formális technika ötvözése szükséges, mert az időbeli követelmények leírása különbözik az adatátvitel, adatmegjelenítés illetve a forgatókönyvek esetén. Jelen dolgozat csak az adatátvitel modellezésével foglalkozik.

Ezen a szinten a rendszert az átvitel minőségi paraméterei jellemzik, melyeket megengedett intervallumok formájában adunk meg:

- átviteli ráta - az adatok átviteli sebességét jellemzi, de a csomagméret is befolyásolja
- a késés esetén meg kell különböztetnünk a forrás okozta illetve az átviteli közeg okozta késéseket
- a késés eltérését bizonyos határok között kell tartani, hogy az adatok szinkron megjelenítése biztosítható legyen
- a hibaszázalék alacsony szinten tartása szükséges, mert nem megengedett a multimédiás csomagok újraküldése.

A kért minőségi paraméterek biztosítása a platform feladata, mely egy központi paraméter kezelő egység és osztott szabályzó illetve osztott monitorizáló egységek segítségével valósítható meg. A szabályzó objektumok szerepe a paraméterek folyamatos ellenőrzése, az időbeli korlátozások biztosítása különböző adaptív stratégiák segítségével, illetve ezen stratégiák módosítása vagy újrakonfigurálása.

## Modellezés

A modellezésnek és matematikai ellenőrzésnek fontos szerepe van az osztott rendszerek megvalósításában ugyanakkor ismerni kell alapvető korlátait is. A feladatok valóság-hű modellezése nagy modellméretet von maga után. Másrészt ezek a rendszerek idővariánsak és nemlineárisak ezért az idődimenzió kezelése rendkívül nehézkes. Nem utolsó sorban modellezni kell a rendszer futtatási környezetét és a pillanatnyilag rendelkezésre álló



erőforrásokat is. Az előbbieket figyelembe vételével csak azok a formális nyelvek elfogadhatóak, melyek lehetővé teszik a modell hierarchikus felépítését funkcionálisan önálló rendszerekből, melyek külön is tesztelhetők. Osztott rendszerek esetén az egyes részek közötti kölcsönhatásnak modellezése és tesztelése ugyanolyan fontos.

Processz algebraik lehetővé teszik konkurens és osztott rendszerek leírását és funkcionális ellenőrzését. A sztochasztikus kiterjesztés mint a PEPA vagy TIPP esetén a rendszer paramétereinek mennyiségi vizsgálata illetve teljesítményelemzés is végezhető.

Kiszolgálási rendszerek formális modelljének a használata lehetővé teszi egyes szolgáltatói (erőforrás-szükséglet, költség, kihasználtság, átvitel, bevétel) és felhasználói (válaszidő, veszteség, ár, korrektség) szempontok elemzését anélkül, hogy a rendszer belső állapotait figyelembe venné. Nagyon hatékony módszer a hierarchikusan lebontott rendszerek tanulmányozására. A felállított alap-összefüggések egyaránt használhatóak az alrendszerek és az összetett rendszer teljesítmény mérésére és ezek az összefüggések csak a megfigyelhető paraméterekre alapoznak (átlagosan beérkező üzenetek száma, átlagosan kiszolgált üzenetek száma, foglaltsági periódus, a rendszerben tartózkodó üzenetek száma, megfigyelési periódus). A multimédiás adatátvitel csoportkommunikáció esetén szabályzó objektumok segítségével történik. Minden adat típus esetén a megfelelő szabályzó beolvassa az adatokat a csoport tagjaitól és szétküldi azokat a csoport tagjainak a központi szabályzó által előírt átviteli ráták betartásával. Ezen szabályzó belső viselkedésének tanulmányozására Markovi modell felállítására van szükség. A felállított állapotgráf segítségével lehetőség nyílik a belső módosítások hatásának a figyelembe vételére.

### Sztochasztikus folyamatok

Sztochasztikus folyamatot az  $\{X(t)\}$  állapotváltozók sokassága alkotja, ahol maga az  $X(t)$  függvény a valószínűségi változó. A  $t$  változó rendszerint az idő, de lehet más is. pl. hely, stb. Stacionárius folyamat az a sztochasztikus folyamat, amelyben az  $\{X(t)\}$  sokasság statisztikai tulajdonságai az időtől függetlenek. A sokasság  $t$  idő pillanatbeli  $E[X(t)]$  várható értéke minden időpillanatban azonos, két időpillanatbeli értékek kovarianciája csak az időkülönbségtől függ. Ergodikus folyamat olyan stacionárius folyamat, amelyben a sokasság szerinti várható érték megegyezik a sokasság egyetlen realizációjából kiszámított átlagértékkel. Markov folyamat esetén a folyamat jövőbeli alakulása közvetlenül csak a jelenlegi állapotától függ, azaz a múlt csak a jelenen keresztül befolyásolja a jövőt.

A rendszer analitikus megoldása érdekében a következő tulajdonságokat kell teljesítenie: Markov típusú, stacionárius, irreducibilis vagyis bármely állapot elérhető bármelyikből, és homogén vagyis nem függ a megfigyelés időpontjától. A diszkrét idejű homogén Markov lánc az átmenetvalószínűség-mátrix megadásával írható le.

A Markov láncok tranziens viselkedésének vizsgálatában gyakori feladat az „elnyelő állapot” várható elérési idejének kiszámítása. Egy ML elnyelő állapota az állapot, ahová a lánc előbb-utóbb elér, bárholnan is indítottuk.

A teljesítményelemzés szempontjából a megfigyelési periódus jóval nagyobb mint az egyedi tranziens idők. Így a kezdeti állapot megválasztása nem befolyásolja a stacionárius valószínűség-eloszlást. Stacionárius állapotban az átmenetvalószínűség-mátrix megmutatja, hogy egy folyamat mennyi időt tölt a  $k$  állapotban, mert a rendszer egyensúlyi állapotából következően a bemeneti fluxus egyenlő a kimeneti fluxussal.

Ezek alapján kiszámítható egy erőforrás kihasználtsága mint azon állapotok valószínűségének összege, mely teljesít egy adott feltételt, vagy az átlagos várakozási idők, illetve a rendszerben várakozó adatok száma. Illetve megjósolhatjuk egy esemény megjelenésének a gyakoriságát, mint az esemény rátájának és annak az állapotnak melyben megjelenhet a valószínűségének szorzata. Ezek után alkalmazzuk a kiszolgáló rendszerek alapösszefüggéseit a rendszer külső paramétereinek becslésére.

## TIPP

TIPP környezet sztochasztikus processz algebrán alapul, lehetővé téve a modell funkcionális ellenőrzését, és stacionárius illetve tranziens elemzését. A modell leírása folyamatok és események segítségével történik. A nyelv lehetővé teszi a folyamatok összetevését, a folyamatok környezetének megadását, parametrikus folyamatok létrehozását, valamint a folyamatok közötti kommunikációt. A kommunikációs események lehetnek azonnaliak, vagy hozzárendelhetünk egy eloszlásfüggvényt (exponenciális vagy Erlang).

Operátorok:

- stop – folyamat nem fogad több eseményt
- $(a, \lambda); P$  – az a esemény megelőzi a  $P$  folyamatot, az a esemény átlagos beérkezési ideje  $\lambda$ .
- $a; P$  – az a azonnali esemény megelőzi a  $P$  folyamatot
- $P \gg Q$  – A  $P$  és  $Q$  folyamatok szekvenciális végrehajtása
- $P[>Q$  – bármely esemény a  $Q$ -ból megszakítja a  $P$ -t (preemptív)
- $P[|Q$  – választás, az első esemény ami bekövetkezik végrehajtható függetlenül, hogy melyik folyamathoz tartozik (az azonnali események dominálják a rendszert)



- P||Q – párhuzamos végrehajtás
- P[[a1, a2]]Q – párhuzamos végrehajtás szinkronizációval az a1 és a2 eseményekre
- hide a1, a2 in P – a1 és a2 események nem láthatóak a P folyamaton kívül
- P[a1, a2, a3](r1, r2, r3) – parametrikus folyamat meghatározása

A rendszer leírása az egyes folyamatok hierarchikus összetevését jelenti, az elrejtési operátor pedig lehetővé teszi a modell többszintes ellenőrzését ami kedvező a számítási komplexitás szempontjából.

A modell leírása után következik annak funkcionális elemzése( ellentmondás-mentesség, holtpont, teljesség), a modell ekvivalens átalakítása és a hozzátartozó Markov modell létrehozása, tranziens és stacionárius elemzése.

### A videó-folyam szabályzók teljesítményelemzése

A videó-folyam természetéből adódóan a szétosztó egységek a kimeneti ráta szabályzását csomagok eldobásával érik el. Ennek érdekében különbséget kell tenniük az egyes csomagtípusok között. A 0 típusú csomagok (audió, I-frame...) feltétel nélkül továbbíthatódnak, míg az 1 típusú csomagok közül eldob annyit, hogy a kimeneti ráta értéke állandó maradjon. A modell feltételezi, hogy a 0 és az 1-es típusú csomagok aránya állandó és független a bemeneti rátától. A modell elemzése során kiderült, hogy egy közös kimenő puffer alkalmazása biztosítja a két típusú csomag szinkronizációját, a feldolgozási késésekből adódóan a nagyobb prioritású csomagok nem voltak elküldve (előregedtek) annak ellenére hogy számukra lett volna elég sáv. A következőkben egy lehetséges megoldást mutatunk be több kimenő puffer használatával, de a csomagok szinkronizációja megmarad.

```

specification System
behaviour
Source0 [[transmit]] Dim0 [[inc1,inc2]] (TR1(0) ||| TR2(0) ||| Timer(3))
[[reset,set]] Decide(0,3) [[write0, write1]] (Sink0 ||| Sink1)
where
process Source0 :=
    (transmit, rtr0); Source0
endproc
process Dim0 :=
    [0.2] ((transmit,1); inc1; Dim0)
    [0.8] ((transmit,1); inc2; Dim0)
endproc
process Sink0 :=

```



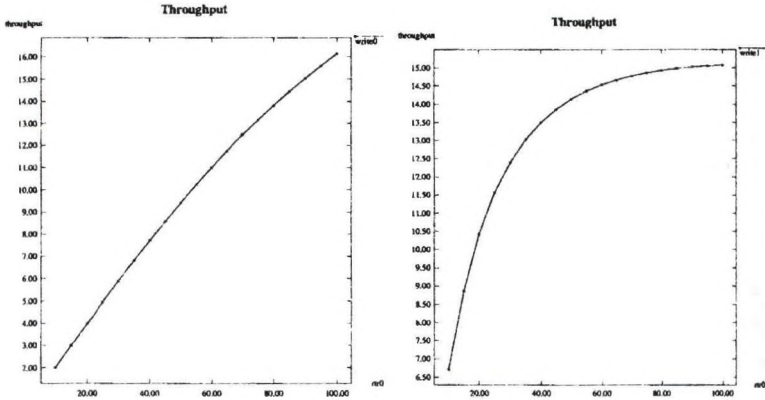
```

(write0, rwr0); Sink0
endproc
process Sink1 :=
(write1, rwr0); Sink1
endproc
process TR1(n) :=
[n < 5] -> inc1; TR1(n+1) []
[n > 0] -> set!1; TR1(n-1) []
[n = 5] -> loss0; TR1(n)
endproc
process TR2(n) :=
[n < 5] -> inc2; TR2(n+1) []
[n > 0] -> set!2; TR2(n-1) []
[n = 5] -> loss1; TR2(n)
endproc

process Decide(n,NR) :=
set?x:int; Count(x,n,NR) []
reset; Decide(0,3)
where
process Count(x,n,NR) :=
[x = 1 and NR > 0] -> (write0,1); Decide(n,NR-1) []
[x = 2 and n+1 < NR] -> (write1,1); Decide(n+1,NR) []
[NR = 0 or (x = 2 and n+1 > NR-1)] -> loss; Decide(NR,NR)
endproc
endproc
process Timer(n):=
[n > 0] -> EXP(tick); Timer(n-1) []
[n = 0] -> reset; Timer(3)
endproc
endspec

```

A továbbiakban bemutatjuk a két típusú csomag kimeneti rátájának változását ha a feldolgozási sebesség egy felhasználó esetén 100 cs/s, a kimenő előírt ráta 30 cs/s, a bemeneti ráta változik és a két típusú csomag arány 2:8.



1. A kimeneti ráták változása

Ha az adatok szinkronizációja nem fontos de nem megengedett a kisebb prioritású csomagok kiküldése amíg vannak nagyobb prioritású csomagok módosíthatjuk a modellt a következőképpen:

```

process TR1(n) :=
  [n < 3] -> inc1; TR1(n+1) []
  [n > 0] -> set!1; TR1(n-1) []
  [n = 0] -> last; TR1(n) []
  [n = 3] -> inc1; TR1(n)
endproc
process T2(p,n):=
  no; TR2(p,n) []
  last; TR2(1,n) []
  reset; TR2(0,n)
where
process TR2(p,n) :=
  [n < 5] -> inc2; T2(p,n+1) []
  [n > 0 and p = 1] -> set!2; T2(p,n-1) []
  [n = 5] -> inc2; T2(p,n)
endproc
endproc

```

### Következtetések

Tekintetbe véve az osztott multimédiás rendszerek nagyon változó minőségi követelményeit egy hierarchikus szabályzórendszer biztosítja az előírt átviteli rátákat. A szabályzórendszer tervezéséhez szükség van egy olyan

környezetre mely lehetővé teszi a rendszer modellezését és teljesítményelemzését. A bemutatott módszerek biztosítják e rendszer globális paramétereinek és egyes szabályzók belső viselkedésének elemzését egyaránt. Vizsgáltuk a komponensek terheltségét, válaszidejét meghatározva a szűk keresztmetszeteket és a szabályzók újrendezési algoritmusát.

A TIPP környezet ugyanakkor lehetővé tette a szabályzók stacionárius és tranzien állapotainak elemzését. Vizsgáltuk a pufferek méretének, a szinkron illetve aszinkron felfolgozási módnak, a választott szabályzási algoritmusnak, a különböző átviteli rátáknak a hatását a rendszerre.

Az összehasonlító tanulmány bebizonyította, hogy a különböző időmegszorítások megvalósítása csak különböző szabályzók segítségével lehetséges, ami a belső pufferek számát, méretét, prioritását illetve feldolgozó szálak számát illeti. A szabályzók hierarchikus kapcsolása pedig lehetővé teszi a felhasználók rendkívül különböző követelményeinek betartását.

### Könyvészet

- [1.] M. Bernardo: Theory and Application of Extended Markovian Process Algebra, PhD thesis, Università di Bologna, Padova, Venezia, 1999
- [2.] Ch. Chou, L. Golubchik, J. C.S. Lui, A Performance Study of Dynamic Replication Techniques in Continuous Media Servers, Technical Report CS-TR 3948 Computing Laboratory University of Kent at Canterbury, 1998
- [3.] S. Gilmore, J. Hillston, L. Kloul, and M. Ribaud. Software performance modelling using PEPA nets. In Proceedings of the Fourth International Workshop on Software and Performance, pages 13-24, Redwood Shores, California, USA, January 2004. ACM Press.
- [4.] P. Haller: Contributii la implementarea sistemelor multimedia distribuite, PhD thesis, Universitatea Tehnica Cluj Napoca, 2002.
- [5.] H. Hermanns: Interactive Markov Chains, PhD thesis, Der Technischen Fakultät der Universität Erlangen-Nürnberg, 1998.
- [6.] J. Hillston: Lectures on Formal Methods and Performance Analysis, E. Brinksma, H. Hermanns and J-P. Katoen (editors), LNCS 2090, Springer-Verlag, 2001
- [7.] U. Klehmet, V. Mertsotakis: TIPPtool User's Guide, Technical Report, Universität Erlangen-Nürnberg, IMMD 7, 1998.
- [8.] Z. Zhang, S. Nelakuditi, R. Aggarwal, R. Tsang: Efficient Selective Frame Discard Algorithms for Stored Video Delivery across Resource Constrained Networks, Dept. of Computer Science & Engineering Sandia National Laboratories University of Minnesota, 1999



# Tűzfalak – működési elvek, tervezésük és működtetésük egy oktatási hálózatban

Firewalls – how they work, how they are built

Budai László

Sapientia EMTE, Műszaki és Humántudományok Kar,  
Marosvásárhely

## Abstract

*Computer networks are meant to permit data exchange among computers. The possibility of accessing any other computer in the network is a blessing only if there is a trust relation among them. But this is not the case today, when large corporate networks need to connect to the Internet where millions of other users are connected too. This is why private networks need to be protected by firewalls.*

*In this paper I present the principles that firewalls rely on, the difference between packet filtering and proxy servers. There are also presented the principles that one should consider when building security policies for firewalls. Also the paper presents open-source solutions that can be used in low-budget educational institutions.*

A tűzfal egy olyan szerkezet amely meggátolja a tűz terjedését. Ilyen szerkezetek találhatók épületeknél amikor az egyes részek egy téglából épült fallal vannak elválasztva, az autóknaál ahol a motorház van egy fém fallal elválasztva az utastértől. A számítógép hálózatoknál a tűzfal fő szerepe az, hogy megvédje a belső hálózatot az internet „tüzétől”.

A számítógép hálózatok robbanásszerű elterjedésével egyre több kritikus/bizalmas információ található meg elektronikus formában az intézmények, cégek belső hálózataikban. Nagyon sok esetben bizonyos információk csak elektronikus formában léteznek, ezért ezek fokozott védelmet igényelnek. Ez a védelem kiterjed az állományok védelmétől (titkosítás, biztonsági mentések) a hálózat védelméig. A tűzfalak a belső hálózatok védelmét próbálják biztosítani az internet tüze ellen, illetve a belső hálózat felhasználóinak a „tisztaságát” őrzik, megtiltva hozzáférést az internet kísértéseihez. :-)

Egy egyszerű tűzfalat megvalósíthatunk egy, két hálózati kártyával rendelkező Unix/Linux géppel. Ennek az egyik hálózati kártyája a belső hálózatra, míg a másik az internetre kapcsolódik. A két kártya között nincs routolás (csomag továbbítás), a felhasználók ha egy internet szolgáltatást akarnak elérni, akkor először erre a gépre bejelentkeznek és innen indítják a

különböző alkalmazásokat (email kliens, böngésző). Az adatokat lementhetnék először erre a gépre, és csak azután másolódna a belső hálózatban levőkre. Ezt a változatot meg lehet tüzdelni egy X szerverrel és a felhasználók X klienseken keresztül kapcsolódna rá. Így akár a saját gépeiket használva is grafikus internet hozzáférésük volna (X kliens).

Habár első pillantásra jónak néz ki, ezt az eljárást mégsem ajánlom senkinek sem mivelhogy a betörések túlnyomó része úgy történik, hogy a támadó bejelentkezési jogot (kontót) szerez a megtámadott gépen.

Manapság a tűzfalak megvalósítására kétféle eljárás van elterjedve:

- proxy szerverek
- csomag szűrés

### **Proxy szerverek**

A proxy szerverek olyanok mint a rádió átváltók. A két kommunikáló fél között vannak elhelyezve, az adatfolyam rajtuk keresztül megy, úgy hogy a kapcsolatot kezdeményező fél egyenesen a proxyval tárgyal (tudatosan vagy a tudomása nélkül az „átlátszó” proxy esetén). A kérés a a proxyhoz jut amely továbbítja a szolgáltató félhez (szerverhez). A válasz szintén a proxyon keresztül jut a klienshez. Az egész úgy működik mintha ahhoz hogy bejelentkezhessünk egy a helyi hálózaton kívül levő gépre először be kell jelentkezzünk a lokális háló és a külső hálózat között levő proxyra, majd innen lehet csak tovább menni a tulajdonképpeni cél felé.

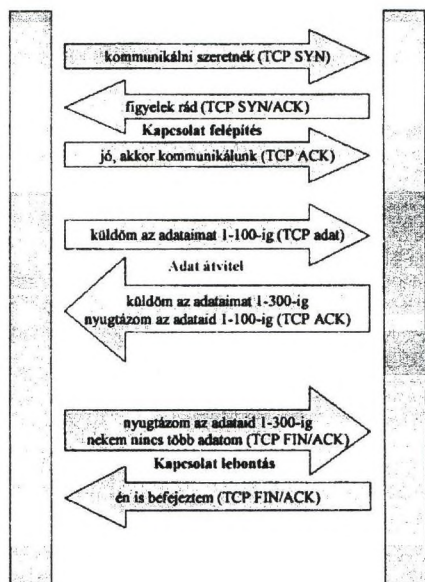
Mivel az egész kifelé történő kommunikációt a proxy végzi, megvan a lehetőség, hogy egy naplót vezessünk erről (lóg állomány). Ugyanakkor a proxy szerverek megadják a lehetőséget, hogy az internet hozzáférés csak egy azonosítás (autentifikálás) után legyen engedélyezett, esetleg a felhasználóknak különböző szolgáltatásokhoz biztosíthatunk hozzáférést.

Egy proxy szerverre példa a squid alkalmazás amelynek alapfeladata a http kérések „átjártása” és ugyanakkor az érkező válaszok kéznél tartása (cache), ha újabb kérés lesz ugyanarra az erőforrásra (weblapra), akkor a squid a helyi adatbázisából fogja kiszolgálni a kérést, ami egy lassú kapcsolat esetén lényegesen jobb válaszidőt eredményez.

### **Csomagszűrés**

A csomagszűrés egy másik módja a tűzfal készítésnek. Ebben az esetben a csomagokat a routerben szűrjük a bizonyos tulajdonságaik függvényében. Ahhoz hogy felismerjük a csomagszűrés által nyújtott lehetőségeket először meg kell nézzük hogyan megy végbe a kommunikáció két számítógép között. Az interneten az adatok IP datagramok formájában közlekednek. Minden IP datagram egy független teljesen leírt adatsomag, amely tartalmazza a forrás (küldő) és a cél címét. A forrás és a cél közötti úton a datagram egyik

routertől a másikhoz lesz továbbítva. Egy tűzfal nélküli felépítésben a datagramok zavartalanul eljutnak a forrástól a célállomáshoz. Ahhoz hogy a gépek között párbeszéd alakuljon ki (pl. egy email elküldése, egy állomány átvitele) az IP datagramok sorozatokba vannak fogva (csoportosítva) amelyeket a TCP (Transmission Control Protocol – átvittelt vezérlő protokoll) vezérel. A TCP protokoll szerint egy adatátvitel a következőképpen zajlik le:



Az adatátvitel részben a TCP/IP protokoll fejléce után következő adatok fogják tartalmazni egy magasabb szint protokollját (amely pl. a tulajdonképpeni levélküldéssel foglalkozik). Létezik egy a TCP-nél egyszerűbb szállítási protokoll: UDP (User Datagram Protocol). Ez egy kapcsolat nélküli protokoll amelyben az adatokat elküldi a forrás, de ezeknek a megérkezésükről a protokoll nem mond semmit. Ebben az esetben nem létezik a kapcsolat felépítési fázis, illetve a lebontás sem. Ahhoz hogy a kliens a szerveren a megfelelő alkalmazáshoz kapcsolódjon, mindkét protokoll esetén (TCP, UDP) be lett vezetve a „port” fogalma, amely egy gépen belül címezi meg az alkalmazásokat.

A csomagszűrés a routeren átmenő csomagok tulajdonságai függvényében eldönti mi legyen a csomag sorsa. A tűzfal szempontjából igazán három féle dolgot végezhetünk a csomaggal:



- elfogadjuk a csomagot – ebben az esetben a csomag tovább lesz küldve a cél felé.
- eldobjuk a csomagot – ebben az esetben a csomag el lesz dobva, mintha meg sem érkezett volna. A küldő nem lesz értesítve.
- visszautasítjuk a csomagot, de értesítjük a faladót.

Azok a tulajdonságok amelyek szerint a csomagokat szűrni lehet általában a csomag fejlécében vannak, ilyen a forrás/cél cím, forrás/cél port, a TCP illetve EP protokollok jelzőbitjei (SYN, ACK, FIN, TOS, stb.).

A csomagszűrés továbbfejlesztett változata az *állapot kivizsgálás* (stateful inspection). Ebben az esetben a csomagsorozatok figyelésével különböző következtetéseket tudunk levonni a kapcsolat állapotáról (most induló, már létrejött, rokon kapcsolat) és ennek függvényében tudunk dönteni a bizonyos csomagok sorsa felett.

Egy másik a csomagszűréssel kapcsolatos eljárás a csomagok fejlécének a megváltoztatása. Habár megjelenésének az oka az IP címek számának a korlátja volt, a NAT (Network Address Translation) egy jól használható eljárás arra, hogy a belső hálózatot „eldugjuk” egy publikus IP cím mögé, amely azon routeré aki végzi a NAT-ot. Ez a megoldás magában nem helyettesíthet egy csomagszűrőes vagy proxys tűzfalat, azonban lehetetlené teszi azt, hogy a hálózaton kívülről kapcsolatokat kezdeményezzenek a belső hálózat felé.

## Tűzfalak tervezése

A tűzfalak tervezésekor a hálózatok adminisztrátorai el kell döntsék, hogy melyek azok a hálózaton kívül levő szolgáltatások amelyekre a belső felhasználóknak szükségük van, melyek azok a szolgáltatások amelyek kívülről is elérhetőek kell legyenek, és ezeknek megfelelően felépíthető egy biztonsági politika. Egy jó biztonsági politika csak annyit enged meg a felhasználóknak amennyire fel van jogosítva, csak azokat az információkat érhetik el amihez joguk van.

Egy biztonsági politika kidolgozásához a következő dolgokat kell figyelembe venni:

- kik az ellenségek – számba kell venni kik azok a személyek akik kikerülnék a biztonsági intézkedéseket, melyek az okaik, motivációik. Állapítsuk meg mit akarhatnak, és milyen károkat okozhatnak a hálózatban. Egy biztonsági rendszer soha sem teszi lehetetlené a jótalan hozzáféréseket, csak megnehezíti őket. A cél az, hogy a felállított védelem a betörő tehetségénél és motivációjánál erősebb legyen.
- ügyeljünk a feltételezéseinkre – mindig bizonyos feltételezésekből indulunk ki. Például egyesek azt feltételezik, hogy a támadók kevesebbet tudnak mint ők, hogy mindig szokásos/szabályos programokat használnak.

Meg kell bizonyosodni arról, hogy feltételezéseink helyesek. Téves feltételekből kiindulva komoly biztonsági réseket hagyhatunk a rendszerben.

- ügyeljünk a titkainkra – a legtöbb biztonsági rendszer titkokon alapszik, ilyenek a jelszavak, a titkos kulcsok. Elég gyakran megtörténik, hogy a titkok nem is olyan titkosak.
- vegyük figyelembe az emberi tényezőt – sok biztonsági rendszer azon bukik, hogy a tervezők nem vették figyelembe a felhasználók reakcióját. Elég gyakran megtörténik, hogy jól előkészített, nehezen kitalálható jelszavakat megtalálunk feljegyezve a képernyő szélére, vagy a billentyűzet aljára. Képezzük ki a felhasználóinkat, hogy azok megértsék/elfogadják a biztonsági intézkedések lényegét. Például egy minimális dolog amit minden felhasználónak tudnia kell, az hogy a jelszavakat nem közöljük telefonon, vagy e-mailen keresztül.
- ismerjük fel a gyenge pontokat – minden rendszerben vannak sebezhető pontok. Ezeket rendszeresen fel kell mérni és meg kell érteni gyengéjüket, ez az első lépés hogy ezek a sebezhető pontok biztonságosokká alakuljanak.
- készítsünk különböző hozzáférési zónákat – a belső rendszer különböző zónákra való bontása megakadályozza azt, hogy valaki automatikusan hozzáférjen az egész rendszerhez ha már sikerült egy részére betörnie.
- ismerjük ki a normális működést – ismerve a rendszer normális működését, könnyebben felfedezzük a betörési kísérleteket.
- korlátozzuk bizalmunkat – tudomásul kell venni, hogy a programokban hibák is vannak, ezért pontosan kell ismerni, hogy milyen alkalmazásokra lehet számítani amikor egy biztonsági rendszert akarunk felépíteni.
- végezzünk biztonsági felméréseket – szinte minden változtatásnak amit a rendszeren belül végzünk van biztonsági oldala/vetülete is. Ez még inkább igaz amikor új szolgáltatásokat vezetünk be. Próbáljuk meg felmérni milyen biztonsági problémákkal járhat egy szolgáltatás, miképpen lehet azt a szolgáltatást manipulálni.
- ne feledkezzünk meg a fizikai biztonságról – a fizikai hozzáférés a tűzfalat biztosító routerekhez, szerverekhez általában elegendő egy tapasztalt felhasználónak, hogy ezeket teljesen az uralma alá vegye.

### **Tűzfalak működtetése**

Ha már elkészült a biztonsági politika akkor a következő lépés ennek a politikának a végrehajtása. A tűzfalak megvalósítására a különböző cégek céleszközöket és programcsomagokat (szoftvereket) gyártanak mint például a Cisco PIX Firewall, Zone Alarm, CheckPoint Firewall. Ezen eszközök általában drágák ezért nem mindennapi jelenségek az oktatási intézményekben, amelyek egyik közös



jellemzője talán a pénzhiány. Azonban léteznek szabadon terjeszthető és általában szabad forráskódú programcsomagok amelyek kevés pénz mellett is megadják a lehetőséget egy biztonságos rendszer kialakítására. Ily ének a Squid http proxy, amely egy nagyon elterjedt http átjáró. Az alap http proxy szolgáltatás mellett a Squid képes ftp és gopher kérések átjártására is, ugyanakkor egy saját adatbázisában tartja az utóljára kért adatokat, amelyeket egy újabb kérés esetén már nem az internéiről hoz megspórolva ezáltal az ismétlődő adatok által elpazarolt sáv-szélességet. Alap konfigurálása nagyon egyszerű, a bonyolultabb szolgáltatások (pl. sebességkorlátozás, tartalom szűrés) beállítására részletes dokumentációt találunk az interneten.

A csomagszűrés megvalósítására egy Linux gép használható, amelyen egy újabb kernel fut. A jelenlegi linux csomagok általában a 2.4. kernel valamelyik változatával vannak ellátva, amelyekbe már bele van kompilálva a csomagszűrő. A mostanában elterjedt iptables nagyon gazdag szűrési lehetőségeket biztosít, így például szűrhetjük a csomagokat a forrás/cél cím, forrás/cél port szerint, az IP, TCP fejléc különböző jelző bitjei szerint, a kapcsolat állapota és még sok más paraméter szerint. Ha pedig valaki nem találná meg az igényét kielégítő szűrőt, akkor a teljes forráskód a rendelkezésére áll, hogy megírja magának. :-)

Egy szabad terjesztésű szoftvereket használó tűzfal hardverigénye eléggé szerény. Egy csomagszűrőt már egy 486-os géppel is megvalósíthatunk, azonban ha squidet is akarunk használni akkor jó, hogy egy gyorsabb számítógépet használjunk (pl. PII, 64 MB RAM, 3 GB merevlemez). Manapság az előbb említett hardver eszközök már nem jelentenek komoly anyagi terhelést, ugyanakkor a megfelelő beállításukra felkészült személyzet inkább az ami hiányzik. Újabban azonban már erre is van segítség a különböző konfiguráló eszközök formájában amelyek a mostani linux csomagokkal (disztribúciókkal) együtt jönnek, vagy letölthetők az internéiről.

## Irodalomjegyzék

- [1] Andrew S. Tanenbaum - **Számítógép Hálózatok**
- [2] Oskar Andreasson - **Iptables Tutorial**, <http://iptables-tutorial.frozentux.net/iptables-tutorial.html>
- [3] Mark Grennan - **Firewall and Proxy Server HOWTO**, <http://www.tldp.org/HOWTO/Firewall-HOWTO.html>
- [4.] 4. Rusty Russell - **Linux iptables HOWTO**, <http://www.linuxguruz.com/iptables/howto/iptables-HOWTO.html>



# DoS<sup>1</sup> támadások elleni védelmi mechanizmusok az IPv6-ban

## Protection Mechanisms Against DoS Attacks in IPv6

Somodi Zoltán

Kolozsvári Műszaki Egyetem

### Abstract

*This paper presents one of the most important features of the IPv6 protocol: security. Since it was developed in order to solve the address space shortage of the current IP protocol, it provides solutions for many security problems detected in IPv4. In the first part of the paper we present some of the most "popular" Internet DoS attacks. For each attack type there is a special solution, a particular protection mechanism. In the second part the IPSec protocol is presented as a protection against DoS attacks based on IP Spoofing. Although, IPv6 is not a panacea for security problems, security provided by IPv6 is a great improvement over IPv4, and is a good reason to deploy IPv6 in our network.*

### 1. Bevezetés

Az Internet immár több mint 30 éves története alatt nagyon sok támadást jegyeztek fel a statisztikákba. Az idő folyamán a támadások egyre sűrűbbek és egyre változatosabbak lettek.

Az Internetre kötött eszközök számának rohamos növekedése szükségessé tette az IP címtartomány kibővítését. Ezt egy új Internet Protokoll, az ún. IPv6 bevezetésével valósult meg. Az IPv6 tervezésénél figyelembe vették az eddigi tapasztalatokat, és kijavították az észlelt hibákat, hiányosságokat. Így, az IPv6 megoldást nyújthat több támadástípus ellen is.

Ebben a cikkben bemutatunk néhány gyakori támadástípust, valamint a jelenlegi védekezési módszereket, majd áttekintjük az IPv6 azon tulajdonságait, amelyek védelmet nyújtanak ezen támadások ellen.

### 2. Biztonsági hibák az IPv4-ben

Az internetes támadásokkal foglalkozó szakemberek arra a következtetésre jutottak, hogy támadásokat négy fő csoportba lehet besorolni [1]. Ezek a támadástípusok a következők:

---

<sup>1</sup> Denial of Service (ang.) – Szolgáltatásmegtagadás

- Szolgáltatás megtagadás
- Információlétrehozás
- Információtorlás vagy módosítás
- Elektronikus fülelés

Minden támadástípus az IPv4 egy-egy gyenge pontját, hiányosságát használja ki. Az alábbiakban bemutatjuk ezeket a hiányosságokat, kiemelve azokat, amelyeken a szolgáltatás-megtagadásos támadások alapszanak, és ismertetjük a jelenlegi védekezési módszereket.

A szolgáltatásmegtagadásos támadás (röv: *DoS - Denial of Service*) jelenleg az egyik legnehezebben kivédhető támadás-típus. Ilyen esetekben a támadó megpróbálja úgy leterhelni a rendszert vagy a hálózatot, hogy egy bizonyos szolgáltatásnak a működése lehetetlenné váljon. Általában kiszolgáló gépek a célpontok, pl. web-szerverek, DNS-szerverek, FTP-szerverek, email-szerverek stb. Sok megoldás létezik arra, hogy hogyan lehet megbénítani egy szolgáltatást, de majdnem mindenik módszer közös vonása, hogy hamis IP címeket használ. A jelenlegi internet protokoll egyik nagy hátránya, hogy lehetővé teszi a címhamisítást (ang.: *IP Spoofing*), és a hamis címekkel való hálózati kommunikációt. Ez az oka annak, hogy nagyon nehezen lehet védekezni a DoS típusú támadások ellen. Hamis címeket két okból használhatnak a támadók. Egyrészt, természetesen, azért, hogy elrejtsek valós azonosságukat. Másrészt, a hamis címek használata a támadási mechanizmus részét képezhetik, ahogy ezt a továbbiakban látni fogjuk.

A DoS támadásoknak több változata ismert, ezekből fogunk az alábbiakban bemutatni egy néhányat.

**A Smurf támadás.** A támadó *ICMP echo request* típusú csomagokat küld broadcast IP címekre. Ez azt jelenti, hogy több gép a helyi hálózaton kap egy-egy ilyen csomagot, amire *ICMP echo reply* csomaggal válaszol. Ezek a csomagok megnövelik a hálózati forgalmat, torlódáshoz vezetve. Általában a támadó nem a saját címét használja, hanem forráscímként egy másik gép IP-jét használja, amely ellen tulajdonképpen irányul a támadás. Tehát az összes ICMP válasz-csomag erre a csomópontra irányul, ami megbéníthatja az illető gép működését.

Az ilyen támadások elleni védekezés lehet az, ha tiltjuk a broadcast címekre irányuló ICMP forgalmat [2]. Így megakadályozhatjuk a támadás továbbterjedését. De ha ez nem történik meg, a célgépet nagyon nehezen tudjuk megvédeni.

**TCP SYN elárasztás** (ang.: *flooding*). A támadás a TCP kapcsolatok kialakításának módját használja ki. Ahhoz, hogy egy ilyen kapcsolat létrejöhessen egy kliens és egy szerver között, három lépésre van szükség. Először a kliens egy SYN csomagot küld, amire a szerver SYN-ACK csomaggal válaszol. Végül a kliens egy ACK csomaggal fejezi be a kapcsolatteremtést. A támadás



abból áll, hogy egy „kliens” elárasztja a kiszolgálót SYN csomagokkal, de nem válaszol a SYN-ACK csomagokra. Így a szerver erőforrásai elhasználódnak a rengeteg félig nyitott kapcsolat nyilvántartására, és nem lesz lehetőség a valódi kliensek kiszolgálására. Természetesen a támadó hamis, és változó forráscímeket használ, így nem lehet kiszűrni a támadást [3].

**TCP ACK vihar.** Minden gép, amelynek élő TCP kapcsolata van más gépekkel, nyilván kell tartsa a fogadott csomagok sorszámát. A támadás lényege, hogy megzavarja ezt a nyilvántartást, hamis csomagokat küldve a kapcsolatban álló állomásoknak. Ez olyan forgalmat eredményez, amely megbéníthatja a megtámadott gépek működését [4].

**LAND támadás.** Egyes TCP/IP implementációk sebezhetőek olyan SYN csomagokkal, amelyeknek forrás- és célcíme, illetve portja azonos. A LAND ezt a hiányosságot használja ki, hamis csomagokat küldve a célgépre [5]. A IP protokoll jelenlegi változatával lehetetlen kiszűrni ezt a támadást.

**Könnycsepp-támadás** (ang.: *Teardrop attack*). A TCP/IP protokoll-együttes egyik fő feladata a csomagok tördelése. Ha az útvonalon a következő router nem képes kezelni egy adott méretű csomagot, akkor azt kisebb darabokban fogja neki továbbküldeni a szomszédja. Minden töredékben található egy ofszet, amely a következő töredékre mutat. A támadó olyan hamis csomagokat gyárt, amelyben az ofszet helytelen. Ez megzavarhatja a töredékek összeillesztésének folyamatát, amely a rendszer összeomlásához vezethet.

**Halálos ping** (ang.: *ping of death*). Ebben a támadásban is a csomagok méretét manipulálja a támadó. A legnagyobb megengedett IP csomag 65.536 bájt hosszú lehet. Ennél nagyobb csomagokat a hálózat nem továbbít. De küldhetünk több olyan csomagrészt, amelyeknek összmérete nagyobb a megengedettnél. Bizonyos operációs rendszerek nem tudnak mit kezdeni ezekkel a túlméretezett csomagokkal (amelyek az összeillesztés után keletkeznek) és a rendszer lefagyásához, összeomlásához vagy újraindulásához vezethet.

### 3. Az IPv6 biztonsági rendszere

A biztonsági mechanizmus fő részét az IPSec adja az IPv6-ban. Az IPSec egy biztonsági szabványrendszer, amelyet eredetileg az IPv6-hoz fejlesztettek ki. Mivel nagy szükség volt a biztonságra a jelenlegi IP-ben is, alkalmazták az IPv4-re is. Mégis, az IPv4-ben az IPSec opcionális szolgáltatás és nagyon gyártófüggő, több nem teljesen implementáció létezik. Az IPv6-ban viszont az IPSec kötelező, végpont-végpont biztonságot nyújtva.

Az IPSec a hitelesítési (authentication) fejléc (AH) és a titkosítási fejléc (ESP) segítségével valósítja meg a biztonsági szolgáltatásokat. Ezek a fejléc-



cek használhatók külön-külön, kombinálva a szükséges biztonság függvényében. Mindkét fejléc két módban használható:

- *transzport mód*: a felsőbb szintű protokollok (TCP, UDP, ICMP), azaz az IP csomag adatmezejének a védelmére szolgál [6]. A transzport mód tipikusan két host (IP kommunikációs szereplő) közti végpont-végpont kapcsolatokban használatos.
- *tunnel mód*: az egész eredeti csomagot egy másik IP csomag belsejébe helyezik (IP-IP tunnelezés), így biztosítva, hogy az egész eredeti csomag a (publikus) hálózaton való áthaladás közben változatlan maradjon. A tunnel módot leginkább két biztonsági gateway (pl. tűzfal vagy router) között használják, amely esetben a két gateway között egy VPN-t (Virtual Private Network, virtuális magánhálózat) hoznak létre.

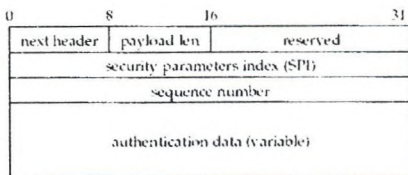
Az IPSec egyik kulcsfogalma a *Security Association* (SA – Biztonsági kapcsolat). A biztonsági fejlécekben található *Security Parameters Index* mező a cél IP címmel és a biztonsági protokollal (AH vagy ESP) együtt egyértelműen meghatároz egy SA-t. Ez egy egyirányú kapcsolat a kommunikáló partnerek között. Általában tartalmazza a hitelesítéshez vagy titkosításhoz szükséges kulcsot, valamint a használt hitelesítési illetve titkosítási algoritmust. Az IKE (Internet Key Exchange) leírja azt a folyamatot, amely által létrehozható egy új SA.

### 3.1. Az AH protokoll

A hitelesítési fejléc (Authentication Header, AH) az egyes IP csomagok hitelesítése, a bennük lévő adat sértetlenségének az ellenőrzésére és ún. replay támadások elleni védelemre nyújt módot. A hitelesítés lehetővé teszi, hogy a csomag címzettje megbizonyosodjon az IP csomag feladójáról. Az adatsértetlenség ellenőrzésével a címzett igazolhatja, hogy a csomagot annak feladása óta egy közbülső ponton nem módosították.

A 1. ábrán az AH fejléc szerkezete látható, amelyet az alábbiakban ismertetünk részletesebben.

A Next Header (következő fejléc) mező adja meg az AH fejlécet közvetlenül követő fejléc típusát. Az AH fejléc teljes hosszát a Payload Length tartalmazza.



1. ábra  
Az AH fejléc

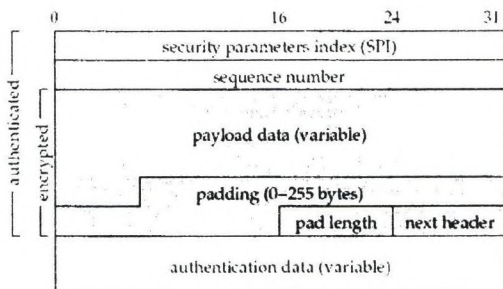
A Security Parameters Index mezőben található azt a számot, amely azonosítja az SA-t. Egy monoton növekvő számláló értékét tartalmazza a Sequence Number, amelynek – ahogy azt a későbbiekben látni fogjuk – a replay támadások elleni védelemben van szerepe. A változó méretű Authentication Data tartalmazza azt az ICV (Integrity Check Value) értéket, amelyet a csomagok hitelesítésére és sértetlenségének ellenőrzésére használunk. Az ICV kiszámítása során az IP csomag következő részeit vesszük figyelembe: az IP fejléc azon mezői, amelyek vagy nem változnak meg az út során, vagy amelyeknek a célpontbeli értéke a feladó számára kiszámítható. A megváltozó és nem kiszámítható mezők tartalmát az ICV kiszámításához a feladó és a címzett oldalon egyaránt zérusokkal helyettesítik; az egész AH fejléc úgy, hogy a hitelesítési mező nullákkal van kitöltve; az egész felsőbb szintű protokoll adat, amelyről feltesszük, hogy az út során nem változik meg.

A küldő a fenti logikai algoritmus alapján számolja ki az ICV értékét, amelyet azután a hitelesítési adatmezőben helyez el. A fogadó szintén ezek alapján számolja ki az ICV értékét és veti össze az eredményt a hitelesítési mezőben kapott értékkel. Ha a kettő megegyezik, a csomag feladójának a hitelesítése és a csomag integritásának az ellenőrzése sikeres volt.

### 3.2. Az ESP protokoll

Az ESP (Encapsulating Security Payload) titkosítási szolgáltatást nyújt, védve az üzenet tartalmát a lehallgatások ellen, valamint korlátozott védelmet tud nyújtani a forgalmi adat-analízisen alapuló támadások ellen. Az ESP opcionálisan az AH-hoz hasonló hitelesítési szolgáltatásokra is képes.

Az ESP fejléc szerkezetét a 2. ábrán mutatjuk be, az alábbiakban a mezők jelentését írjuk le.



2. ábra  
Az ESP fejléc



Ez a fejléc is tartalmaz egy SPI mezőt, amely azonosítja az SA-t, valamint egy Sequence Number-t, ugyanazzal a jelentéssel, mint az AH-ban. A titkosított adatot az adatmezőben (Payload Data) mezőben találhatjuk, ezt egy helykitöltés (padding) követ, amely a titkosító algoritmusok helyes működéséhez szükséges. A helykitöltés hosszát a Pad Length adja meg, az ezt követő Next Header pedig a következő fejléccet azonosítja.

Az adatmező, helykitöltés, helykitöltés hossza és következő fejléc mezőket az ESP az SA-ban tárolt szimmetrikus titkosító algoritmussal titkosítja. Az ESP először titkosítja a hasznos adat, helykitöltés, helykitöltés hossza és következő fejléc mezőket, majd hitelesíti az ESP csomagot. Az ESP csomagot tartalmazó IP csomag fejléc-elemeinek a hitelesítését az ESP – az AH-val ellentétben – nem végzi el, ezért az ESP NULL titkosítás melletti autentikációs szolgáltatása nem teljesen azonos az AH szolgáltatásával.

### 3.3. Támadások elleni védelem IPSec-kel

Amikor az IPv6-ot tervezték, az előző fejezetben bemutatott támadások mind ismertek voltak. Így bizonyos védelmi mechanizmusokat be lehetett építeni az új protokollba.

Amint említettük, az IPv6-ban a biztonsági szolgáltatásokról az IPSec protokoll gondoskodik. Az IPSec *opcionálisan* az IPv4-gyel is használható, az új protokollban viszont *kötelező* szolgáltatás. Lényeges különbség van a két megközelítés között. Az IPv4 esetében használhatjuk a biztonsági protokollt, de csak akkor, ha a kapcsolat másik végén levő csomópont is ismeri a szolgáltatást. Egy IPv6-os hálózaton viszont, biztosak lehetünk abban, hogy minden csomóponton működik az IPSec. Ez megoldást nyújt a legtöbb IPv4-ben tapasztalt biztonsági hiányosságra. Ebben a részben főleg azokat a védelmi mechanizmusokat ismertetjük, amelyek a DoS típusú támadásokat hiúsítja meg.

A különböző támadások ismertetéséből kiderült, hogy a legtöbbjük az IP csomag tartalmának meghamisítására alapszik. Sok esetben a forrás- illetve célcímet hamisítják a támadók, de az is előfordul, hogy más mezőt változtatnak meg. Ezt a támadási módszert, amelyet az angol szakirodalom IP Spoofing-nak nevez, az új Internet protokoll lehetetlenné tesz. Az AH vagy az ESP fejléc használatával egy hálózati csomópont mindig biztos lehet abban, hogy a fogadott csomag arról az IP címről jött, amelyet a csomag tartalmaz, hogy nem változott meg útközben az információ tartalma, és hogy a kommunikáló partnereken kívül senki más nem látta a csomag tartalmát.

A DoS támadásoknál a támadónak két célja lehet az IP csomag meghamisítására. Egyrészt, minden támadó el szeretné rejteni valós azonosságát, ezért megváltoztatja a forráscímet. Másrészt, ha a csomag egyes mezőit egy megadott formában módosít, bizonyos működési zavarokat okozhat a célgépen, vagy az egész hálózaton. Például a LAND támadásoknál a forrás- és célcím,



illetve forrás- és célpont azonos kell legyen, ahhoz, hogy hatásos legyen a támadás. Ezek a támadási formák mind lehetetlenné válnak, ha az IP csomagokat azonosítjuk. Ha a fogadott csomag kiszámolt ICV értéke nem egyezik meg a csomagban levővel, akkor azt a csomagot vissza kell utasítani.

Vannak viszont olyan DoS támadások, amelyek címhamisítás nélkül is megvalósíthatók. Ilyen például a TCP SYN elárasztás. Ha a támadó a saját valós címéről küldi a SYN kéréseket, és nem válaszol a SYN-ACK csomagokra, egy idő után pont úgy megbéníthatja a szerver működését, ahogy azt az előző fejezetben láttuk. Ebben az esetben viszont, be lehet építeni olyan védekezést, amely nem enged be a hálózatba csomagokat olyan címről, amely már több fél-nyitott kapcsolatot kezdeményezett. Ugyanakkor, a támadót (vagy legalább annak gépét) azonosítani lehet a címe alapján, és ez elbátortalanítja a rossz szándékú felhasználókat.

A Smurf, TCP ACK vihar és LAND támadások ellen is védelmet nyújt az AH protokoll sértetlenségét ellenőrző szolgáltatása. Ha egy csomag megváltozott tartalommal érkezik egy csomópontra, ezt a célgép észleli, és visszadobja a csomagot. Így nem történhet meg az, hogy egy hamis sorozatszámú csomag megbontsa a kommunikáló partnerek közötti szinkront.

Bemutattunk két olyan DoS támadást is, amely nem csomaghamisításon alapszik, hanem a TCP/IP protokoll-együttes tördelési mechanizmusának hiányosságait használja ki. A „könnycsepp” és a „halálos ping” támadásokról van szó. Igaz, ezekben is esetekben is szükség van egy módosított tartalmú csomag küldésére, de ezt a módosítást elvégezheti a támadó is az ICV érték kiszámítása előtt. Így az ICV-t egy hibás csomagra számolja ki a biztonsági protokoll, és a célgép nem fogja észlelni, hogy valami nincs rendben. De az IPv6-nak van egy másik fontos tulajdonsága is, amely megakadályozza az ilyen típusú támadásokat. Az új protokollban csak a végpontokon levő állomások tördelhetik szét és állíthatják ismét össze a csomagokat, a közbeesők nem. A forrás állomás meghatározza a maximálisan átvihető méretű egységet (MTU), és ha kell, tördelést végez. Ekkor az összeillesztéshez szükséges információkat a Fragmentation Header nevű kiegészítő fejlécben tárolja el, mely csak ebben az esetben van jelen a csomagban.

Kutatócsoportunk méréseket végez, amelynek célja, megvizsgálja az IPv6 stabilitását tördelési támadások esetén valós hálózatokban.

#### **4. Következtetések**

Az IPsec jelenleg az egyik legjobb biztonsági szolgáltatást nyújtó protokoll. Kötelező szolgáltatásként jelenik meg az IPv6, míg az IPv4-nél csak opcionális volt. Az IPsec jelenléte miatt, az IPv6-os hálózatok biztonságosabbá válnak a címhamisításos, illetve szolgáltatás-megtagadásos támadások és a lehallgatásokkal szemben.

Habár az IPv6 által nyújtott biztonsági szolgáltatások nagy fejlődést jelentenek az IPv4-hez képest, mégis nem tekinthetjük ezt sem egy csodaszernek, neki is megvannak a saját biztonsági rései és hátrányai. Ezek közül íme néhány [1]:

- Az export törvények miatt, a használható titkosítási algoritmusok erőssége korlátozott.
- Az IPsec a Public Key Infrastructure-re (PKI) alapszik, amely még nem teljesen szabványosított.
- További fejlesztés szükséges az IKE protokoll illetve a DoS és az el-  
árasztásos támadások elleni védekezés területén.

Az IPv6 megold néhány fontos biztonsági problémát, de ugyanakkor újabb támadási lehetőségek is adódnak [7]:

- A DoS típusú támadások általi sebezhetőség, mivel a titkosítási eljárások processzor-igényes folyamatok.
- A tűzfalak részére lehetetlenné válik a forgalmi analízis, ha használjuk az ESP titkosítási protokollt. Tunnel módban használva, létre lehet hozni kapcsolatot egy nem megbízható állomással egy privát hálózaton belül, mivel az IP cím láthatatlan a tűzfal számára.
- Az IPv6 autokonfigurációs követelményei lehetőséget nyújtanak bizonyos DoS támadásokra, mivel a rendszer sebezhetővé válik hámított alhálózat maszkokat tartalmazó csomagokra.

Mind ezek ellenére elmondhatjuk, hogy az IPv6-ot még akkor is érdemes bevezetni hálózatunkba, ha van elég szabad IP cím. Az új protokollnak, a kibővített címtartomány mellett, más fontos tulajdonságai is vannak: biztonság, szolgáltatásminőség garanciák, autokonfiguráció stb.

## 5. Könyvészet

- [1.] S. Hagen, IPv6 Essentials, O'Reilly, Sebastopol, CA, pp. 77-88, 2002
- [2.] "CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks", 1998, <http://www.cert.org/advisories/CA-1998-01.html>
- [3.] "CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks", 1996, <http://www.cert.org/advisories/CA-1996-21.html>
- [4.] D. Anderson, B. Teague, "ACK Storms and TCP Hijacking", <http://www.owl.net.rice.edu/~bteague/papers/comp527.pdf>
- [5.] "I-019: Tools Generating IP Denial-of-Service Attacks", in CIAC Information Bulletin, December 19, 1997, <http://ciac.llnl.gov/ciac/bulletins/i-019.shtml>
- [6.] TIPSTER6, Testing Experimental IPv6 Technology and Services in Hungary, 2001, [http://tipster6.ik.bme.hu/tipster6\\_ener6\\_en.html](http://tipster6.ik.bme.hu/tipster6_ener6_en.html)
- [7.] Y. Chauhan, D. Sanghi, "Security in the wake of IPv6", A Term Paper Report for Advanced Computer Networks, Indian Institute of Technology Kanpur, 2003



# Ipari kommunikációs hálózatok

## Industrial Communication Networks

Dr. Sebestyén Pál György

### Abstract

*Modern distributed control systems require an adequate communication infrastructure. Industrial networks are specially designed to fulfill the requirements of an industrial environment. The paper presents issues and solutions concerning the design and implementation of control applications based on industrial networks. The first part of the paper summarizes communication requirements and conditions in a distributed control application. A number of industrial protocols are analyzed as solutions to these requirements. Real-time and reliability issues are analyzed for some representative protocols, such as: Profibus, WorldFIP, CAN, Asi, and others. The last part of the paper studies the possibility to use Internet technologies for control purposes.*

Manapság a kommunikáció fontos szerepet játszik bármilyen emberi tevékenységben. Így a gyártási folyamatok vezérlésének minősége és hatékonysága egyre inkább függ az alkalmazott kommunikációs technológiáktól. Ma már nehezen lehet elképzelni egy komplex, elosztott folyamat vezérlésének tervezését kommunikációs eszközök használata nélkül. Ezért fontos meghatározni, milyen adatközlési eszközök állnak rendelkezésünkre, milyen sajátos igényeknek kell megfeleljenek és hogyan lehet kiértékelni és biztosítani a sajátos tulajdonságokat. Mindezekre a kérdésekre próbál fényt deríteni az alábbi cikk.

### Adatközzvetítés a vezérlőrendszerekben

A vezérlőrendszerek fejlődése folyamán különböző adatátviteli eszközöket alkalmaztak. Ezek az eszközök követték a létező domináns technológiát. Így az első eszközök mechanikus jellegűek voltak. Egyszerű mechanizmusok (pl. huzalok, karok) segítségével sikerült közvetíteni a folyamatparamétereket és a parancsokat. Legtöbbször egyedi megoldásokról beszélhetünk; nehezen lehetett általánosítani és elméleti alapokra építeni az irányítási megoldásokat. A mechanikusan közvetített jelek nehezen dolgozhatók



fel és kevésbé pontosak. Az eszközök kopása befolyásolja a vezérlés minőségét.

Egy fontos előrelépést jelentett a hidraulikus és pneumatikus eszközök alkalmazása. A folyamatadatokat nagyobb távolságra és pontosabban lehet közvetíteni ilyen fajta eszközökkel. Ugyanakkor szét lehet választani a vezérlőlogikát tartalmazó egységet a közvetítő elemektől. Gyakorlatilag ugyanaz a szabályozó szerkezet alkalmazható több fajta paraméter szabályozásához. Viszont a hidraulikus vagy pneumatikus vezérlés aránylag költséges és kevésbé flexibilis csőrendszert igényel. Szükség van egy nyomásgenerátorra, amely akkor is kell működjön, ha nincs változás a rendszerben.

A villamos jelek bevezetése egy újabb előrelépést jelentett. A folyamatadatokat olcsóbban, pontosabban és jóval nagyobb távolságokra lehet közvetíteni villamos jelek által. A jelek feldolgozása terén is lényegesen javult a helyzet. Az elektronikus elemek (erősítők, ellenállások, kapacitások, induktív elemek) alkalmazása lehetővé tette a bonyolult jelfeldolgozási eljárások megvalósítását. Gyakorlatilag egy új szabályozási elmélet jön létre – a rendszerelmélet –, amely matematikai alapokra állítja a rendszerek irányítását.

A szabványosított (egységesített) jelek bevezetése leegyszerűsítette a különböző típusú automatizálási eszközök egybekapcsolását. A lényege az, hogy függetlenül milyen eszközt alkalmazunk vagy milyen folyamatmennyiséget közvetítünk, az információt hordozó jel (feszültség vagy áram) ugyanabban a tartományban változik. A számítások is leegyszerűsödnek, mivel bármilyen jel (mennyiség) nulla és egy között változik.

A digitális (két állapotú) jelek alkalmazása több szempontból is előnyt jelent: nagyobb zajimmunitást biztosítanak, az adatokat jóval nagyobb távolságra lehet közvetíteni hibamentesen, a mért mennyiségek pontossága egyértelműen meghatározható (a használt bitek száma által), a jelfeldolgozási eljárások könnyen megvalósíthatók, stb. Vannak viszont bizonyos korlátok és hátrányok: a folyamatparaméterek csak diszkrétan (nem folytonosan) figyelhetők, a jelek maximális frekvenciája korlátozott és az adatok kvantizálása zajt okozhat. Ennek ellenére a digitális technológia egyre nagyobb teret hódít a folyamatirányítás területén. Ez azzal is magyarázható, hogy a mai digitális technológiák és főleg a mikroprocesszorok egy aránylag olcsó és biztonságos fejlesztési hátteret nyújtanak az irányítási eszközök megvalósításához.

A digitális jelektől már csak egy lépés volt a kommunikációs hálózatok alkalmazásáig. A hálózati kommunikáció minőségi lépést jelentett a folyamatvezérlésben. A hálózat által jóval komplexebb (tartalomban gazdagabb) adatokat lehet közvetíteni a különböző irányítási elemek között. Elosztott vezérlőrendszereket lehet kiépíteni, amelyekben az „intelligencia”, vagyis a

döntési jog megosztható. Ezáltal a rendszer megbízhatósága nő és csökken a reagálási idő. Viszont ahhoz, hogy egy hálózati protokoll megfeleljen egy ipari környezetnek, sajátos tulajdonságokkal kell rendelkeznie. Sajnos a számítógépek közti kommunikációra használt hálózatok gyakran nem alkalmazhatók vezérlési célokra. A számítógép-hálózatokban alkalmazott protokollokból hiányoznak azok a biztonsági és valós-idejű tulajdonságok, amelyek szükségesek a legtöbb irányítórendszerben.

### **Mit és hogyan kell közvetíteni?**

A vezérlőrendszerekben közvetített információ tartalmilag és formailag lényegesen különbözik az általános jellegű számítógépes alkalmazásoktól. Az utóbbiakra jellemző a nagy adattömbök, állományok közvetítése. Nincsenek időkorlátok és ritkán alkalmaznak minőségellenőrző és biztosító eszközöket. Ezzel ellentétben a vezérlőrendszerekben aránylag egyszerű adatstruktúrákat alkalmaznak, mint például:

- állapotértékek (nyitott/zárt, bekapcsolt/kikapcsolt, engedélyezet/nem engedélyezett) – digitális jellegű értékek
- fizikai mennyiségek értéke (hőmérséklet, nyomás, szint stb.) – analóg jellegű értékek
- beállítási (konfigurálási) adatok – vegyes értékek

Ezért az üzenetek átlaghossza jóval rövidebb, mint a számítógép-hálózatokban. A protokoll hatékonyságát rövid üzenetekre kell beállítani. Ugyanakkor az adatokat periodikusan kell közvetíteni, mivel a legtöbb vezérlési eljárás három lépés ismétlését jelenti: adatgyűjtés, feldolgozás és (szabályozó) parancsgenerálás. A protokoll olyan mechanizmusokat kell tartalmazzon, amelyek támogatják és biztosítják a periodikus üzenetek közvetítését. Bármilyen eltérés a periodikus közvetítéstől negatívan befolyásolja a szabályozási eljárás minőségét.

Egy vezérlőrendszerben vannak kritikus és kevésbé kritikus üzenetek. A kritikus üzenetek kezelési módja egyenesen meghatározza a rendszer biztonságát és helyes működését. Ezért az ilyen üzeneteket nagyobb prioritással kell közvetíteni. Ehhez viszont szükséges, hogy a protokoll támogassa a több prioritásos üzenetközvetítést.

Három fontosabb jellemző határozza meg, hogy hogyan kell egy vezérlőrendszerben közvetíteni:

- biztonságosan
- pontosan
- időben



A biztonságos közvetítés szükséges ahhoz, hogy garantálni lehessen az egész rendszer biztonságát. Általában nem fogadható el egy olyan helyzet, amelyben a gyártási folyamat meghibásodása a kommunikációs eszközök miatt történik. Egy általános számítógép-alkalmazásban egy ilyen helyzet zavaró, de nem kritikus; nem jár komoly anyagi károkkal vagy akár emberi sérüléssel, ahogy ez gyakran előfordulhat egy ipari vezérlésben. Emiatt a hálózatnak tartalmaznia kell hiba-toleráns eszközöket, amelyek által egy esetleges közvetítési hiba takarható (maszkolható). Ehhez viszont szükséges a fizikai és időbeli redundancia, vagyis az, hogy több eszköz teljesítse ugyanazt a funkciót és minden üzenet többszörösen legyen közvetítve. Például egy ilyen megoldást választottak a MARS referenciamodellben, ahol minden üzenetet kétszer közvetítenek, és a hálózatnak két párhuzamos szakasza van. Ahhoz, hogy garantálni lehessen az üzenetek időbeli és hibamentes közvetítését, egy statikus, időszeletekre alapozott üzenet-tervezési algoritmust alkalmaznak.

A közvetítés pontosságát főleg a digitális technológiák biztosítják. A kívánt pontosság eléréséhez megfelelő bitszámú analóg-digitális és digitális-analóg konvertereket alkalmazunk. A hálózati kommunikáció lehetővé teszi a mérőeszközök tesztelését és kalibrálását. Több érzékelő szolgálhat egy fizikai mennyiség meghatározásához. Korrelációs és predikciós alapon fel lehet ismerni egy helytelen értéket több mért érték közül.

Az üzenetek meghatározott időkorlátokon belüli közvetítése a vezérlő-rendszerek egyik sajátos és ugyanakkor fontos kérése. Az úgynevezett „best effort” típusú megoldások, amelyeket hatékonyan alkalmaznak más általános hálózati megvalósításokban, nem felelnek meg az irányítástechnikában. A tapasztalat azt mutatja, hogy egy elkésett üzenet gyakran veszélyesebb, mint egy hiányzó üzenet. Az utóbbi helyzet könnyen érzékelhető és ezáltal kijavítható. Viszont egy későn érkezett mért-érték nem tükrözi a rendszer pillanatnyi helyzetét, és emiatt rossz irányba befolyásolja a szabályozási algoritmust. Egy elkésett parancsnak hasonló negatív eredménye lehet.

A legtöbb szabályozási függvény egyik meghatározó paramétere az idő. A digitális szabályozásban feltételezzük, hogy a mintázás periódusa állandó. Amennyiben ez nem tartható be, főleg a kommunikációs késések miatt, a kiszámított parancs pontossága csökken. Ezért a hálózati protokollnak valamilyen formában támogatnia kell a szigorúan periodikus üzenetközvetítést.

### **Hálózati kommunikáció az iparban**

A hálózati kommunikáció több előnyt nyújt a hagyományos, vezérlő-rendszerekben alkalmazott, kommunikációs eszközökkel szemben:

- olcsóbb adatközlési infrastruktúra



- egy eszköz (környezet) több kapcsolatot biztosít
- komplex adatok két- vagy többirányú közlése
- biztonságos adatközlés:
  - digitális technikák alkalmazása révén
  - különleges biztonsági eljárásoknak köszönhetően (protokollban beágyazott hibaérzékelő és javító módszerek)
- könnyen fejleszthető és újrafelállítható környezet
- együttműködési lehetőség (interoperabilitás) és szabványosítás

Mindezek mellett a hálózati technológiákat aránylag későn (a '90-es évek elején) kezdték alkalmazni az ipari folyamatok irányításában. Ez részint azzal magyarázható, hogy az általános számítógép-hálózatok nem rendelkeznek olyan tulajdonságokkal, amelyek szükségesek egy ipari alkalmazásban. Szükség volt kifejleszteni új hálózati protokollokat, amelyek kimondottan az ipari alkalmazások kéréseire épültek.

Így születtek az ipari kommunikációs hálózatok. Ehhez a családhoz tartoznak mindazok a hálózatok, amelyeket főként irányítási célokra fejlesztettek. A család aránylag sok protokolltípust tartalmaz; gyakorlatilag minden nagyobb, automatizálási eszközöket gyártó cég kifejlesztette a saját hálózatát. Ez, főleg az első időszakban, komoly együttműködési és inkompatibilitási gondokat okozott a rendszerfejlesztőknek és a felhasználóknak egyaránt. Ezért az utóbbi időben több konzorcium alakult, amelynek célja az ipari hálózatok egységesítése és szabványosítása. Sajnos, objektív és szubjektív okok miatt, a szabványosítási folyamat nehezen halad. Sok cég nem akar lemondani a saját protokolljáról, mivel meg szeretné tartani a kompatibilitást a már létező megvalósításokkal. Az is igaz, hogy nagyon sok típusú és komplexitású irányítási rendszer létezik, amelyek kommunikációs igényei nagyon különböznek. Ezét nehéz fejleszteni egy olyan szabványt, amely minden igényt kielégít és hatékonyan működik több környezetben.

Az ipari hálózatok fejlődésében a következő fázisok észlelhetők:

- a '80-as évek közepén – MAP – az első ipari hálózat; fejlesztő: General Motors, célja: a szerelvények automatizálásának korszerűsítése
- 1990-1996 – a nagy „bumm” – nagyon sok új protokoll fejlesztése (több mint 30)
- 1996-2000 – egyrészt szabványosítási próbálkozások, másrészt elméleti kutatások a kritikus tulajdonságok (idő, biztonság) matematikai elemzése és kiértékelése céljából
- 2000-2003 – Internet technológiák alkalmazása irányítási célokra

Ami az utóbbi periódust illet, annak ellenére, hogy az Interneten alkalmazott technológiák nagymértékben nem felelnek meg az ipari környezet kéréseinek, a felhasználók igényelnek Internetes felügyelést és vezérlést. Ezért komoly kutatási tevékenység folyik annak érdekében, hogy a jelenlegi Internetes protokollokat ipari célokra is alkalmazhatóvá tegyék.

### **Az ipari hálózatok osztályozása**

A használati kör szempontjából az ipari hálózatok három csoportra oszthatók:

- *irányító eszközök hálózata* (Instrumentation bus, Actuator/Sensor network)
- *terepsínek* (fieldbuses)
- *sejthálózatok* (cell networks)

Az első csoporthoz tartoznak azok a hálózatok, amelyek a folyamatirányítás legalacsonyabb szintjére kerülnek. Ezek a hálózatok az egyszerű automatizálási eszközök (pl. érzékelők, végrehajtó elemek, programozható kontrollerek stb.) bekapcsolására alkalmasak. A kommunikációs protokoll aránylag egyszerű kell legyen ahhoz, hogy könnyen be lehessen építeni az erőforrásokban korlátozott eszközökbe is, viszont nagyon gyors reaklási időt kell biztosítani (1-10 mikroszekundum körül). Az utóbbi tulajdonság fontos, mivel az adatközvetítésnek a helyi szabályozási hurkok szűk időkorlátjainak kell megfelelnie.

A hálózathoz való hozzáférést leggyakrabban egy master-slave típusú eljárással oldják meg. Ez a megoldás több előnyt nyújt:

- lehetővé teszi a központosított és determinisztikus hálózati irányítást
- könnyen megvalósítható egy szigorúan periodikus üzenetütemezés
- a protokoll komplexitását a több erőforrással rendelkező master-egységbe lehet beágyazni, ezzel egyszerűsítve a slave-egységek hálózati interfészét

Az irányítási eszközök hálózataiban az üzenetek hosszát a jellemző adatstruktúrák hosszához adaptálták. Ezért az üzenetek hosszát bitekben mérik. Például az ASI hálózatban egy üzenet szigorúan csak négy hasznos bitet tartalmaz, mivel a slave-egységek leggyakrabban csak bináris jeleket kezelnek.

A legjellemzőbb hálózatok ebből a családból a következők:

- ASI – Actuator Sensor Interface
- CAN – Control Area Network
- Interbus-S



Az ASi protokoll sínre épül és master-slave típusú hálózat-hozzáférési algoritmust alkalmaz. Egy hálózatzakaszra egy mestert és 32 slave egységet (64-et a kiterjesztett változatban) lehet kapcsolni. A protokoll lehetővé teszi az egységek automatikus felismerését és beállítását. Meghibásodás esetén az érintett egység feladatait automatikusan egy másik, várakozásban levő egységre hárulnak. Az üzenetközvetítés szigorúan periodikus és a master felügyelete alatt történik. Egy perióduson belül több fázis létezik: adatközvetítési fázisok, új egységek felismerésére használt fázis és tesztelésre használt fázis. Egy üzenet 4 kimeneti vagy 4 bemeneti bitet tartalmaz (a master-egység szempontjából nézve). Ezek a bitek egy slave egység bemeneti és kimeneti jeleinek állapotát mutatják.

A CAN protokoll eredetileg az autóiparnak készült, pontosabban a versenykocsikon levő processzorok kapcsolásához tervezték. Ma már nem csak a versenykocsikhoz, hanem más járművek irányításához is alkalmazzák; sőt, a protokoll nagyon hatékonyan bizonyult ipari környezetben is. A CAN protokoll egy CSMA/BA (Carrier Sense Multiple Access with Bitwise Arbitration) típusú hálózat-hozzáférési eljárást alkalmaz. Első pillantásra az ütközésre (kolizióra) alapuló eljárás nem determinisztikus, viszont az üzenetek fix prioritásából pontosan ki lehet számítani a különböző típusú üzenetek maximális késését. Az üzenetek rövid hossza biztosítja a gyors reagálási időt.

Az Interbus-S egy gyűrű alakú hálózatra épül. Létezik egy master-egység, amely szabályozza a periodikus adatközvetítést. Az üzenetek olyanok, mint valami vonatok, amelyek a gyűrűben forognak. A slave-egységek vesznek és tesznek adatokat a „vonat” vagonjaiba. A protokoll támogatja a termelő-fogyasztó (producer-consumer) kommunikációs modellt, ami feltételezi, hogy minden adatnak van egy „termelője” és egy vagy több „fogyasztója”. Ebben a hálózatban is a periodikus üzenetközvetítés biztosítja az időhatárok betartását.

A *terepsínek* (Fieldbuses) középszintű komplexitású automatizálási eszközök kapcsolására alkalmasak (pl. programozható kontrollerek, szabályozók, folyamat-számítógépek). Emiatt az üzenetek hosszabbak és bonyolultabb adatstruktúrákat és parancsokat tartalmaznak. Egyes protokollok lehetővé teszik a multi-masteres konfigurációk kialakítását. Ez fontos akkor, amikor hiba-toleráns rendszereket szükséges építeni. Az üzenetek maximális késési ideje korlátolt, és a közvetítés determinisztikus módszerekre alapzik. Több protokollt lehet ebbe a csoportba sorolni:

- Profibus 3 változatban: FMS, DP és AP
- WorldFIP
- P-NET
- LONWorks



Ezek a protokollok főleg abban különböznek, hogy milyen hálózathozzáférési eljárást alkalmaznak. A Profibus egy token-passing mechanizmust alkalmaz, amelyben a hálózatban való közvetítési jogot a master-egységek rendre átadják egymásnak egy irányító üzenet által. A hálózat indításakor, az inicializáló fázisban, a master-egységek egy logikai gyűrűbe rendeződnek. Minden master korlátolt ideig tarthatja a közvetítési jogot, amely idő alatt elküldheti a saját puffereiben várakozó üzeneteket. A slave-egységek nem kezdeményezhetnek közvetítéseket, viszont válaszolhatnak a masterek kéréseikre.

A három Profibus-változat egy széles sávú alkalmazástípusra alkalmas. A FMS (Fieldbus Message Specification) főleg a magas szintű kommunikációt támogatja. Ez a protokoll tartalmaz úgynevezett „profilokat”, amelyek megszabják a különböző automatizálási eszköz-típusokkal folytatott párbeszéd formáját. Például léteznek profilok motorvezérlő egységeknek, hőmérő-érzékelőknek, felvevő egységeknek stb.

A Profibus DP (Device Protocol) egy korlátoltabb változat, viszont hatékonyabb a kis- és közepes méretű vezérlőrendszerekben. Például a masterek száma háromra korlátolt. Az AP (Automation Protocol) változatot a robbanásra veszélyes környezetekre fejlesztették ki. Az alacsony villamos energiával végzett közvetítés biztosítja a szikramentes működést még abban az esetben is, ha a huzalokban rövidzárlat keletkezik.

A WorldFIP egy központosított sínrendszer. A hálózatban létezik egy egység, amely felelős a kommunikáció irányításáért. Ez az egység, előre meghatározott terv alapján, periodikusan lekérdezi a többi egységeket a folyamatparaméterek pillanatnyi értékei felől. A válaszokat bármilyen egység, amely a hálózatban van, hallhatja. Ilyen formában egy paraméternek van egy „termelője” (az egység, amely adja a választ) és lehet egy vagy több fogyasztója (akiknek szükséges a válasz). Létezik egy olyan eljárás is, amely lehetővé teszi a véletlenszerű események közlését. Az üzeneteket, amelyek az ilyen eseményeket hordják, kisebb prioritással kezeli a protokoll.

A P-NET ugyancsak egy multi-master protokoll, amely hatékonyan oldja meg a hálózathoz való hozzáférést egy „idő-token” által. Az idő-token egy időszület, amelyben egy bizonyos egységnek joga van igényelni a hálózatot közvetítés céljából. Minden master-egység kap az inicializálási fázis folyamán egy ilyen időszületet. Ha egy egység pufferében vannak várakozó üzenetek, akkor az időszület érkezésekor lefoglalja a hálózatot, és üzeneteket küld egy meghatározott ideig. Az előnye ennek a rendszernek abban rejlik, hogy a token (közlési jog) nem veszhet el, mint például a klasszikus token-passing hálózatokban, és ezért nem szükségesek bonyolult token-visszaállító eljárások.

A LONWorks egy komplex protokoll, amely az ISO-OSI (International Standard Organisation – Oper System Interchange) hét szintes modellre épült. Ennek az az előnye, hogy a hálózatot könnyen lehet az Internethez kapcsolni, és ezáltal a vezérelt folyamatot az Interneten keresztül lehet figyelni és irányítani.

Az utolsó csoporthoz, a *sejt hálózatokhoz* tartoznak azok az ipari hálózatok, amelyek komplex ipari folyamatok automatizálására alkalmasak. A sejt szó onnan ered, hogy a hálózat egy ipari folyamat flexibilis gyártási sejtjeit köti össze. Például egy szerelővonalon létezhet több gép, manipulátor, szállítószalag vagy akár önálló robot; mindezeket összhangba kell hozni és egységesen irányítani egy bizonyos optimális kritérium alapján. A sejhálózatok kezelni tudják azt a kommunikációt, ami egy ilyen rendszerben szükséges. Ezek a hálózatok legközelebb állnak a helyi (számítógép) hálózatokhoz, viszont tartalmaznak olyan plusz elemeket, amelyek által nő a megbízhatóság és korlátozt az üzenetek közvetítési ideje. Ehhez a családhoz a MAP és a TOP protokoll tartozik; az elsőt a General Motors cég fejlesztette ki, a másodikat meg a Boeing cég; mind a kettőt szerelővonalak automatizálására dolgozták ki.

Az utóbbi években egyre inkább szükségessé vált az ipari hálózatok egyeztetése a számítógép-hálózatokkal. Egyrészt ez azért szükséges, mert a gyártási folyamatokat integrálni kell a cég többi funkcionális egységeivel (pl. könyvelőség, tervezés, menedzsment, beszerzés, eladás stb.), ahol hagyományos hálózatokon folyik az adatközlés. Másrészt manapság egy cégnek lehetnek elszórt gyártási egységei (fiókjai) több helységben, sőt több kontinensen. Ezen egységek hatékony és időbeli menedzsmentje Internetes hozzáférési lehetőséget igényel. Sajnos a mai Internetes technológiák még nem nyújtják azt a megbízhatósági szintet, ami egy ilyen rendszer felépítéséhez szükséges volna. A remény abban rejlik, hogy egyre inkább más fajta alkalmazások is jóval megbízhatóbb minőség-ellenőrző eljárásokat igényelnek az Internetes kommunikációtól. Például a banki ügyletek nagyobb biztonságot igényelnek, vagy a multimédia alkalmazásokban garantált periodikus közvetítésre van szükség.

### **Következtetések**

Az ipari és vezérlési alkalmazások sajátos igényeket követelnek a kommunikációs médiumtól. Emiatt dedikált protollokat szükséges használni ahhoz, hogy ezeknek a kérelmeknek a betartását garantálni lehessen. Az ipari hálózatok egy megoldást nyújtanak ebben a kérdésben. Viszont ezen a területen még sok a megoldatlan feladat, mint például:

- az ipari hálózatok bekötése hagyományos (számítógépes) hálózatokba
- a kritikus hálózati tulajdonságok analitikus kiértékelése
- az ipari hálózatok egységes szabványosítása.



## Irodalom

- [1.] Cavalieri S. , Di Stefano A. , Mirabella O. , “ Mapping Automotive Process Control on IEC/ISA Fieldbus Functionalities”, Computers in Industry (ELSEVIER), No. 28, 1996, pp 233-250
- [2.] Cardeira C. , Simonot-Lion F, Bayard M. , “ Intelligent Field Devices and Field Buses: Impact on Applications Design Methodology”, Studies in Informatics and Control Vol 4 No. 3 1995, pp 255-262
- [3.] Fredriksson L.B. “ Controller Area Networks and the Protocol CAN for Machine Control Systems”, Mechatronics , Vol. 4 No.2 pp 159-192, 1994
- [4.] Jacobson J., Johansson L. , Lundin M. , “Safety of Distributed Machine Control Systems”, Swedish National Testing and Research Institut, Boras, Sweden , 1996
- [5.] Lennartsson K. , Fredriksson L, “Fundamental Parts in SDS , DeviceNet and CAN-Kingdom, a brief comparission”, 2nd International CAN conference 1995
- [6.] Papadopoulos, G., Koubias S., Leventi S., “A new Protocol for a Distributed Loop Communication Network”, Advances in Communications, Vol 1, pp 297-304 , Patras, Gece,1980
- [7.] Pasadas R., Almeida L., Fonseca J. , “A Proposal to Improve Flexibility in Real-Time Fieldbus Networks “ IFAC SICICA '97 , 3rd IFAC Symposium on Intelligent Components and Instruments for Control Application , Annesy, Franta,1997
- [8.] Pasadas F., Cardeira C., “Mechanisms to Improve Response time for Time-Critical Aperiodic Traffic within FIP Fieldbus”, 3rd IFAC Symposium on Intelligent Components and Instruments for Control Application , Annesy, Franta,1997
- [9.] Pasadas F., Cardeira C., “ Real-Time Protocols for Industrila LANs”, in Proceedings of the Network of Excelence in Intelligent Control and Integrated Manufacturing Systems. , Portugal , 1995
- [10.] Rauchaupt L. “ Performance Analysis of CAN Based Systems , 1st International CAN Conference, Mainz, 1994
- [11.] Sebestyen G., Buzas G. “ Intelligent Node Based Distributed Control System”, FieldComms Boston USA, 1997
- [12.] Ghe. Sebestyen , G. Buzas „Intelligent Node-Based Distributed Control System” FieldComms '97 - The Industrial Networking Conference Boston SUA, 1997 track4
- [13.] [Ghe. Sebestyen „Real-Time Communications through Industrial Networks” Proc. A&Q'98 International Conference on Automation and Quality Control, Cluj-Napoca, 1998, p A243-249
- [14.] Ghe. Sebestyen, G. Buzas „Internet Technologies for Remote Process Control and Supervision” OSPMA-FieldComms99 Conference, Open Solutions for Process and Manufacturing Automation, London, UK, 2000



- [15.] Ghe. Sebestyen, K. Puzstai "Web-Control - Distributed Control System Implementation through Web Technologies" - International Conference on Quality and Automation, Q&A2000, Cluj-Napoca, 2000,
- [16.] Ghe. Sebestyen "Modeling And Simulation Of Industrial Networks With Timed Petri Nets" - International Conference on Quality and Automation, Q&A2000, Cluj-Napoca, 2000,
- [17.] Tindell K., Hansson H., Wellings A.J. "Analysing Real-Time Communication: CAN" 1'st International CAN Conference 1994, Mainz
- [18.] Tindell K., Burns A., "Guaranteeing Message Latencies on CAN", Proceedings of 1st International CAN Conference, Mainz, 1994
- [19.] Upender B., Dean A. , "Variability of CAN Network Performance" Proceedings of the 3rd International CAN Conference , Paris, 1996

**A mobil eszközök  
és az objektum-orientált programozás:  
A Java2ME**

*Mobile devices and the object-oriented programming  
language: Java2ME*

**Varjasi Norbert**

Széchenyi István Egyetem, Számítástechnika Tanszék

**Abstract**

*The spreading of Java enabled devices in everyday life and the effective object-oriented softwares written for these devices can mean new directions and possibilities in IT training in the future. In my presentation I am going to give an overview of the structure of Java Micro Edition, which has been created for the development of mobile applications, the application models based on it, and finally how it is used in education at our university.*

Keywords: *MIDlets, wireless programming, MIDP programming*

**Összefoglaló**

A Java-képes eszközök mindennapos elterjedése és ezen eszközökre írt hatékony objektum-orientált programok (a továbbiakban mobil-oo, MIDlet) az informatikus képzés új irányait és lehetőségeit jelenthetik a jövőben. Az alábbiakban bemutatom a mobil alkalmazások fejlesztésére megalkotott Java Micro Edition felépítését, az erre épülő alkalmazás modelleket és végül egyetemünkön az oktatásban megvalósított formáit.

Kulcsszavak: MIDletek, vezeték nélküli fejlesztés, mobilprogramozás, MIDP

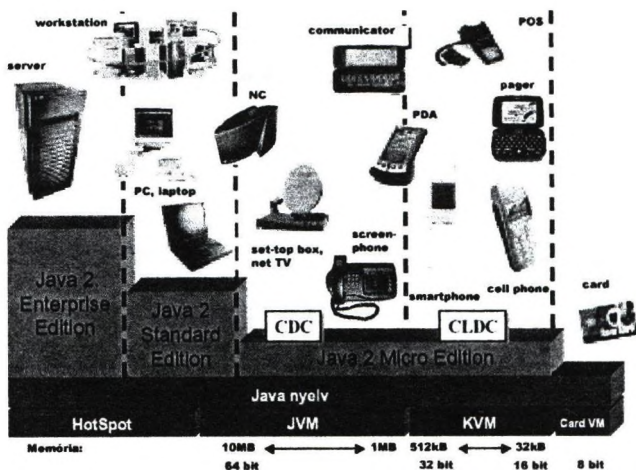
**A Java2 Micro Edition fejlődése és architektúrája**

A mobiltelefonok jelenlegi fejlődési ütemét felismerve a Sun Microsystems 2000 nyarán bejelentette a MIDP (Mobil Information Device Profile) 1.0-s verzióját[1]. Ennek keretében elsőként fogalmazott meg egy szabványos, platformfüggetlen és skálázható mobiltelefon szabványt. Ezzel

sikerült a fejlesztők és gyártók érdeklődését visszaterelni a Java megoldások felé, és szinte teljes mértékben meghódította a piac ezen szegmensét, hiszen mára már többmillió készülékbe implementálták a Java szoftvert. Az első verzió megjelenésével eleinte csak egyszerűbb alkalmazásokat, játékokat és segédprogramokat fejlesztettek (mint a kalkulátorok, e-mail olvasók, a tetris, az aknakereső stb.) de fellelhetők komolyabb algoritmusokat megvalósító útvonaltervező, tőzsdei elemző és MP3 lejátszó programok is.

A MIDP 2.0 fejlesztésébe már egyre több mobilgyártó cég bekapcsolódott[2], megalakítva a Java Community Processt (JCP). Így 2002 novemberére, a 2.0-s verzió bejelentésekor már 49 gyártó cég volt tagja a JCP-nek. A hardver és szoftver fejlesztések összehangolásával a mobil eszközök grafikai és audio képességei maximálisan kihasználhatóvá váltak, és sokat fejlődött az alkalmazások biztonsága is.

A Java2 ME fejlődése a konkurens termékekkel szemben a nyílt forráskódnak, a szabványosított fejlesztői felületnek köszönhető, mert az egyedi készülékek egyedi megoldásaival szemben szabványos és hordozható alkalmazások fejleszthetők, hiszen a Java alapelvei szerint a megírt szoftver bármilyen készüléken futtatható, függetlenül attól, hogy milyen operációs rendszer és milyen processzor működik alatta.



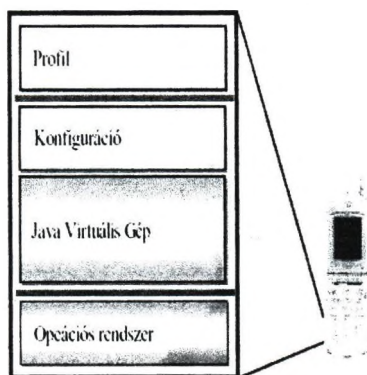
13. ábra  
A Java2 kiadásai és a céleszközök.

Maga a Java2 ME nem önálló szoftver, hanem a „kisméretű elektronika cikkek” piacára tervezett technológiák és szabványok gyűjteménye. A plat-



formáját a központi Java könyvtárak és két –különböző termékekhez fejlesztett– konfiguráció adja. Ezek a Connected Device Configuration (CDC) és a Connected Limited Device Configuration (CLDC)[3]. A CDC-t a nagyteljesítményű hordozható készülékekhez (kommunikátorok, Net TV dobozok), míg a CLDC-t a kisebb teljesítményű mobiltelefonokhoz, személyhívókhoz tervezték. A CDC a hagyományos Java Virtuális gépet használja, míg a CLDC a K Virtuális gépet, melynek neve arra utal, hogy a mobil eszközökben a rendelkezésre álló erőforrások néhány 10 kb-ja korlátozódnak. A KVM olyan 16/32 bites RISC/CISC processzorokkal működik együtt, melyek 128 kb-ot tárolják a virtuális gépet az osztálykönyvtárakkal és további 32 kb-ot áll a futásidejű adatok rendelkezésére.

A konfigurációkon alapulnak, és ezekre épülnek az egyes készülékkategóriákat meghatározó profilok. Jelen pillanatban a CLDC-re épülő MIDP van használatban de már fejlesztés alatt áll a PDA Profil is.

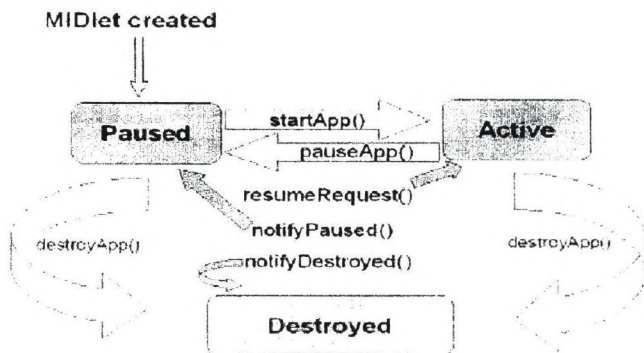


2. ábra

*A J2ME szabvány logikai szerkezete. Az eszköz operációs rendszere felett futó virtuális gép (VM) az erre épülő konfiguráció (CDC/CLDC), majd a profil réteg (MIDP/PDAP) alkotja.*

A program-elnevezési tradíciók szerint a CLDC-t és MIDP-t használó Java alkalmazásokat MIDleteknek nevezik. A MIDletnek céleszközei tehát olyan mobil eszközök, melyek minimum 96x54 pixeles kijelzővel, billentyűzettel vagy érintőképernyővel, vezeték nélküli összeköttetéssel és minimum 160Kb-ot memóriával rendelkeznek. A MIDletnek „jar” fájlba csomagolható és szabványos leírófájlok vezérlik a helyes működésben. A mobil eszközre feltöltött MIDletnek mindegyike rendelkezik egy-egy startApp(),

pauseApp() és destroyApp() metódussal, melyek az alkalmazás életciklusát vezérlik.



3. ábra  
Egy MIDlet életciklusa[2]

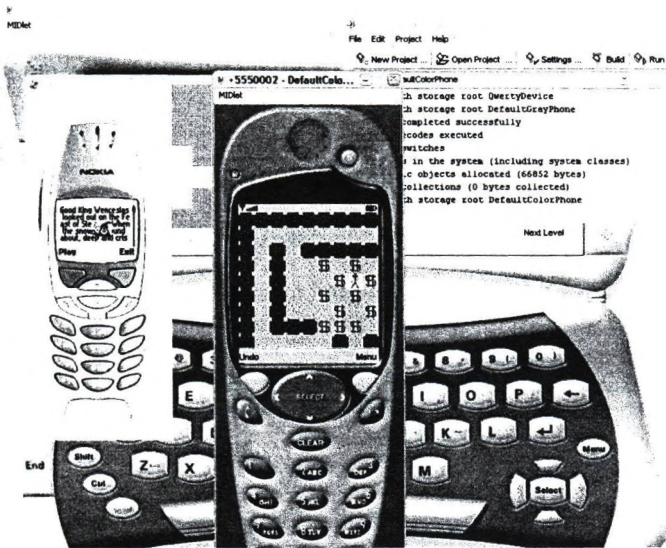
### Megvalósítások az oktatásban

A hagyományos programozási módszerek és a „klasszikus” objektumorientált programfejlesztés mellett egyetemünkön lehetőségünk nyílt – a Nokia támogatásával – a beágyazott programozó szakirányos képzésbe beilleszteni a Java 2 Micro Editiont. Az oktatásnak természetesen nem célja, hogy a piaci szereplőket és az aktuális divatirányzatokat kövesse. Azonban az olyan szakemberek képzése, akik a padokból kikerülve a megszerzett tudást hatékonyan tudják alkalmazni, szükségessé teszi a az objektumorientált képzés kiterjesztését olyan platformok felé, mint a mobil eszközök (intelligens kártyák, személyhívók, mobiltelefonok, PDA-k stb.). A hallgatók olyan fejlesztési irányokban vesznek részt szívesen, melyeket maguk is hasznosnak látnak. A szakmai fórumokon és a fiatal, már a mobilkultúrában nevelkedett tinédzserekben manapság igen élénk az érdeklődés a mobil programozás és a MIDletek iránt. (Debrecenben általános iskolai szakkörök keretében is oktatják[4]).

Az ingyenesen beszerezhető Java vezetéknélküli fejlesztőcsomag (Java Wireless Toolkit[6]), és a hozzá letölthető emulátorok[5] segítségével olyan alkalmazások készíthetők, melyek:

- magas absztrakciós szintet képviselnek,
- eseményvezéreltek,

- előtérben áll a funkcionalitás,
- hatékony algoritmusokra ösztönöz,
- figyelmes és pontos tervezést igényelnek,
- alkalmasak hálózati, multimédiás működésre,
- nagyfokú biztonsággal ruházhatók fel,
- hordozhatók az egyes eszközök között.



4. ábra

*A Sun vezeték nélküli fejlesztőcsomagja és az emulátorok*

A hallgatók körében magas fokú érdeklődéssel (90%-os óralátogatás), mély ötletgazdagsággal, önálló fejlesztési célokkal folyt le a meghirdetett kurzus. A motiváltság és a téma aktualitása bizonyította a mobil-oo technológiák létjogosultságát az informatikusok képzésében.

A témakör fiatalsága miatt főleg csak angol nyelvű szakirodalom és néhány magyar nyelvű cikk áll rendelkezésre.

## Összegzés

A XXI. század mindennapos eszközei lettek a mobiltelefonok, és a fel-növekvő generációk érdeklődéssel fordulnak nemcsak használatára, hanem a



fejlesztések felé is. A hatékonyabbá és olcsóbbá váló készülékek elterjedésével a mobil eszközök terén otthonosan mozgó szakemberekre lesz szükség. A könnyen átlátató és elsajátítható osztályhierarchia, a rendelkezésre álló szoftverkörnyezet új irányokat, és lehetőségeket nyújt az objektum orientált technológiák elsajátítása és a fejlesztések előtt. Az erőforrások korlátozottsága módszertani szempontból ösztönzőleg hat a hallgatókra, mert a kitűzött feladatok és célok csak hatékony és kompakt megoldásokkal oldhatóak meg.

A mobil eszközök szerepe napról-napra nyílik meg az újabb lehetőségek, mint a kliens-szerver kommunikáció, a nagy sávszélességű és megbízható kapcsolatok, az on-line szolgáltatások előtt, és a piaci érdeklődés ebben a szektorban nagyon jelentős.

### Irodalomjegyzék

- [1.] J2ME White Paper on KVM and the Connected, Limited Device Configuration (CLDC), Sun Microsystems, Inc. May 19.2000.
- [2.] Mobile Information Device Profile, <http://java.sun.com/products/midp/>
- [3.] J2ME Step by Step <http://www.ibm.com/developerWorks/>
- [4.] Bártfai Norbert: Jávacska projekt, <http://www.clib.dote.hu/javacska/>
- [5.] Java2 Platform Micro Edition, Wireless Toolkit, <http://java.sun.com/products/j2mewtoolkit/>
- [6.] Nokia Mobile Toolset (<http://www.forum.nokia.com/>)

## Hasznos tudnivalók

### Címek

- Bethlen Kata Diakóniai Központ: str. Ponorului nr. 1
- Confort Hotel: Calea Turzii nr. 48
- Báthory István Elméleti Líceum: str. M. Kogălniceanu nr. 2

### Telefonszámok

- Magyar Főkonzulátus, Kolozsvár: 0264-596300
- Bethlen Kata Diakóniai Központ: 0264-440510
- Confort Hotel: 0264-598410
- EMT mobil: 0744-783237
- Matekovits Hajnalka: 0722-570868

### Étkezések

A reggelit mindeki a szálláshelyén fogyasztja el.

A csütörtöki és pénteki vacsora, a szombati ebéd és az állófogadás helyszíne a Bethlen Kata Diakóniai Központ.

A pénteki ebéd Torockón a Tóbiás-házban lesz.

### A pénteki kirándulás útvonala

Kolozsvár – Torda – Tordai Sóbánya – Torockó – Torockószentgyörgy – Kolozsvár.

## A tordai sóbányászat rövid ismertetése

Az Erdélyi-medence kősógazdagsága az ókortól ismert. Az itteni sóbányászatra vonatkozó dokumentumok már Dácia római hódoltsága korából – akár Torda területéről előkerült sírfeliratok révén is – említhetők. Minden valószínűség szerint azonban, a felszíni sóelőfordulások fejtevése sokkal korábbi.

A só gazdasági jelentősége a középkori és újkori Erdély szempontjából óriási, hisz a legnagyobb állami bevételt biztosító forrás volt. Ez a magyarázata, hogy mindenkor királyi, fejedelmi vagy állami monopóliumban volt a sótermelés és -kereskedelem. Nagyrészt ennek a jövedelemnek köszönhető a két nagyhatalom közt önállóan megmaradt Erdélyi fejedelemség léte is. Ez a kiemelt gazdasági fontossága a só bányászatának csak a XIX. század folyamán vált másodrendűvé.

A só, mint kicsapódási üledékes kőzet, az Erdélyi-medence területén a középsőmiocén (ezen belül a középsőbáden, ú.n. wielicziai alemelet) ideje alatt (k.b. 15-16 millió évvel ezelőtt) képződött. A geológus társadalom ma is vitatja keletkezési feltételeit: mármint azt, hogy mélytengeri vagy lagúnáris környezetben rakódott-e le. A feké és fedő üledékek egyértelmű mélyvízi jellege mindenképp a második esetet valószínűsíti. A partmenti, sekélyvízi térségre inkább a gipsz képződése a jellemző (így Mészkö vagy Tordatúr határában). Az azonban minden keletkezéselméletben közös, hogy forró, csapadékszegény éghajlat határozta meg ezt a nagyméretű kicsapódási folyamatot.

A középsőmiocén Kárpát-medencebeli sóképződési folyamat következtében, csak az Erdélyi-medence területén lerakódott só összmenyisége meghaladja a 4100 km<sup>3</sup>-t (!). Ez az irdatlan mennyiségű só igen egyenlőtlen vastagságban oszlik meg, megközelítőleg 16300 km<sup>2</sup> területen. Az egyenlőtlen vastagság elsősorban nem az egykori képződési feltételekből adódik, hanem a kősó kristályszerkezete által meghatározott plaszticitásából, az egyenlőtlen fedőnyomás és a laterális tektonikai feszültségek következményeként alakult ki. A só tehát ott éri el legnagyobb vastagságát, ahol a fedőrétegek csekélyebb terhelése irányában, az oldalirányú nyomás serkentő hatása következtében migrált. Ez a migráció az oka az Erdélyi-medence területén kialakult több mint 80 diapírredő keletkezésének. A diapírredő olyan sómaggal rendelkező rétegboltozódás, amelynek kialakulásában a só előbb vázolt természete játsza a fő szerepet, s melynek során a só eredeti vastagsága helyileg megtöbbszöröződik, miáltal a fedő rétegeket felboltozza, elvékonyítja, de össze is törheti és át is szakíthatja. Sok esetben az így képződött diapírredő csúcsát az erózió lemetszi, így kerül a só felszínközelsébe.



Egy ilyen, erózió metszette diapírrdő a tordai sőtömsz is, amelynek két felszínre került nyulványa különíthető el. Egyik, a délebbi, a város központjától K-re, a Magyarós-patak völgyfőjében, a sósfürdők, azaz a római, majd koraközépkori sóbányák helyén azonosítható. Itt a só vastagsága meghaladja az 1,2 km-t. Nem sokkal kisebb a vastagsága a másik kibúvásnak sem, mely az előbbitől ÉÉK-re, a Bánya-hegy peremétől kezdődően, főként a Kissós-patak völgyét foglalja el. Ezen a részen található a közép, de főleg az újkori sóbányák.

Torda város, akár az egykori Potaissa léte és gazdagsága ennek a sónak köszönhető. Az erdélyi só bányászatát és kereskedelmét felügyelő Tordai Sókamarát már 1075-ből említik az okiratok. Az első középkori írásos emléke az itteni sóbányáknak azonban csak 1271-ből származik (ami korábbi létüket és működésüket távolról sem korlátozó adat). A város már 1291-ben kiváltságokkal rendelkezett. Nem véletlenszerűen, hisz a középkorban, majd a korai újkorban a legjelentősebb sókitermelő központja volt Erdélynek. Jelentőségét csak az előnyösebb termelési feltételeket biztosító Marosújvár melletti újkori fejtésnek megindításával (1871) – már modernebb fejtési módszerekkel (paralelepipedonkamrás fejtés) – veszítette el. Később, 1932-ben, a désaknai só jobb minősége miatt zárták be végleg a tordai, ipari méretű sófejtés kapuit.

A Kissós-patak völgyében 5 bánya maradt meg. A régebbiek: a Terézia-, Antal- és József-bánya harangkamrás fejtésűek. Hasonló volt a beomlott Karolina-bánya is. A későbbiek: a Rudolf- és Gizella-bánya már paralelepipedonkamrás fejtésű, gazdaságosabb eljárás szerint mélyített.

Ma a József-bánya látogatható, mely 110 m mélységet meghaladó, harang alakú, impozáns kupolatermet képez, egy 90 m magasságban levő körfolyósóval, ahonnan lépcsőn le lehet ereszkedni a fenéken lévő vajtérig. A lépcső mentén sóba faragva megtaláljuk a különböző éveken elért fejtési mélységértékeket. Ez a bányater különös földalatti visszhangjáról is nevezetes. Ma oldalról, a később hajtott hosszú ereszkén (mely járat a só fedőréteget is harántolja) jutunk a bányához. Egykoron azonban a fejtők a mennyezet közepén, lovakkal hajtott csörlőkre tekert kötélben, bőrülésben közlekedtek a mélybe. Ezen az úton emelték ki a kitermelt sót is.

*Wanek Ferenc*

## Kolozsvár – Torda – Torockó

Kolozsvártól 8 km-re található a Feleki Tető (730 m). Erdőfelek (Feleacu) román lakóit Nagy Lajos, magyar király telepítette ide Moldvából (1360). A falu XV. századi ortodox templomát Ștefan, moldvai fejedelem építtette (1460) gótikus stílusban. A falu határában vannak a „feleki gömbök”.

15 km után Tordatúr (Tureni) régi magyar település. A Rákos patak völgyében: Túri-hasadék, 5 km emelkedő Dobogókő (Dăbăgan), az utolsó balra Ajtony, jobbra Koppánd falvak.

Torda (Turda), Kolozs megye második városa 2000 éves sóközpont. A rómaiak idején Potaissa nevű castrum és polgárváros állt itt. A XI. században itt épül az első magyar királyi vár Aranyosvár néven.

Ótorda felett vannak a sóbányák – itt új város, Újtorda alakult.

Az önálló Erdélyi Fejedelemség idején Tordán 127 országgyűlést tartottak. A legjelentősebb az 1568-as – a vallásszabadság meghirdetése. Tordát elhagyva az Aranyos folyó völgyében nyugat felé haladunk.

Szentmihályfalva (Mihai Vitezul), Mészkö (Cheia), Sinfalva (Cornești) és Várfalva (Moldovenești) Árpád-kori települések. A térség magyar falvai a székely Aranyosszékhez tartoztak, melyet a XIII. században szakítottak ki Torda vármegyéből.

Jobbra, Mészkö falu felett a Tordai-hasadék, melyet a Hcsdát patak szel át. A hasadék 3 km hosszú, falai 200-300 m magasak. A hasadék és környéke (100 ha) 1950-től természeti rezervátum.

20 km után Borév (Buru), itt kezdődik a mőcvidék – az erdélyi Érchegeység – nyugat felé, a Nagyenyed felé vezető völgyben található a két híres magyar település Torockó (Rimetea) és Torockószentgyörgy (Colțești), melyek felett a 1130 m-es Székelykő áll. A hegyen már a római-korban katonai támpont, őrhely állt. A vasbányákat a rómaiak használták. A VI. században szlávok éltek itt, a falu neve a troszk (vaskő) vagy treszkava (zúgó patak) szláv szavakból származtatható. A X. században magyarok telepednek ide (Ákos nemzetsége), majd a XII. században székelyek keverednek az itt élőkkel, 1284 után V. István a hegyen lévő várat a kézdiszékelyeknek adományozta, ezért nevezik Székelykőnek.

Még 1243-1245 között felső Ausztriából német bányászok kerültek ide – a település 1291-ben városi kiváltságot kapott.

Torockót a századok során számos veszedelem fenyegette. Legnagyobb elgenségük a szomszédos Szentgyörgy hűbérura – a Toroczky család, akikkel 500 évig pereskedtek. 1702-1704 között a labancok kétszer is feldúlták. 1848-ban a Muntean Gligor vezette román szabadcsapatok rabolták ki. A vasbányászat és kohászat megszűnésével a település hanyatlásnak indul, szegényes határában csak szívós munkával lehet megélni.

*Deák Árpád*

## Résztevők névsora

- Bakó László** Sapientia EMTE  
540053 Tg. Mureş (Marosvásárhely)  
P-ja Trandafirilor nr. 61  
tel./fax: +40-265-231786  
e-mail: lbako@ms.sapientia.ro
- Benedek Dezső Dr.** University of Georgia  
Comparative Literature Dep.  
131 Joseph Brown Hall  
Athens, GA 30602, USA  
e-mail: cmsdezso@uga.edu
- Borbély Endre Dr.** Budapesti Műszaki Főiskola  
Kandó Kálmán Villamosmérnöki Főiskolai Kar  
1084 Budapest, Tavaszmező u. 17  
Tel: +36-1-2101415/255  
e-mail: borbely.endre@kvk.bmf.hu
- Brem Walter** EMT  
400750 Cluj (Kolozsvár) CP 1-140  
tel./fax: +40-264-590825  
e-mail: walter@emt.ro
- Budai László** Sapientia EMTE  
540203 Tg. Mureş (Marosvásárhely)  
str. Aradului nr. 3  
tel.: +40-741-051081  
fax: +40-265-213786  
e-mail: lbudai@ms.sapientia.ro
- Bujdosó Gyöngyi** Debreceni Egyetem Informatikai Intézet  
4010 Debrecen Pf. 12  
fax: +36-52-416857  
e-mail: ludens@inf.unideb.hu
- Dávid László Dr.** Sapientia EMTE  
540053 Tg. Mureş (Marosvásárhely)  
str. Trandafirilor nr. 61  
tel.: +40-265-213726  
e-mail: ldavid@ms.sapientia.ro
- Égető Albert** Gábor Áron Szakközépiskola  
525400 Tg. Secuiesc (Kézdivásárhely)  
str. Nouă nr. 17/19/D/19  
tel.: +40-267-363684  
e-mail: egetob@myx.net



- Égető Éva Cristina** 525400 Tg. Secuiesc (Kézdivásárhely)  
str. Nouă nr. 17/19/D/19  
tel.: +40-267-363684  
e-mail: egetob@myx.net
- Farkas Károly Dr.** Szolnoki Főiskola  
Budapesti Műszaki Főiskolák  
1112 Budapest, Bodajk u. 10  
tel.: +36-1-3194720  
e-mail: farkaskaroly@hotmail.com
- Fejér Magdolna** Bolyai Farkas Elméleti Líceum  
540370 Tg. Mureș (Marosvásárhely)  
str. Parângului nr. 29/20  
tel.: +40-265-266372  
e-mail: magdifej@yahoo.com
- Giese Piroska Dr.** KFKI RMKI  
1525 Budapest Pf. 49  
tel./fax: +36-1-3922279  
e-mail: giese@rmki.kfki.hu
- Görög Levente** Sapientia EMTE  
547218 Iceland, com. Ernei (Ikland) nr. 68  
tel.: +40-742-606038  
e-mail: goroglev@yahoo.com
- György Edit** Nagyernyei Átalános Iskola  
547215 Ernei (Nagyernye) nr. 156  
tel.: +40-265-335051  
e-mail: gyorgy\_edit@yahoo.com
- Haller Piroska Dr.** Petru Maior Egyetem  
540088 Tg. Mureș (Marosvásárhely)  
str. Nicolae Iorga nr. 1  
e-mail: phaller@upm.ro
- Házy Sándor** Inspectoratul Teritorial de Muncă Brașov  
500168 Brașov (Brassó)  
str. G-ral Dumitrache nr. 20/257/9  
tel.: +40-268-425621  
e-mail: shazy@xnet.ro, shazy@index.hu
- Héray Tibor Dr.** Széchenyi István Egyetem  
9026 Győr, Egyetem tér 1  
tel.: +36-96-503400/3186  
e-mail: heray@sze.hu
- Horváth Erika** EMT  
400750 Cluj (Kolozsvár) CP 1-140  
tel./fax: +40-264-590825  
e-mail: erika@cmt.ro

- Illyés Lajos Demeter** Nagyszalontai Elméleti Líceum  
415500 Salonta (Nagyszalonta)  
P-ța Unirii nr. 40/22/1  
tel.: +40-259-373869  
e-mail: lilyes@rdslink.ro
- K. Princz Mária** Debreceni Egyetem MFK  
4029 Debrecen  
Ótemető u. 2-4  
tel.: +36-52-415155/7730  
fax: +36-52-418643  
e-mail: pmaria@delfin.unideb.hu
- Kallós Gábor Dr.** Széchenyi István Egyetem  
9026 Győr  
Egyetem tér 1  
tel.: +36-96-503400/3102  
e-mail: kallos@sze.hu
- Kása Zoltán Dr.** Babeş-Bolyai Tudományegyetem  
Cluj (Kolozsvár)  
e-mail: kasa@cs.ubbcluj.ro
- Kátai Zoltán** Sapientia EMTE  
540398 Tg. Mureş (Marosvásárhely)  
Bd. 1848 bl. 58/30  
tel.: +40-265-227111  
e-mail: katai\_zoltan@ms.sapientia.ro
- Kiss Réka** Báthory István Elméleti Líceum  
400050 Cluj (Kolozsvár)  
P-ța Abator Bl. C2/16  
tel.: +40-264-535701  
e-mail: kissreka@hotmail.com
- Kovács Barna** Papiu Ilarian Kollégium  
540143 Tg. Mureş (Marosvásárhely)  
Aleea Cornișa nr. 10/7  
e-mail: t\_barna\_ro@yahoo.com
- Kovács Enikő** EMT  
400750 Cluj (Kolozsvár) CP 1-140  
tel./fax: +40-264-590825  
e-mail: eniko@emt.ro
- Kovács Lehel István** Babeş-Bolyai Tudományegyetem  
400462 Cluj (Kolozsvár)  
str. C. Brâncuși nr. 196/25  
tel.: +40-744-251245  
e-mail: klehel@cs.ubbcluj.ro

- Kovács Zoltán Dr.** Babeş-Bolyai Tudományegyetem  
Cluj (Kolozsvár)  
e-mail: kovzoli@phys.ubbuj.ro
- László Imola** Aurel Moşora Állami Gimnázium  
545400 Sighişoara (Segesvár)  
str. Caisului nr. 2  
tel.: +40-265-772706  
fax: +40-265-777432  
e-mail: limola2000@yahoo.com
- Majdik András** Kolozsvári Műszaki Egyetem  
400305 Cluj (Kolozsvár)  
str. G. Enescu nr. 9  
tel.: +40-740-135876  
e-mail: adogos@freemail.hu
- Matekovits Hajnalka** EMT  
400750 Cluj (Kolozsvár) CP 1-140  
tel./fax: +40-264-590825  
e-mail: hajni@emt.ro
- Máthé Zsolt** Kolozsvári Műszaki Egyetem  
540077 Tg. Mureş (Marosvásárhely)  
str. Corneşti nr. 42  
tel.: +40-740-351000  
e-mail: mathezsolt@acm.org
- Nagy Imecs Hunor** Sapiientia EMTE  
535600 Odorheiu Secuiesc (Székelyudvarhely)  
str. Kós Károly nr. 4/4  
tel.: +40-744-860383  
e-mail: sigmanet@ms.fx.ro
- Nagy Imecs Vilmos** Eötvös József Mezőgazdasági Szakközépiskola  
535600 Odorheiu Secuiesc (Székelyudvarhely)  
str. Tompa László nr. 12  
e-mail: nivos@freemail.hu
- Nagy Magdolna** David Prodan Iskola  
400367 Cluj (Kolozsvár), str. Lunii nr. 13/25  
tel.: +40-264-522172  
e-mail: nagyzs@xnet.ro
- Nagy Olga** Nagyszalontai Elméleti Líceum  
415500 Salonta (Nagyszalonta)  
str. Republicii nr. 27-29/AN3/7  
tel.: +40-259-371232  
e-mail: nagyolga@rdslink.ro



- Nagy-Kóródi László** Aurel Moşora Állami Gimnázium  
545400 Sighişoara (Segesvár)  
str. Crizantemelor nr. 36/6  
tel.: +40-265-776908  
fax: +40-265-771040  
e-mail: nklaszlo@yahoo.com
- Németh Géza Dr.** BME Távközlési és Médiainformatikai Tanszék  
1117 Budapest  
Magyar tudósok krt. 2  
tel.: +36-1-4633883  
fax: +36-1-4633107  
e-mail: nemeth@tmit.bme.hu
- Pap Tünde** EMT  
400750 Cluj (Kolozsvár) CP 1-140  
tel./fax: +40-264-590825  
e-mail: tunde@emt.ro
- Pukler Antal** Széchenyi István Egyetem  
9026 Győr, Egyetem tér 1  
tel.: +36-96-503400  
e-mail: pukler@sze.hu
- Pusztai Kálmán Dr.** Kolozsvári Műszaki Egyetem  
Automatika és Számítástechnika Kar  
400027 Cluj (Kolozsvár)  
str. Bariţiu nr. 28  
e-mail: kalman.pusztai@cs.utcluj.ro
- Sebestyén György Dr.** Kolozsvári Műszaki Egyetem  
Automatika és Számítástechnika Kar  
400027 Cluj (Kolozsvár)  
str. Bariţiu nr. 28  
tel.: +40-745-362211  
e-mail: gheorghe.sebestyen@cs.utcluj.ro
- Selinger Sándor Dr.** Gábor Dénes Főiskola  
Erdélyi Konzultációs Központ  
400474 Cluj (Kolozsvár)  
str. I.B.Deleanu nr. 64  
tel.: +40-264-431841  
e-mail: selinger@gdf.ro
- Somodi Zoltán** Kolozsvári Műszaki Egyetem  
400027 Cluj (Kolozsvár)  
str. Bariţiu nr. 28  
tel.: +40-264-401476  
fax: +40-264-594491  
e-mail: zoltan.somodi@cs.utcluj.ro

- Stanciu György** Axente Sever Elméleti Liceum  
551028 Mediaş (Madgyes)  
str. G. Topârceanu nr. 12/27  
tel.: +40-269-833227  
e-mail: stgyuri@yahoo.com
- Szalay Istvánné** KFKI RMKI Számítógép Hálózati Központ  
1525 Budapest Pf. 49  
tel./fax: +36-1-3922562  
e-mail: szalay@sunserv.kfki.hu
- Szász Tünde** Báthory István Elméleti Liceum  
400452 Cluj (Kolozsvár)  
Aleea Padiş nr. 3/39  
tel.: +40-740-957657  
e-mail: szasz\_tunde@yahoo.com
- Szél Sándor** Electrocentrale Rt.  
410083 Oradea (Nagyvárad)  
str. Jean Jaures nr. 23  
tel.: +40-259-238464
- Szilágyi Ferenc** Tanulók Palotája  
520008 Sf. Gheorghe (Sepsiszentgyörgy)  
str. Kossuth Lajos nr. 1/2/B/15  
tel.: +40-267-314988
- Tibád Zoltán** EMT  
400750 Cluj (Kolozsvár) CP 1-140  
tel./fax: +40-264-590825  
e-mail: angyal@emt.ro
- Törtely Éva** Templom Téri Általános Iskola Pilisvörösvár  
1126 Budapest  
Márvány u. 27/F/2  
tel.: +36-20-3640942  
e-mail: tortelyeva@freemail.hu
- Varjasi Norbert** Széchenyi István Egyetem  
9026 Győr, Egyetem tér 1  
tel.: +36-96-503400  
e-mail: varjasin@sze.hu
- Veres Levente** EMT  
400750 Cluj (Kolozsvár)  
CP 1-464  
tel.: +40-741-089500  
e-mail: bergermanus@yahoo.com

## Tartalomjegyzék – Content

A konferencia programja	4
e-kereskedelem, e-vásárlás e-sales, e-purchase <i>Borbély Endre</i>	10
Távoktatás a világhálón Az Internet alapú oktatás egy kommunikációs folyamat Distance Learning, a Communication Tool <i>Dr. Selinger Sándor</i>	21
Hogyan viselkedjünk az Interneten! Good Behaviour on the Internet <i>Kása Zoltán</i>	26
Információkeresési stratégiák a weben és tanításuk tapasztalatai Web Search Strategies and Experiences of their Teaching <i>K. Princz Mária</i>	28
A szöveg- és kiadványszerkesztés oktatásának néhány tapasztalata Experiences on Teaching Word Processing and Some Parts of Typography <i>Bujdosó Gyöngyi</i>	35
Elektronikus jegyzet a párhuzamos programozás oktatásához An Electronic Lecture-book for Studying Concurrent Programming <i>Dr. Kallós Gábor</i>	42
Hogyan tanítsuk a programozási technikákat? How Should We Teach Programming Technics? <i>Kátai Zoltán</i>	50
Mit adott nekem a Logo? What Have I Got by Logo? <i>Farkas Károly CSc</i>	57
Játékos Informatika – Logo és programozás Playful Informatics – Logo and Programming <i>Törtely Éva</i>	65



Számítástechnika tanár = Internet pedagógus !? (vagy csak a tanár dilemmája) Informatics Teacher is Equal to Internet Educator?! (Or it's Just Only a Teachers Dilemma) <i>Nagy Imecs Vilmos</i>	72
Mesterséges intelligencia társasjátékokban Artificial Intelligence in Board-games <i>Máthé Zsolt, Görög Levente, Komáromi Lóránd, Szilágyi Sándor Miklós</i>	83
Optimális vezérlési stratégia megvalósítása mesterséges neuronhálók segítségével Comparison Between LTI LQ Optimal Controller with Hierarchical Numerical Solution vs ANN Solution <i>Dr. Dávid László</i>	90
Fejlett neuronmodellek szimulációja és megvalósítása Simulation and Implementation of Advanced Neural Models <i>Bakó László, Brassai Sándor Tihamér</i>	98
A dinamikus öröklődés The Dynamic Inheritance <i>Kovács Lehel István</i>	108
A mikroprocesszoros rendszerek megjelenésének hatása a vasúti irányítórendszerekre Effects of Microprocessor Based Systems on the Railway Signaling Technology <i>Dr. Héray Tibor</i>	116
Prediktor – korrektor és konzervatív integrátorok az égi mechanikában A korlátozott háromtest-probléma integrálása prediktor-korrektor és konzervatív integrátorral Predictor – Corrector and Conservative Integration Methods in Celestial Mechanics The Numerical Integration of the Restricted Three Body Problem with Predictor – Corrector and Conservative Integrators <i>Kovács Barna</i>	127
Multimédiás adatátvitel teljesítményelemzése processz algebra segítségével Performance Modelling of the Multimedia Stream Controller Using Stochastic Process Algebra <i>Haller Piroska</i>	136

Tűzfalak – működési elvek, tervezésük és működtetésük egy oktatási hálózatban Firewalls – how they work, how they are built <i>Budai László</i>	144
DoS támadások elleni védelmi mechanizmusok az IPv6-ban Protection Mechanisms Against DoS Attacks in IPv6 <i>Somodi Zoltán</i>	150
Ipari kommunikációs hálózatok Industrial Communication Networks <i>Dr. Sebestyén Pál György</i>	158
A mobil eszközök és az objektum-orientált programozás: A Java2ME Mobile devices and the object-oriented programming language: Java2ME <i>Varjasi Norbert</i>	169
Hasznos tudnivalók	175
Résztevők névsora	179

## Jegyzetek





**EMT**

Kolozsvár, 1989. december 21. sugárút 116. szám

Postacím: 400750 Cluj, C.P. 1-140, România

Tel./fax: +40-264-590825; 594042; +40-744-783237

E-mail: [emt@emt.ro](mailto:emt@emt.ro)

Honlap: <http://www.emt.ro>

---

ISBN 973-86097-8-X