

NIJ
SI
ZA

*V. Országos
Kongresszusa*

*információ-
technológia '92*

II.



Debrecen
1992. június 21-24

ITA / 337/2.

ELŐADÁSOK

II. kötet

Debrecen

1992. június 21-24

A kiadványt szerkesztette:

dr.Bakonyi Péter - a Programbizottság elnöke

A "Kormányzati informatikai rendszerek" szekció előadásait tartalmazó rész a BM Választási és Informatikai főosztály támogatásával készült.

Neumann János Számítógéptudományi Társaság

V. Országos Kongresszusának

elnöke:

Vámos Tibor

akadémikus, a NJSZT tiszteletbeli elnöke

védnöke:

Hevessy József - Debrecen főpolgármestere

Programbizottság:

Bakonyi Péter (elnök)

Aladics Sándor

Demetrovics János

Gergely Csaba

Havass Miklós

Lajtha György

Majtényi György

Pongrácz Tibor



TARTALOMJEGYZÉK

I. kötet

PLENÁRIS ÜLÉS

	oldal
1. Vámos Tibor: Kooperatív számítástechnika - az emberi tevékenység jövendő szervezője	1
2. Kolosi Tamás: Az információtechnológia szerepe a társadalomtudományban.	9
3. Babai László: Áttetsző bizonyítások.	12
4. Horváth Pál: A távközlés fejlesztési stratégiája.	20
5. Pongrácz Tibor: A gazdaság és az információs politika.	26

I. szekció INFORMÁCIÓSZOLGÁLTATÁS

1. Enyedy Orsolya-Szepesváry Pál-Túri Gabriella: Az országos izotópnnyilvántartás.	29
2. Barta-Enyedy-Kaposi-Nagy-Vári: Az országos izotópnnyilvántartás megvalósítása HP-3000 és IBM PC számítógépekből álló hálózaton.	38
3. Turchányi Géza: Hogy legyen olcsó, ami ingyen van.	47
4. Sándori Zsuzsanna: Új információtechnológia és könyvtári arculat.	53

II. szekció KORMÁNYZATI INFORMÁCIÓK

	oldal
1. Lendvai János, Kertészné Gérecz Edit: A statisztikai rendszer átalakításának fő irányai és annak informatikai megvalósítása.	61
2. Kerepesi Károly: Lappal támogatott adatgyűjtés rendszere a KSH munkaerőfelvételében.	69
3. Csicsman József-Papp Péterné-Rudas Jánosné: Az új fogyasztói árstatistika számítástechnikai háttere.	75
4. Dr.Nagy Péter-Dr.Széphalmi Géza-Dr.Somodi Magdolna: A külügyi informatika fejlesztési irányai egy európai országban.	82
5. Dr.Nagy Péter: A külügyi integrált információs rendszer strukturális modellje.	96
6. Dr.Horváth Ferenc: Az állami népességnyilvántartás rendszere mint a kormányzati informatika része.	107

III. szekció OKTATÁS

1. Bakos Tamás: OKKER - egy magyar számítógépes oktatási keretrendszer.	112
2. Molnár János: Számítástechnikai műszerészképzés magyar-angol nyelven.	116
3. Varga Géza: Kézzel és/vagy CASE-zel?	121
4. Paál Jenő: Agrár információs modell építés.	127
5. Nagy Eleméerné-Nagy Elemér-Heves Csilla: Számítógéppel támogatott oktatás aktuális problémái.	133
6. Hegedűs Gy.Csaba: A CADI számítógépes képfeldolgozási oktató rendszer.	140

7.	Németh Róbert-Szabados László: Térinformatika az információtechnológia új dimenziója.	147
<i>IV. szekció Banki- és pénzügyi rendszerek</i>		
1.	Jánosi Julianna: Új Zsiro-rendszer a magyar bankvilágban.	153
2.	Kerékfy Pál: A magyar bankközi elszámolási rendszer (GIRO) központja.	161
3.	Bernáth Ákos: A GIRO rendszer kapcsolata a bankfiókokkal.	166
4.	Aszalós János - Laczay István: SARAH: A GIRO Help-Desk Szakértői Rendszere.	175
5.	Sághy András: A Magyar Nemzeti Bank számítógépes információs rendszere.	182
6.	Fejér Imre - Várady József: A Magyar Külkereskedelmi Bank egységes deviza és forint számla- vezetési rendszere a VAX számítógépeken.	191
7.	Horváth Jánosné - Pintér Zsuzsa: A kötelező gépjármű felelősségbiztosítás alapján rendezett károk adatainak statisztikai elemzése.	197
8.	Szabó László: Készen vásárolt nyugati ügyviteli szoftverek a bank- és pénzügyekben.	203
9.	Alföldi Ferenc - Szántó Tamás: Döntéstámogató rendszerek a stratégiai pénzügyi tervezésben.	212
10.	Kátai Szabolcs: SunSystems-Integrált pénzügyi-számviteli-áruforgalmi rendszer.	221

II. kötet

<i>V.szekció</i>	KUTATÁS-FEJLESZTÉS	oldal
1.	Almási Béla - Kuki Attila: Matematikai statisztikai módszerek a hálózat és szoftver- elemzésben.	2
2.	Kőrösi Gábor: ALL-EX/3 szakértői rendszerkeret.	13
3.	Arató Mátyás: Hatékonyságmodellezés.	14
4.	Benczúr András: Adatbáziskezelés.	16
5.	Kacsuk Péter - Ferenczi Szabolcs: Transzputer alapú párhuzamos számítási technológia és helyzete Magyarországon.	25
6.	Kacsuk Péter - Erényi István: Párhuzamos számítási rendszerek fejlődésének nemzetközi trendjei és várható hatásuk a magyarországi információ- technológiára.	35
7.	Bíró Miklós: Fejlett kommunikációs lehetőségek hatása az emberi döntés- hozatalra.	42
8.	Remsző Tibor: Hipertext és multimédia rendszerek.	51
9.	Csáki Zsuzsanna: Intervallum alapú tudásreprezentáció.	59
10.	Pekker Mária-Sáry Zoltán-Pősze Lajos: A DODS döntésobjektíváló és dokumentáló rendszer.	68
11.	Dr.Strajber Benedek-Dr.Bakó András: Adatkonverziós technológiák térinformatikai rendszerek fejlesztéséhez.	75

12. Herczeg István - Mag.Peter Graf:
Korszerű Fulltext rendszer UNIX alatt. 80

VI. szekció INFRASTRUKTÚRA

1. dr.Nagy Ákos - Szokolay Tamás:
Helyi hálózatok. 84
2. Seregdy Tamás:
AT&T SYSTIMAX^R PDS strukturált kábelezési rendszer. 92
3. Bertáné - Czigány - dr.Kántor - dr.Koós:
A vezeték nélküli távközlő rendszerek tervezésének
információ-ellátása terepadatbázisok segítségével. 98
4. Farkas Gábor-Ivánka Gabriella-Leporisz György:
Országos X.25 bázisu IBUSZ rendszer megoldásai és
technológiai tapasztalatai. 107
5. dr.Agócs-Dombrádi-Gál-dr.Herdon-Korcsolay-Kovács-dr.Krausz-
Sági-dr.Székely-dr.Terdik:
Debreceni Universitas Adatátviteli Hálózata. 116
6. Bérci Márton:
A nyilvános csomagkapcsolt adatátviteli szolgáltatás. 124
7. Gál Zoltán:
A Kossuth Lajos Tudományegyetem informatikai rendszere. 132

VII. szekció EGYÉB ALKALMAZÁSOK

1. Unyi Gábor:
BLAST, az igazi kommunikációs szoftver. 142
2. Jakó János-Kiss Csaba Zsolt-Molnár Tamás:
ORACLE Card: Grafikus eszköz a relációs adatbázis-kezelők
világában. 149

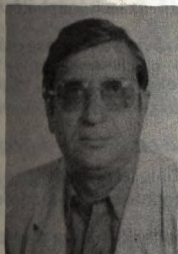
- | | | |
|----|--|-----|
| 3. | Hargitai Róbert:
Mentőállomások szimulációja SIMULA'67 nyelven. | 158 |
| 4. | Peller Róbert:
Az információ technológia fejlődésének néhány szempontja
szubjektív közelítésben. | 166 |
| 5. | Homonnay Gábor:
A nemzetközi élvonalhoz való felzárkózás esélyei. | 171 |
| 6. | Fazekas Béla:
A BasisPlus és alkalmazásai. | 173 |

ZÁRÓ PLENÁRIS ÜLÉS

- | | | |
|----|---|-----|
| 1. | Mojmir Baumgartner:
SAA AD/Cycle Status and Directions | 175 |
| 2. | Heinz Prokop:
Information technology strategies of international corporations
in the 90s. | 182 |
| 3. | Harold K. Wilson:
Strength and Diversity: The keys to growth in the 1990s. | 189 |
| 4. | Manfred Joseph:
Software for people who can't predict the future. | 195 |
| 5. | Ian J. Leach:
A nyílt rendszerek világa. | 198 |
| 6. | Alfons Demmler:
DOMINO - a Siemens-Nixdorf CASE-stratégiája. | 201 |

KUTATÁS-FEJLESZTÉS ("SZ V.")

Elnök: Demetrovics János



Az MTA SZTAKI Informatika Laboratóriumának vezetője, valamint az ELTE Általános Számítástudományi Tanszékének egyetemi tanára, az MTA levelező tagja.

A Moszkvai Állami Egyetem matematika szakát 1970-ben végezte el. Kandidátusi értekezését 1974-ben védte meg, végességértékű logikai témából. 1974 óta foglalkozik az adatbázisok elméletével; 1981-ben megvédett akadémiai doktori értekezésének tárgyát a relációs adatbázisok tervezésével kapcsolatos problémák köréből merítette.

Mintegy 160 tudományos közlemény szerzője, öt szakfolyóirat szerkesztő bizottságának tagja. Több konferencián szerepelt meghívott előadóként, a program bizottság tagjaként, illetve elnökeként. 1990-ben elnyerte a NJSZT Kalmár László díját.

MATEMATIKAI STATISZTIKAI MÓDSZEREK A HÁLÓZAT ÉS SZOFTVERELEMZÉSBEN

Almási Béla Kuki Attila

Kossuth Lajos Tudományegyetem
4010 Debrecen, Egyetem tér 1.

I. Számítógéphálózatok Matematikai Modellezése

Bevezetés

Ebben a fejezetben egy központi géppel (egy szerverrel) működő számítógéphálózat matematikai modellezésével foglalkozunk. A már jól ismert modellek rövid áttekintése után a számítógépek meghibásodhatóságát is figyelembe vevő modelleket vizsgálunk, végül néhány numerikus példán szemléltetjük az eredményeket.

A számítógéphálózatok, multiterminálos rendszerek sztochasztikus modellezésének alapelveit számos mű (lásd [1],[2]) részletesen tárgyalja. Foglaljuk össze röviden a modellek legfontosabb eredményeit:

1./ Homogén sorbanállási modell

A hálózatban n db terminál (munkaállomás) csatlakozik egy központi processzorhoz (CPU-hoz, szerverhez). A terminálok (amelyeket az egyszerűség kedvéért jelölhetünk az $1, 2, \dots, n$ számokkal) egymástól függetlenül, - átlagosan $1/\lambda$ - "gondolkodási idő" elteltével igényeket generálnak a szerver felé (programok futtatását kérik a szervertől), és mindaddig várakoznak, míg az igényüket a szerver - valamilyen kiszolgálási elv alapján ki nem szolgálja. A legismertebb kiszolgálási elvek a következők (részletes elemzésükre lásd pl. az [1], [2] irodalomban): FIFO, Időosztásos, Körbenjárásos, Prioritásos. A központi egység egy igény kiszolgálására - függetlenül attól, hogy az igény melyik terminálnál keletkezett - átlagosan $1/\mu$ időt fordít.

A rendszer állapotát minden időpillanatban a szervernél tartózkodó igények száma jellemzi ($0, 1, \dots, n$), melyeket matematikailag valószínűségi változókkal kezelhetünk. Feltételezve hogy a "gondolkodási" és "kiszolgálási" idők - mint valószínűségi változók - eloszlásuk exponenciális, a sztochasztikus folyamatok elméleti eredményei alapján meghatározhatjuk ezen állapotok bekövetkezésének stacionárius valószínűségeit (lásd [1], [2]).

valószínűségek meghatározásához egy $n+1$ ismeretlen tartalmazó lineáris egyenletrendszert kell megoldani.) A stacionárius valószínűségek ismeretében a legfontosabb rendszert jellemző átlagértékek könnyen kiszámíthatók. Pl.:

- A szervernél tartózkodó igények átlagos száma.
- A szerver átlagos kihasználtsága.
- A terminálok átlagos kihasználtsága.
- Egy igény átlagos válaszideje (várakozási idő + kiszolgálási idő).

2./ Inhomogén Multi-terminál rendszer

Az 1./ pontban leírt modell egy kézenfekvő továbbfejlesztése, ha az egyes terminálok "gondolkodási" és "igénykiszolgálási" idejének átlagértékét a termináltól függőnek tekintjük. Ebben az esetben a rendszert a szervernél tartózkodó igények száma mellett az igényeket generáló terminálok sorszámainak sorozata jellemzi. Az állapotok száma FIFO

kiszolgálás esetén :

$$\sum_{k=0}^n \frac{n!}{k!}$$

Az állapotok stacionárius bekövetkezési valószínűségeinek meghatározása ebben az esetben is egy lineáris egyenletrendszer megoldását igényli (lásd [3]), de az ismeretlenek nagy száma miatt a megoldás kiszámítása még számítógéppel is nehéz és csak viszonylag kevés terminálra kaphatunk egzakt megoldást. A stacionárius valószínűségek ismeretében az 1./ pontban említett rendszerjellemzők már könnyedén kiszámíthatók.

3./ A hálózati forgalom modellezése

Az 1./ és 2./ pontokban tárgyalt modellek elsősorban a csomópontok (szerver, terminálok) szemszögéből vizsgálták a hálózat működését. Nézzük meg most a csomópontok által használt átviteli közegnél felmerülő várakozásokat.

Az egyes csomópontok valamilyen elv alapján kisajátítják az átviteli közeget (a kisajátítási elvekről lásd a [4] irodalmat), s a kisajátítás időtartama alatt csomagokat forgalmaznak. Ha feltételezzük, hogy az átviteli közeg átlagosan $1/\lambda$ időközönként kap átlagosan $1/\mu$ bit hosszúságú csomagot és a csatorna átviteli sebessége C bit/sec, akkor az 1./ pontban tárgyalt modellhez hasonló matematikai modellt kapunk: csomagok állnak sorban az átviteli közeg előtt továbbításra várva. A legfontosabb rendszerjellemzők

kiszámítása is hasonló elven történik. A képlet egyszerűsége és gyakorlata interpretálhatósága miatt közöljük az egy csomag továbbítási (várakozási + átviteli) idejének átlagértékének kiszámítási formuláját:

$$T = \frac{1}{\mu C - \lambda}$$

4./ A számítógépek meghibásodása

Az 1./ pontban ismertetett modellnél nem beszéltünk a számítógépek meghibásodásáról (feltételeztük, hogy a gépek nem hibásodnak meg). Vizsgáljuk meg, hogyan kellene módosítanunk a modellt, ha a számítógépek meghibásodását is figyelembe akarjuk venni. Célszerű külön kezelni a szerver és a terminálok meghibásodását, mivel a szerver meghibásodása lényegében az egész hálózat működésképtelenségét eredményezi. A szerver hibamentes működési idejének átlagát jelöljük $1/\alpha$, a javítási idejének átlagát pedig $1/\beta$ szimbólumokkal. A meghibásodott terminálok javítását - valamilyen kiszolgálási elv szerint - egy javító egység, szervíz végzi, azaz várakozási sor alakulhat ki a javítás során is. Jelölje $1/\gamma$ a terminálok hibamentes működésének átlagidejét, $1/\tau$ pedig egy hibás terminál javításának átlagidejét (lásd a Numerikus példánknál). Célszerű a szerver javítását prioritással kezelni a terminálok javításához képest. A rendszert úgy képzelhetjük el, hogy a terminálok gondolkodási állapotban lehetnek, a szervernél levő igényük kiszolgálására várhatnak, illetve javításra, esetleg a szerver megjavítására várakozhatnak. Az így leírt működést matematikailag a következő valószínűségi változókkal kezelhetjük (homogén modell esetén):

- $X(t) =$ 0, ha a szerver működésképtelen a t időpillanatban
 1, egyébként
 $Y(t) =$ a szervernél kiszolgálásra váró igények száma a t időpillanatban
 $Z(t) =$ a hibás (javításra váró) terminálok száma a t időpillanatban

Az állapotok száma ebben az esetben $(n + 1)(n + 2)$. Jelentős változás még az 1./ pontban tárgyalt modellhez képest az is, hogy az állapotokat ebben az esetben háromdimenzióban ábrázolhatjuk. A modell részletes leírását lásd az [5] irodalomban.

Természetesen ezt a modellt is továbbfejleszthetjük és inhomogén rendszerként

vizsgálhatjuk (lásd [6]). Ebben az esetben azonban az állapotok számát (FIFO kiszolgálási elv esetén) a következőképpen számíthatjuk ki:

$$\sum_{k=0}^n (k+1) \frac{n!}{(n-k)!}$$

Ez a magas dimenziószám azt jelenti, hogy még számítógéppel is maximum 4-6 terminálra tudunk pontos megoldást adni. Számítógépes szimulációval azonban viszonylag rövid idő alatt megkaphatjuk a rendszerjellemzők közelítő értékét.

5./ Numerikus példák

a./ Nem meghibásodható rendszer jellemzői:

Terminálok száma: 4

Terminálparaméterek:

Átl. válaszidő

$\lambda = 0.5$

$\mu = 0.9$

$T = 2.8$

Rendszerjellemzők:

Szerver kihasználtság: 92%

Kiszolgálandó igények átlagos száma: 2.3

Működőképes terminálok várható száma: 4

Ezt a példát a következőkben ismertetésre kerülő - meghibásodást feltételező - numerikus eredményekkel való összehasonlításra használhatjuk fel.

b./ Terminálok meghibásodása :

Terminálok száma: 4

Terminálparaméterek:

Átl. válaszidő

$\lambda = 0.5$

$\mu = 0.9$

$T = 1.9$

$\gamma = 0.15$

$r = 0.3$

Rendszerjellemzők:

Szerver kihasználtság: 70%

Kiszolgálandó igények átlagos száma: 1.38

Működőképes terminálok átlagos száma: 2.9

Látható, hogy az igények válaszideje csökkent, azaz ilyen szemszögből a rendszer

jobb eredményt mutat az előzőnél. A működőképes terminálok várható számának csökkenése miatt a rendszer úgy viselkedik, mintha kevesebb terminál lenne, ez magyarázható a jobb válaszidő.

c./ Szerver meghibásodás :

Terminálok száma: 4

Terminálparaméterek:

Átl. válaszidő

$$\lambda = 0.5$$

$$T = 3.9$$

$$\gamma = 0.15$$

Rendszerjellemzők:

Szerver kihasználtság: 66%

Kiszolgálandó igények átlagos száma: 2.3

Működőképes terminálok átlagos száma: 4

Látható, hogy a szerver meghibásodása valamennyi mutatót vizsgálva gyengébb eredményekhez vezet: a szerver kihasználtsága csökkent, az átlagos válaszidő értéke növekedett az a./ példa eredményeihez képest.

II. Szoftvermegbízhatóság

Bevezetés

Cikkünk második részében az eddigi alapvetően hardverekkel kapcsolatos vizsgálatokról áttérünk a szoftverekre. A megbízhatóságelméletnek egy új, dinamikusan fejlődő ága foglalkozik ezzel a területtel, melynek problémakörét, fejlődését és fontosabb irányait szeretnénk röviden bemutatni.

Az információs technológia egyre jelentősebb szerepet kap napjainkban az élet szinte minden területén, s ennek megfelelően kiemelten kezelik az ún. 'software engineering' feladatokat. Fontosságát az információs rendszerek kifejlesztésének, valamint ezek meghibásodásának növekvő költségei támasztják alá, ezen kívül lépést kell tartani a számítástechnika fejlődésének egyre gyorsuló ütemével, és az információs rendszerek növekvő komplexitásigényével.

1./ Alapfogalmak

A hardver elemek megbízhatóságával már jóval korábban foglalkoztak, számos eredmény született különböző rendszerek leírására és modellezésére. Ezek az eredmények azonban a két terület között mutatkozó lényeges különbségek miatt nem vihetők át közvetlenül a szoftverekre.

Az említett eltérések közül emeljünk ki néhányat:

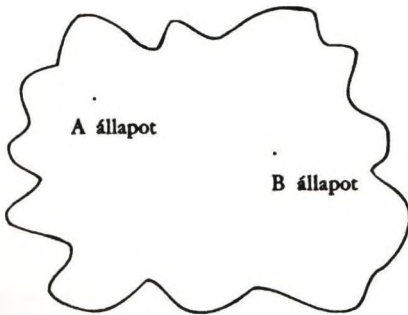
- A szoftverhibák forrása általában tervezési elégtelenség, programozói tévedés, míg a hardver esetében az esetek többségében a fizikai előregedés, elhasználódás.
- A hardverekkel ellentétben a szoftverhibák forrása általában nehezen lokalizálható, nem mindig rendelhető hozzá egyértelműen valamely programkódhoz, vagy szegmenshez.
- A szoftver megbízhatóságának jellemzői a tesztelés és hibafeltárás közben is módosulhatnak, mivel javítás közben újabb kódok, programrészek kerülnek be, ill. esnek ki a programból.

A szoftver termékek legfontosabb jellemzői a minőség és az ár. A minőségi mutatók közül talán a megbízhatóság a legjelentősebb. A megbízhatóságot intuitíven úgy lehet megfogalmazni, hogy a jó szoftver a felhasználó igényeinek megfelelően működik. Ezek szerint a szoftverben akkor van 'hiba', ha működése

valamilyen módon eltér attól amit a felhasználó vár (futás közbeni leállítás, rossz eredmények, hibás input/output műveletek, stb.).

Ahhoz, hogy pontosabban értelmezhesük a megbízhatóság fogalmát, néhány alapfogalmat kell tisztáznunk.

Különbséget kell tenni a *hibaforrás* és a *futási hiba* (továbbiakban hiba) között. Már említettük, hogy a hibaforrást a programozó tévedése eredményezi. Ha a végrehajtás során a vezérlés ilyen hibás kódot tartalmazó sorra vagy programrészletre kerül, akkor az futási hibát eredményezhet (egy hibás kód önmagában nem feltétlenül okoz futási hibát, ehhez esetleg egyéb tényezők pl. bizonyos típusú input adatok jelenléte is szükséges lehet). Ezt könnyen szemléletesé lehet tenni azáltal, hogy egy kétdimenziós tartományként ábrázoljuk az input állapotteret (1. ábra), melynek pontjait (az input állapotokat) hozzárendeljük a program futás közbeni lehetséges állapotaihoz [7]. Ebben az állapotterben lehet olyan tartomány, mely a tesztelés után jóformán hibamentesnek tekinthető, ill. tartalmazhat olyan részt is, amelyben a hibaforrások az eddig felderítettektől lényegesen különböznek, mivel olyan körülmények között aktivizálódhatnak, melyek a tesztelés során nem fordultak elő.



1. ábra

Input állapotter

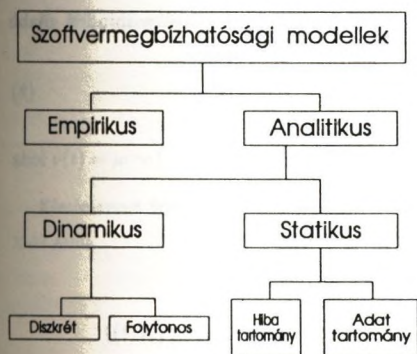
A programhibák időbeni előfordulásait a következő jellemzőkkel lehet megadni

1. A hibák előfordulásának időpontja (t_i),
2. A hibák előfordulásai között eltelt idő ($t_i - t_{i-1}$),
3. Adott időpontig észlelt hibák száma,
4. Adott időintervallumban észlelt hibák száma,
5. A következő hiba előfordulásáig eltelő várható idő,
6. A szoftverben maradt hibák várható száma.

2./ Szoftvermegbízhatósági modellek csoportosítása

A szoftvermegbízhatóság elméletében kidolgozott modellek egy csoportosítását láthatjuk a 2. ábrán [8]. Ez a táblázat a hibák előfordulásának eloszlására tett újabb feltevésekkel tovább finomítható. - *Empirikus modellek* Ezeknél a modelleknél a megbízhatósági mutatók és a megfelelően definiált paraméterek közötti kapcsolatot a megfigyelt adatok empirikus elemzése által nyerjük (pl. Moranda (1976) [9]).

- *Analitikus modellek* Alkalmazásuk általában négy lépésből tevődik össze. Az első lépésben meghatározzuk az alapfeltevéseinket a tesztelési eljárásnak megfelelően, majd a másodikban ezen alapokat használva felépítjük az analitikus modellt. A harmadik lépésben az adatok alapján meghatározzuk a rendszer paramétereit, végül a modell segítségével a szoftver további működésére nyerhetünk előrejelzéseket. Az analitikus modelleken belül megkülönböztetünk dinamikus (a szoftver jellemzői az idő során változhatnak) és statikus modelleket.



2. ábra

A megbízhatósági modellek csoportosítása

- *Dinamikus modellek* Két fő csoportjuk van, a folytonos és diszkrét idejű modellek. A folytonos idejűekkel kezdtek el legkorábban foglalkozni, a legtöbb eredmény itt született (pl. Jelinski-Moranda (1972) [10]). A diszkrét idejű modelleket még viszonylag keveset tanulmányozták. Itt az alapfeltevés az, hogy az adatokat rögzített vagy véletlen időintervallumokból nyerik.

- *statikus modellek* Ezeket is diszkrét idejű modelleknek lehet elképzelni, s alapvetően kétfajta megközelítési módot szokás használni. Az egyikben a hibák alkotta állapotteret tekintik, s erre építik fel a modelleket, míg a másikban a hibák helyett az input adatok összességét veszik alapul (hiba- ill. adat-tartomány modellek).

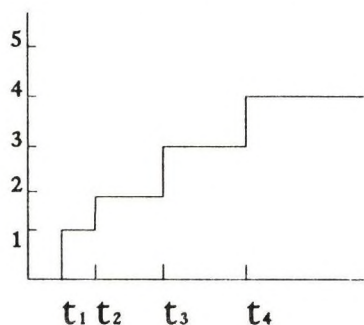
Ezekon kívül van még a modelleknek egy olyan csoportja is, amely azt a kérdést hivatott eldönteni, hogy egy szoftvert mennyi ideig érdemes tesztelni, mikor érkezik el a piacra dobás optimális ideje (pl. Okumoto-Goel (1979) [11]).

3./ Markov-modellek

Ebben a részben röviden ismertetjük, hogy a legelterjedtebb Markov-modellek alapfeltevéseire támaszkodva hogyan építhető fel egy konkrét modell.

A Markov-folyamatok jól használhatóak a szoftverek időbeli viselkedésének jellemzésére, különös tekintettel egy adott t időpontig megfigyelt hibák és a szoftverben maradt hibák számának becslésére. Tekintsünk a következőkben egy speciális Markov-folyamatot, nevezetesen a $\lambda(t)$ paraméterű inhomogén Poisson-folyamatot.

Jelöljük $X(t)$ -vel a t időpontig megfigyelt hibák számát. A 3. ábrán látható az $X(t)$ folyamatnak egy lehetséges realizációja. ($X(t)$ értéke minden újabb hibaelőfordulásnál eggyel nő.)



3. ábra

A hibák száma ($X(t)$)
az idő függvényében

Alapfeltevéseink:

1. $X(0) = 0$, azaz a $t = 0$ pillanatban nem észlelünk hibát.
2. A folyamat független növekményű.
3. Annak valószínűsége, hogy a $(t, t + \Delta t)$ intervallumban hibát észlelünk $\lambda(t)\Delta t + o(\Delta t)$, ahol $\lambda(t)$ a folyamat paramétere; $\lim_{x \rightarrow 0} \frac{o(x)}{x} = 0$.
4. Annak valószínűsége, hogy $(t, t + \Delta t)$ -ben egynél több hiba fordul elő $o(\Delta t)$.

Ezután a 2, 3 és 4 feltevések alapján felírhatjuk a folyamat átmeneti függvényét, s a Markov-tulajdonság felhasználásával megmutatható, hogy

$$(1) \quad P(X(t) = x) = \frac{(\mu(t))^x}{x!} \exp(-\mu(t)),$$

ahol $\mu(t) = \int_0^t \lambda(x) dx$. Ha a t_0 időpontig x_0 hibát észleltünk, akkor a (t_0, t) intervallumban lévő hibák várható számának eloszlására a független növekményűség figyelembe vételével azt kapjuk, hogy

$$(2) \quad \frac{(\mu(t) - \mu(t_0))^{x-x_0}}{(x-x_0)!} \exp\{-(\mu(t) - \mu(t_0))\}.$$

Ha $Y(t)$ -vel jelöljük a t időpontban a programban maradt hibák számát, akkor

$$(3) \quad Y(t) = X(\infty) - X(t).$$

Az $Y(t)$ eloszlására a (2) felhasználásával a következő adódik:

$$(4) \quad P(Y(t) = y) = \frac{\nu(t)^y}{y!} \exp\{-\nu(t)\},$$

ahol $\nu(t) = \mu(\infty) - \mu(t)$ a t időpontban még létező hibák száma.

Köszönetemet fejezem ki az MHB Kurátor Alap- és Alapítványkezelő Kft. támogatásáért.

Kuki Attila

IRODALOMJEGYZÉK

- [1]. L. Kleinrock, "Queueing Systems, Vol. I-II., Computer Applications," Wiley-Interscience, New York, 1976.
- [2]. R. Goodman, "Introduction to Stochastic Models," Benjamin/Cummings, California, 1988.
- [3]. L. Csige and J. Tomkó, *The machine interference for exponentially distributed operating and repair times*, *Alk. Mat. Lapok* 8 (1982), 107-124 (in Hungarian).
- [4]. A. S. Tanenbaum, "Computer Networks," Prentice-Hall International (UK) Ltd, 1989.
- [5]. J. Sztrik and T. Gál, *A recursive solution of a queueing model for a multi-terminal system subject to breakdowns*, *Performance Evaluation* 11 (1990), 1-7.
- [6]. B. Almási and J. Sztrik, *A queueing model for a non-homogeneous terminal system subject to breakdowns*, Technical Report 92/37, Department of Mathematics Lajos Kossuth University, Debrecen Hungary (1992).

- [7]. Musa, J.D., Iannino, A., Okumoto, K., "Software Reliability," McGraw-Hill Book Company, 1987.
- [8]. Shanthikumar, J.G., *Software Reliability Models: A Review*, Microelectron. Reliab. 23 (1983), 903-943; Great Britain.
- [9]. Moranda, P.B., *Limits to Program Testing with Random-Number Inputs*, Proc. COMPSAC (1978), 521-526].
- [10]. Jelinski, Z., Moranda, P.B., *Software Reliability Research*, Stat. Comp. Performance Evaluation (1972), 465-484, W. Freiberger.
- [11]. Okumoto, K., Goel, A.L., *Optimum Release Time for Software Systems*, Proc. COMPSAC (1979), 500-503.



Almási Béla matematikus,
a debreceni Kossuth Lajos Tudományegyetem
Matematikai és Informatikai Intézetének do-
gozója. Kutatásai a sorbanállási eredmények
gyakorlati, számítástechnikai alkalmazásá-
v kapcsolatosak. A sorbanállási rendszereken
belül a számítógéphálózatok meghibásodásán-
alízisével és sztochasztikus modellezésé-
vel foglalkozik.



Kuki Attila matematikus,
a debreceni Kossuth Lajos Tudományegyetem
Matematikai és Informatikai Intézetének do-
gozója. Kutatásai a szoftvermegbízhatósági
elmélettel kapcsolatosak, speciálisan a me-
bízhatósági modellek statisztikai vizsgálá-
tával.

ALL-EX/3 szakértői rendszerkeret

(Kőrösi Gábor, ALL Számítástudományi Kutató-Fejlesztő Kiszövetkezet, Budapest)

1. Bevezetés

Az ALL-EX/3 rendszer az ALL-EX szakértői rendszerkeretek sorában a harmadik generációt jelenti, így többéves fejlesztés és tapasztalatgyűjtés eredménye.

A korábbi (ALL-EX, ALL-EX PLUS) rendszerek egyik alkalmazási területe a jogszabályok tervezésének és szakértői rendszerbe történő leképezésének elősegítése volt. Ez nagymértékben meghatározta az ALL-EX/3-mal szemben támasztott igényeket, főképpen a tudásreprezentáció és magyarázatadás rugalmasságát illetően.

Az alábbi célok fogalmazódtak meg:

- aránylag egyszerű, de rugalmas tudásleíró nyelv frame-ill. szabályalapú tudásbázisok kifejlesztésére
- hatékony következtetési stratégiák (előre- ill. hátrafelé következtetéssel)
- rugalmas és kényelmes tudásbázisfejlesztő eszközök (szerkesztés, tesztelés)
- felhasználóbarát fejlesztő és futtató környezet (korszerű ablaktechnika, egér)
- opcionálisan külső adatbázisok elérése, külső nyelvi rutinok futtatása
- opcionális indukciós előtét, a tudás tapasztalati példákkal történő megadására

A tudásleíró nyelv, valamint a fejlesztő környezet úgy lett kialakítva, hogy a rendszert a kezdő felhasználótól kezdve a képzett tudásmérnökig, különböző képzettségi fokú alkalmazó egyaránt használhassa, s igen rövid idő alatt is egy futtatható prototípust építhessen fel.

A tudásleíró nyelv és a fejlesztő környezet angol kulcsszavakat használ, de a szakértői rendszer használatakor megjelenő összes szöveges információ programozható, akár ékezetes magyar betűket is tartalmazhat.

2. Tudásreprezentáció

2.1 Fogalmak

Az összefüggésekkel leírt szaktudás által használt fogalmakat frame struktúrába definiálhatjuk, amely a fogalmak rendszerezését, valamint további segédinformációval való ellátását teszi lehetővé. Ha egy fogalmat tulajdonságokkal (attributumokkal) ruházunk fel, ezt összetett fogalomnak nevezzük. Az egyszerű - tulajdonságokkal nem rendelkező fogalmakat globális változóknak tekinthetjük. Az összetett fogalmak tulajdonságaihoz, ill. az egyszerű fogalmakhoz szöveges vagy numerikus értékek rendelhetők, pl.

biztosított.neve = Nagy_János
végösszeg = 1500

ahol a '.' jel a fogalom - tulajdonság birtokviszonyra utal. Az attributumokhoz rendelhető értékek nem feltétlenül végleges, bármely érték törölhető, ill. felülírható.

Minden attributumra speciális vezérlő információk, ún. metaslotok vonatkoznak, amelyek definiálása az attributummal együtt történik. Ezek a vezérlő elemek határozzák meg az attributum kiértékelésének körülményeit.

A fogalmakat hierarchikus struktúrába szervezhetjük, lehetővé téve ezáltal a tartalom információ csökkentését az öröklés megengedésével. Ha több fogalom ugyanazokkal a tulajdonságokkal rendelkezik, akkor létrehozhatunk egy közös őst frame-et, és a tulajdonságokat elegendő ennél a frame-nél definiálni, a tulajdonságok értékei automatikusan öröklődik a leszármazottakra. A tulajdonságok létezésének öröklés erősebb feltételekhez van kötve a tulajdonságok értékének örökölhetősége. A hierarchia tetszőleges mélységű lehet. Egy frame-hez több közvetlen őst (ill. közvetlen leszármazott) is definiálhatunk, ezek megadási sorrendje egyben fontossági sorrendet is jelent.

2.2 Összefüggések

A fogalmak közti összefüggéseket, logikai kapcsolatokat - vagyis a tényleges szaktudás szabályok írják le. Kétféle HA - AKKOR szerkezetű szabályt definiálhatunk: hátrafelé előrefelé következtetőt.

Egy hátrafelé következtető szabály az alábbiakat fejezi ki:

HA bizonyos feltételek teljesülnek

AKKOR egy fogalom valamely tulajdonságához hozzárendelünk egy értéket,
ezt opcionálisan kiegészíthetjük speciális mellékhatásokkal

Egy előre felé következtető szabály az alábbiakat fejezi ki:

HA bizonyos feltételek teljesülnek

AKKOR ennek speciális mellékhatásai vannak

A szabályok feltétele mindkét esetben ÉS kapcsolatban álló elemi feltételekből áll, a mellékhatások pedig ÉS kapcsolatban álló akciók.

Egy elemi feltétel egy vagy több attributumra vonatkozó logikai kapcsolatot (értékvizsgálatot, állapotvizsgálatot, aritmetikai relációt), vagy két fogalom hierarchikus viszonyát fejezheti ki.

Az akciók segítségével közvetlenül módosíthatók v. törölhetők az attributum értékek, szöveges információt írhatunk ki a képernyőre, printerre, külső file-ba, megszakíthatjuk v. felfüggeszthetjük a szakértői rendszer futtatását, stb. .

A szabályokban többszörös birtokviszony, valamint indirekt értékkihívkozás is kifejezhető, amint az alábbi hátrafelé ill. előre felé következtető szabály is mutatja:

```
if jogesemény.tárgya = keresőképtelenség
and jogalany = ügyvéd
and táppénz.megállapítás.ellenőrzése = rendben
then táppénz.összeg = táppénz.fizetett_napok_száma * táppénz.napi_összeg
and display("A táppénz összege:#táppénz.összeg# Ft",eredmény)

if jogesemény.tárgya = keresőképtelenség
and jogalany = ügyvéd
and táppénz.megállapítás.ellenőrzése = rendben
then set(táppénz.összeg,táppénz.fizetett_napok_száma * táppénz.napi_összeg)
and display("A táppénz összege:#táppénz.összeg# Ft",eredmény)
```

Speciális összefüggéseket írnak le az attribútumok értékváltozásait figyelő démonok amelyek adott típusú változás esetén - esetleg további feltételektől függően - egy akciósorozatot hajtanak végre.

2.3 Környezeti deklarációk

A tudásbázisban - a fogalmak és összefüggések megadása mellett - definiálhatjuk szakértői rendszer felhasználói felületét is.

Egyrészt definiálnunk kell mindazon ablakokat, amelyekre akciókkal hivatkozunk, másrészt lehetőség van a futtató rendszer által nyújtott alapértelmezett konzultációs háttér ablak ill. rendszerüzenetei módosítására.

2.4 Stratégiák

A szakértői rendszer futtatása, azaz a konzultáció egy problémamegoldást jelent, amely tudásbázisra, valamint külső forrásokra (felhasználói válasz, külső adatbázis) támaszkodhat. A végrehajtás módját globális stratégia deklarációkkal írjuk elő.

A következtető mechanizmus a tudásbázisban deklarált öröklési és következtetési stratégia alapján működik.

A választható öröklési stratégiák:

- először széltében kereső (breadth-first)
- először mélységben kereső (depth-first)

Öröklési konfliktushelyzetben (azaz több lehetséges és létezése esetén) az öröklési stratégia szerint választunk a szóba jöhető ősök közül.

A választható következtetési stratégiák:

- hátrafelé következtető (backward reasoning)
- előre felé következtető (forward reasoning)

A következtetési stratégia egyben meghatározza a definiálható szabályok típusát is.

Hátrafelé következtetés esetén egy célsorozatot kell megoldanunk. Egy cél megoldása vagy egy szabály konklúziója, vagy egy felhasználói válasz alapján történhet. Az első esetben a szabály feltételének teljesülése újabb részcélok megoldását teheti szükségessé, s így egy célból kiindulva az egymáshoz láncoló szabályokon keresztül deduktív módon jutunk el a már igazzá vált elemi állításokig, vagy kérdést teszünk fel a felhasználónak.

Előrefelé következtetés esetén kezdeti adatokból kiindulva azon szabályokat tüzeljük (vagyis azon szabályok akcióit hajtjuk végre), amelyek feltétele az ismert adatok alapján teljesül. Ezen szabályok akciói módosíthatják az ismert adatok körét, s újabb szabályok válhatnak tüzelésre alkalmassá, amelyeket a tüzelhető szabályok listájának végére helyezünk. Ez a folyamat addig tart, amíg vagy nincs több tüzelhető szabály, vagy egy adott feltételhez kötött leállító akció megszakítja a futást. Előrefelé következtetés esetén is feltehetünk kérdéseket a felhasználónak.

Egy tudásbázis elemei (pl. stratégiái) a konzultáció alatt közvetlenül nem módosíthatók. Lehetőség van azonban több komponensből álló tudásbázis felépítésére, amelyben az egyes komponensek önmagukban is futtatható tudásbázist képeznek, de futtatáskor mindig csak egy komponens kerül be a memóriába. Egy speciális akcióval újabb komponenst tölthetünk be, amelynek lefutásakor a konzultáció az előző komponenssel, a felfüggesztési ponttól folytatódik. A komponenshívásokat tetszőlegesen egymásba ágyazhatjuk, így struktúrált tudásbázisokat hozhatunk létre. Ezen lehetőség egyik alkalmazásaként kevert következtetési stratégiájú tudásbázist is felépíthetünk.

2.5 Magyarázatok

A szakértői rendszereknek - a hagyományos programoktól eltérően - egyik fontos tulajdonsága, hogy magyarázataadásra képesek. Emellett persze az is fontos, hogy a magyarázatok ne csupán a tudásbázis fejlesztőjének adjanak elegendő információt, hanem a szakértői rendszer felhasználóinak is.

Az ALL-EX/3 által szolgáltatott magyarázatok mozaikokból épülnek fel, melynek elemei attributumok, szabályok, démonok, valamint rövid összekötő szövegek. A felsorolt tudásbázis elemek alapértelmezésben eredeti alakjukban jelennek meg a magyarázatokban, de bármelyiket szöveges leírással helyettesíthetjük, így a magyarázat nem tartalmaz kulcsszavakat vagy egyéb szintaktikai jeleket.

2.6 A tudásreprezentáció általánosítása

Az ALL-EX/3 tudásbázisok reprezentációja egy aránylag egyszerű, de magasszintű - emberi gondolkodáshoz közelebb álló - programozási nyelv segítségével történik, aminek lehetővé teszi, hogy a tudásbázis felépítésekor ne a technikai részletekre, hanem inkább magára a tudásra koncentrálhassunk. Bár a nyelv maga is szolgáltat bonyolultabb funkciókat és lehetőségeket, amelyek kihasználásához mélyebb tudásmérnöki ismeretekre és nagy gyakorlatra van szükség, ezen túlmenően a tudásbázishoz opcionálisan külső adatbázisok (pl. dBase, Oracle), valamint külső nyelvi programokat (pl. C, Prolog) illeszthetünk. A reprezentáció ezáltal alacsonyabb szintű lesz, de lehetőségeink nagy mértékben bővíthetők.

2.7 A tudásreprezentáció egyszerűsítése

Az ALL-EX/3 az opcionális indukciós előtét segítségével lehetőséget nyújt arra, hogy a tudás megadása ne a standard szintaxis szerint történjen, hanem ismereteinket tapasztalati példákkal adjuk meg, majd ezekből egy ALL-EX/3 szintaxisú tudásbázist generáljunk.

Az indukciós előtét a példák közül egyrészt szabályokat generál, a példákban szereplő változók közreműködésével. A szabályokat két különböző módszer szerint lehet létrehozni: vagy az ún. optimális szabálystruktúrát az információelméleti számításokon alapuló algoritmusokkal, vagy pedig a megadott változókat tetszőlegesen felállított sorrendjük szerint beépítve a szabályokba. A szabályok mind ugyanarra a (felhasználó által kijelölt) célváltozóra következhetnek, ez alól csak a több szintű szabályokat is megengedő opció beállítása esetén van kivétel. Az indukciós előtét a szabályok mellett automatikusan létrehozza a használt fogalmak reprezentációját is, így egy egyszerű tudásbázist kapunk, amelyet szükség esetén a standard fejlesztési eszközökkel bővíthetünk.

3. A szakértői rendszer felépítése

A szakértői rendszer felépítése az ALL-EX/3 fejlesztő moduljával végezhető el, aminek opcionálisan kiegészülhet az adatbázis ill. nyelvi interfésszel, valamint az indukciós előtéttel. Az alábbiakban a fejlesztő modul fő funkcióit tekintjük át:

- szerkesztés
- a struktúra kirajzolása
- tesztelés

3.1 A tudásbázis szerkesztése

A tudásbázis új elemeinek bevitele, ill. a meglévők módosítása egy speciális editorral történik, amely egyrészt rendelkezik a hagyományos editorfunkciókkal (beszúrás, módosítás, törlés, keresés, kicserélés, stb.), másrészt célirányosan támogatja a különböző tudásbázis elemek szerkesztését. A fejlesztő a tudás lényegi részére koncentrálhat, az elvárt szintaktikai elemeket tartalmazó űrlapokat kell csak kitöltenie. A szerkesztő szintaktikai és konzisztencia ellenőrzést is végez.

A tudásbázis forrásalakja tetszőleges külső szövegszerkesztővel is elkészíthető, az ALL-EX/3 szintaxisának megfelelően. A szöveges alak betöltése viszont hosszabb időt vesz igénybe a szintaktikai ellenőrzés miatt.

3.2 A tudásbázis struktúrájának kirajzolása

A tudásbázis felépítését nagy mértékben megkönnyítheti, ha a fejlesztő vizuálisan is látja a logikailag egymáshoz kapcsolódó tudásbázis elemeket.

Egyrészt kirajzolhatjuk a fogalmak hierarchikus rendszerét, másrészt a hátrafelé, ill. előrefelé következtetésben egymáshoz kapcsolódó szabályokat és démonokat.

3.3 A tudásbázis tesztelése

A fejlesztő modul lehetővé teszi, hogy a tudásbázis felépítésével párhuzamosan annak helyességét is ellenőrizzük, azaz tesztfuttatásokat végezzünk. Így már a fejlesztés kezdetén is működő prototípust tudunk előállítani, amelyet újabb és újabb elemekkel bővíthetünk.

A tesztelés ugyanazt a konzultációt szimulálja, amelyet az elkészült szakértői rendszerrel fogunk folytatni. Könnyebben tesztelhetjük a tudásbázist és ellenőrizhetjük a következtetés helyességét, ha a konzultációkat interaktív nyomkövetéssel hajtjuk végre. A fejlesztő a nyomkövetés mélységének megválasztásával, valamint nyomkövetési töréspontok beállításával tetszőleges módon láthatóvá teheti az egyes funkciók végrehajtását.

4. A szakértői rendszer használata

A tudásbázis felépítését és kitesztelését követően elkészíthetjük a szakértői rendszert, amely egyrészt a futtató modulból, másrészt a lefordított tudásbázisból (egy vagy több lefordított komponensből) áll. A tudásbázis fordítása kiszűri a még meglévő inkonzisztens problémákat.

4.1 Konzultáció

Az elkészült szakértő rendszert egy adott szakterület problémáinak megoldására, vagy tanácsadásra használhatjuk. A szakértő rendszerrel történő párbeszéd - a konzultáció során kérdések jelenhetnek meg, amelyet a végfelhasználónak kell megválaszolnia vagy a menüből történő választással, vagy közvetlen begépeléssel. A végfelhasználó a tudásbázis módosítani nem tudja, a válaszait, valamint további kikövetkeztetett részeredményeket a tudásbázistól elkülönítve és csak ideiglenesen őrizzük meg. Egy új konzultáció indításakor korábbi dinamikus adatok elvesznek, de lehetőség van egy konzultáció megismétlésére módosított felhasználói válaszokkal is.

4.2 Magyarázatadás

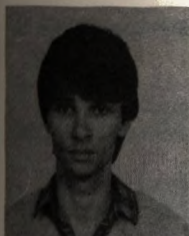
Az ALL-EX/3 kérdésfeltevésakor, ill. egy konzultáció végén tud magyarázatot adni.

Egy kérdés megválaszolása előtt a Miért? parancs kiadásával a rendszer megmagyarázza a kérdés feltevésének okát. Ez azt jelenti, hogy megjelenik az éppen vizsgált szabvány, amelynek a vizsgálata előidézte a kérdés feltevését a tudás hiányos volta miatt. Az aktuális elemről kezdve az egész következtetési lánc megvizsgálható, egészen a kiindulási pontig.

Egy konzultáció után a Hogyan? parancs kiadásával a rendszer megmagyarázza, hogy határozta meg egy adott attribútum értékét, azaz rekonstruálja az eredeti konzultációt. A következtetés minden pontját bejárhatjuk lépésről lépésre.

5. Implementáció

Az ALL-EX/3 rendszer implementációs nyelve a hazai fejlesztésű CS-PROLOG (MULTILIGOC Kft., ALL Kiszövetkezet), valamint a C nyelv. A rendszer DOS, OS/2, SCO UNIX, valamint T414/T800 transzputeres változatának fejlesztése van folyamatban.



Kőrösi Gábor matematikus,
az SZKI-ban, majd a Multilogic Kft-ben dol-
gozott tudományos munkatársként, ahol a mes-
terséges intelligencia szakterületén belül
szimulációval, kooperatív rendszerekkel, va-
lamint szakértői rendszerekkel foglalkozott.
Jelenleg az ALL Számítástudományi Kutató-
Fejlesztő Kiszövetkezetben a szakértői
rendszerek témáját vezeti.

Áttekintés és kitekintés két kutatási területről:

- | | |
|----------------------------------|-----------------------|
| 1. Hatékonyságmodellezés: | Arató Mátyás |
| 2. Adatbáziskezelés: | Benczúr András |

A két előadás a 70-es évek elején a SZTAKI-ban indult közös kutatások szétágazását végigkövetve összefoglalja a hazai kutatásokat, eredményeket.

Kitekintést ad a nemzetközi kutatások aktuális kérdéseire, kihatásaikra az információtechnológiára, s célkitűzéseket, perspektívákat vázol. A két terület jelenlegi legfontosabb közös kérdései a nagy alkalmazási rendszerek kialakítása köré helyezhetők. Hogyan fogható meg ezeknek a hatékonysága, s az adatbáziskezelés vajon nem egy "alkalmazásbáziskezelés" irányába halad-e?

1. HATÉKONYSÁGMODELLEZÉS

A 70es években üzemeltetési feladatok motiválták a hatékonyságmodellezést, ennek hatására kialakult kutatási irányok:

- Operációs rendszerek modellezése, (ütemezés, kihasználtság, sztochasztikus és tömegkiszolgálási modellek, diffúziós közelítések).
- Lapozási módszerek, háttértároló átrendezés.

Később inkább irodalmi motívációjú kutatások:

- Tömegkiszolgálási hálózatok kutatása.
- Számítógépes hálózatok, osztott rendszerek sztochasztikus modellezése.

Mai aktuális irányok:

Nagy osztott rendszerek működésére ma sincsenek egzakt modellek.

A kommunikációs, számítógépes hálózatok hatékonyságvizsgálata újabb feladatokat ad.

Előtérbe kerül az alkalmazási rendszerek hatékonyságának kérdése, de nem a gépek kihasználtsága, hanem a felhasználók szempontjából.

2. Adatbáziskezelés

Nagy adatfeldolgozási szerződéses munkák váltották ki a szervezett ABKR kutatásokat a 70-es években, sorrendben:

- statisztikus működéselemzés, hatékonyság és biztonság (erősen kapcsolódott a hatékonyságmodellezéshez),
- diszkrét matematikai módszerek alkalmazása, (relációs adatmodellezés, függőgi rendszerek logikai, kombinatorikus vizsgálata),
- információs rendszerek tervezési módszerei, adatmodellezés, nyelvek (ISDOS PSL/PSA és SDLA)

A "nagy ütközet"-ben a hálós-hierarchikus-relációs adatmodellek között nem vettünk részt.

A 80-as évek fellendülése:

- újabb vizsgálatok az akkorra külföldön már elterjedt relációs ABKR-ek körül (irodalmi motíváció),
- új hatékonysági modell, a Kolmogorov entrópia alkalmazása,
- fogalmi modellezés,
- adatbáziskezelés PC-LAN-okon.

Mai aktuális irányok:

- objektumorientált és deduktív adatbázisok,
- tudásbáziskezelés, multimedia adatbázisok, intelligens adatbáziskezelés,
- 4 GL adatbáziskezelő rendszerek

Megint a piac dönti-e el az új akadémikus ütközetet az objektumorientált és deduktív adatbázisok között?

Jóslat: az alkalmazási rendszerek kezelése válik a legfontosabb felületté, ezt jelzi a most kitalált "alkalmazásbázis" fogalom.

TRANZPUTER ALAPÚ PÁRHUZAMOS SZÁMÍTÁSI TECHNOLÓGIA ÉS HELYZETE MAGYARORSZÁGON

Kacsuk Péter
KFKI Mérés és Számítástechnikai Kutatóintézet
Párhuzamos Számítási és Módszertani Központ
1525 Bp. Pf. 49.

Ferenczi Szabolcs
ALL Számítástechnikai Kutató Fejlesztő Kiszövetkezet
1539 Budapest Pf. 675.

1. BEVEZETÉS

A párhuzamos számítógépek fejlődésének egyik domináns irányzata az elosztott memóriájú multiprocesszorok osztálya, vagy más néven kommunikáló processzorok osztálya. Olyan népszerű és széles körben elterjedt párhuzamos számítógépek tartoznak ide, mint az Intel hypercube topológiájú iPSC-2 (Intel Personal SuperComputer) számítógépe, vagy a tranzputerekből felépülő különböző kiépítettségű (4-1000 processzort tartalmazó) számítógépek. Európában elsősorban a tranzputer alapú, míg Amerikában az iPSC számítógépek terjedtek el. A tranzputer által megtestesített párhuzamos gép szervezési elv azonban Amerikában is terjed, a Texas TMS320C40 DSP processzorban is ezt az elvet alkalmazzák.

A tranzputerek megjelenése forradalmasította a párhuzamos számítógépek alkalmazását több okból:

1/ A tranzputerek közvetlenül köthetők egymáshoz, minden további hardver alkalmazása nélkül.

2/ A hagyományos mikroprocesszorokból felépített multiprocesszorok egy közösen használt sín köré voltak szervezve, ami szűk keresztmetszetet képezett a multiprocesszoros rendszerben és az alkalmazható processzorok számát jelentősen korlátozta (10-20 processzorra). A tranzputerek közvetlen pont-pont összekötési módja lehetővé teszi lényegesen nagyobb számú processzor hatékony összekötését (10-10K), ráadásul könnyen változtatható topológia szerint.

3/ A pont-pont összekötés további következményeként a tranzputeres rendszerek könnyen bővíthetők. A fejlesztés kiindulhat egészen kis kiépítettségű rendszerből (2-4 processzor) és később a rendszer mérete az igények szerint növelhető.

4/ Magasszintű párhuzamos programozási nyelvek (Occam-2, Parallel C, Parallel Fortran, stb.) támogatják a hardver párhuzamos lehetőségeinek kiaknázását.

Mindezek eredményeképp **a tranzputer olyan áttörést hozott a párhuzamos rendszerek területén, mint a hagyományos mikroprocesszorok a szekvenciális számítógépekén**, nevezetesen az olcsó szekvenciális személyi számítógépek mellett

megjelentek az olcsó párhuzamos személyi számítógépek is, nagy tömegek számára lehetővé téve a párhuzamos programozás gyakorlati elsajátítását.

Ennek a folyamatnak a gyors végbemenetelét Magyarországon is támogatni kell, mivel ez elengedhetetlen feltétele az Európához való felzárkózásnak az információtechnológia területén. Ennek érdekében hoztuk létre az NJSZT-én belül a Magyarországi Transzputer Felhasználók Klubját (Hungarian Transputer User Group (HTUG)), amely a WoTUG (World Transputer User Group) magyarországi szekciójaként működik.

Az előadásban 2 fő témakörrel kívánunk foglalkozni. Az első részben a transzputerek kritikai elemzését adjuk kimutatva az 1. generációs transzputerek előnye mellett legfontosabb korlátaikat és megmutatjuk, hogyan tervezik ezek kiküszöbölését a 2. generációs transzputerekben. A második részben ismertetjük a HTUG céljait és tervezett programjait, továbbá az eddigi tevékenység eredményeképpen feltérképezett hazai transzputer alapú fejlesztések helyzetét.

2. A TRANSZPUTER KONCEPCIÓ

Az egész transzputer technológia azt a felismerést tükrözi, hogy nagy feladatok hatékony és gyors megoldásához a feladatot olyan elemi részfeladatokra kell bontani, amelyek

1/ önmagukban már nagyon egyszerűek és ezért nagy valószínűséggel hibamentesek

2/ egymással párhuzamosan, egyidejűleg képesek működni így csökkentve a program végrehajtási idejét.

A teljes feladat megoldása ezen egyszerű, feltételezhetően hibamentes részek együttműködésének, kommunikációjának eredményeként valósul meg. Ezt az elvet először Hoare fogalmazta meg CSP (Communicating Sequential Processes) koncepciójában [Hoar78], majd fejlesztette tovább [Hoar85]. A CSP koncepció egyidejűleg egy nyelvi javaslatot is tartalmazott az ilyen típusú megközelítésben készíthető programok írására. Ezt a nyelvi javaslatot fejlesztették tovább egy gyakorlatban is alkalmazható nyelvvé az INMOS cég munkatársai [inmos88]. Ezt az új nyelvet Occam-nek nevezték el és ez lett a transzputerek "anyanyelve".

Az Occam nyelv legfontosabb a CSP koncepciót tükröző tulajdonságai a következők:

1/ Minden utasítás (értékadó, input és output) önálló primitív részfeladatként (processzként) jelenik meg.

2/ Ezen primitív processzekből az ún. SEQ (szekvenciális) konstrukció segítségével lehet összetett, nagyobb bonyolultságú processzeket létrehozni. Egy-egy ilyen SEQ konstrukción belül a primitív részfeladatok végrehajtása garantáltan szekvenciálisan történik.

3/ Primitív és összetett processzek párhuzamos végrehajtását a PAR konstrukcióval lehet definiálni. Például:

SEQ

aa = 11

PAR

bb = 22

SEQ

cc = 33

dd = aa + cc

ee = 44

ff = bb + dd + ee

A külső SEQ konstrukció alapján először az aa értékadást kell végrehajtani, ezután a PAR konstrukciót, majd legvégül az ff értékadást. A PAR konstrukción belül viszont a bb és ee értékadást illetve a belső SEQ konstrukciót egyidejűleg is végrehajthatjuk. Figyeljük meg, hogy az ff értékadást mindaddig nem lehet megkezdeni, amíg a PAR konstrukcióban párhuzamosan elindított összes processz be nem fejeződött. Ez a párhuzamos processzek indításának és befejezésének strukturált, cobegin/coend-szerű megvalósítását jelenti.

4/ A párhuzamosan futó processzek un. csatornákon keresztül kommunikálhatnak egymással, melyeket a párhuzamos processzek létrehozása előtt szintén létre kell hozni és a processzeknek paraméterként átadni. Egy csatorna szigorúan csak két processzt köthet össze: az adót és a vevőt. A csatornán a kommunikáció szinkron módon történik: csatorna művelet esetén mind az adó, mind a vevő processznek meg kell várni amíg a csatorna művelet befejeződik, azaz a csatorna műveleten a két processz garantáltan összeszinkronizálódik.

5/ Gyakori eset, hogy egy vevő processz több különböző adó processztől vár üzenetet és az üzenetek vételi sorrendje nem kötött. Ennek az esetnek a lekezelésére szolgál a Dijkstra által javasolt őrzött utasítások [Dijk75] bevezetése az Occam-be. A transzputer nem más, mint az Occam nyelv hardverben történő megvalósítása. Egy transzputert úgy lehet elképzelni, mint egy hardver processzt. Az így kialakított hardver processzek hardver csatornán keresztül kommunikálnak egymással. A hardver csatorna az un. link szigorúan két transzputert köt össze, pont-pont összekötést biztosítva a transzputerek között. Az átvitel bitsoros a linkeken, így egyetlen vezeték elegendő két transzputer kommunikációjához. Valójában minden link két ellentétes irányú vezetékből áll, amelyek két ellentétes irányú szoftver csatorna leképezésére adnak lehetőséget. Minden transzputernek 4 linkje van, ami lehetővé teszi, hogy a transzputereket különböző topológiák szerint kössük egymáshoz. A legnépszerűbbek ezek közül:

- gyűrű
- fa
- négyzetrács (különböző él lezárásokkal)

Ezek a topológiák lehetővé teszik nagyszámú processzor rendszerbe integrálását anélkül, hogy a közös sínhez hasonló architekturális szűk keresztmetszet jelenne meg az így felépített rendszerekben.

Természetesen egy transzputerre több szoftver processzt is lehet képezni. Ilyenkor a transzputeren egy hardver kernel biztosítja a szoftver processzek megfelelő ütemezését. A transzputer belső szervezése következtében a processzcserérendkívül gyorsan, 1 microsec alatt megy végbe.

3. AZ 1. GENERÁCIÓS TRANZPUTER CSALÁD ÉS ALKALMAZÁSA

Az 1. generációs transzputer család legfontosabb elemei:

- a. T212 (T222) 16-bit processzorok
- b. T414 (T425) 32-bit processzorok
- c. T800 (T805) 32-bit processzorok 64-bit lebegőpontos aritmetikával
- d. M212 16-bit diszk vezérlő processzor

Fontosak még az alábbi funkcionális kiegészítő elemek:

a. C004 programozható 32 vonalas cross-bar kapcsoló, amely lehetővé teszi soktranszputeres rendszerek topológiájának programmal történő kialakítását és dinamikus módosítását.

b. C011 adatkonverziós chip, amely lehetővé teszi a transzputer soros linkjeinek és a perifériák párhuzamos adatútjainak illesztését.

c. C012 adatkonverziós chip, amely lehetővé teszi a transzputer soros linkjeinek és a mikroprocesszorok párhuzamos adatsínjeinek illesztését.

A fenti elemcsaládra alapozva számos transzputeres rendszert hoznak forgalomba.

Ezek 3 nagy csoportba sorolhatók:

- a. PC-be dugható kártyák
- b. munkaállomásba dugható kártyák
- c. önálló szuperszámítógépek

A PC-be és munkaállomásba dugható kártyák további 2 alosztályba sorolhatók [inmos89]:

- a. rögzített funkciójú kártyák
- b. modulárisan kiépíthető alapkártyák

A modulárisan kiépíthető alapkártyákhoz különböző transzputer modulok (TRAM-ok) vásárolhatók. Egy-egy alapkártyára tipikusan 10-32 TRAM egységet lehet elhelyezni. Egy TRAM tipikusan egy transzputerből és valamekkora memóriából áll. A különböző TRAM modulok 1-10 TRAM egységnek felelnek meg attól függően,

hogy mekkora memóriát tartalmaznak. Például az IMS B008 PC-be dugható alapkártya 10 TRAM helyet tartalmaz, amit például 10 db IMS B411 (32-bit transzputer, 1 Mbyte, 1 egység) TRAM-mal, vagy 5db IMS B404 (32-bit transzputer, 2 Mbyte, 2 egység) TRAM-mal lehet kiépíteni (vagy ezek valamilyen kombinációjával). Természetesen vannak speciális funkciójú TRAM-ok is, mint például az IMS B409 display TRAM (T222 transzputer, 8 egység), vagy az IMS B420 vektor processzor TRAM (T800 transzputer, 1 Mbyte, 4 egység). A modulárisan kiépíthető alapkártyák nagy előnye éppen abban rejlik, hogy nagy variációs lehetőséget kínálnak a felhasználóknak különböző típusú soktranszputeres rendszerek kiépítésére.

A transzputer alapú szuperszámítógépek közül a következők a legismertebbek:

- a. Meiko Computing Surface (128-1024 transzputer)
- b. Parsys Supernode (16-512 transzputer, 190 MFLOP)
- c. Parsytec Megaframe (64-256 transzputer, 386 MFLOP)

A transzputer rendszerek szoftver ellátottsága sajnos messze elmaradt a hardver fejlettségi szintjétől. Felhasználói környezetük nem elég barátságos, hibakeresést támogató eszközeik kezdetlegesek és nem elég hatékonyak. Mindent összevetve ezen a területen még komoly fejlődésre lehet számítani.

A transzputerek általános célú számítógépként való alkalmazásához alapvetően két UNIX-alapú operációs rendszer terjedt el:

- a. Helios (sokfelhasználós, soktaszkos, sokprocesszoros)
- b. Genesys (soktaszkos, sokprocesszoros)

Az Occam-2 mellett a C, PASCAL és FORTRAN nyelvek párhuzamos kiterjesztése áll rendelkezésre soktranszputeres környezetben. Jelentős kutatási/fejlesztési tevékenység irányul a Prolog soktranszputeres megvalósítására. Egy ilyen hazai eredmény a CS-Prolog soktranszputeres változata [KaFF90], amely megjelenésekor világszerte újdonságnak számított. Meg kell még említeni a STRAND'88 [FoTa90] nyelvet, amely Prolog szintakszisú, de valójában funkcionális programozási nyelv párhuzamos számítógépek számára és amely Helios alatt soktranszputeres gépeken is rendelkezésre áll.

4. AZ 1. GENERÁCIÓS TRANSZPUTER CSALÁD KRITIKÁJA

Az 1. generációs transzputer család jellemzője, hogy a processzorközi kommunikáció a nemszomszédos processzorok között lényegesen lassabb, mint a szomszédos processzorok között. Ennek oka, hogy a nemszomszédos processzorok között nincsen közvetlen fizikai összeköttetés és így az üzenetnek át kell haladnia mindazonokon a transzputeren, amelyek az adó és vevő transzputer közötti láncban helyezkednek el. Ha a lánc hossza N , akkor az üzenetátvitel ideje legalább N -szerese a szomszédos (1 hosszúságú láncon való) átvitelnek. Másrészt a láncban lévő transz-

putereken lelassul az odarendelt processzek végrehajtása az idegen transzputerek számára történő üzenetátvitel következtében. Mindezek alapján az 1. generációs transzputerek esetén nagy gondot kell arra fordítani, hogy a szoftver processzeket úgy rendeljük a hardver processzekhez, hogy az üzenetláncok hossza minimális legyen.

További leképezési problémát jelent, hogy egy transzputerből csak 4 szoftver csatorna indulhat ki és ugyanennyi futhat be (a 4 transzputer linken keresztül). A transzputerre leképezett szoftver processz azonban tetszőleges számú ki- és bemenő szoftver csatornát használhat. Ha tehát ezek száma meghaladja a fizikai csatornák számát, akkor a leképezés során kell gondoskodni arról, hogy a kommunikáló processzek közül csak annyi kerüljön más transzputerre, amennyit a hardver csatornák száma megenged.

A leképezési probléma miatt a transzputerek programjait alapvetően két osztályba kell sorolni:

- a. szomszédsági algoritmusok
- b. általános algoritmusok

A szomszédsági algoritmusok szabályos, homogén adatstruktúrák feldolgozására szolgálnak és így könnyen leképezhetők szabályos topológiájú, homogén processzor terekbe pusztán a szomszédsági kommunikációt kihasználva. Az általános algoritmusok megkívánják a processzek tetszőleges kommunikációját és így az optimálist közelítő leképezésük szabályos topológiájú, homogén processzor terekbe rendkívül nehéz, sok esetben lehetetlen.

A leképezési probléma nehézségeit egy ún. kommunikációs réteg kialakításával lehet feloldani. Ennek lényege, hogy minden transzputeren elhelyezünk egy olyan szoftver réteget, amely a transzputerre leképezett processzek elől eltakarja a transzputer hálózat tényleges topológiáját. Összegyűjti a transzputeren feladott üzeneteket. Minden egyes üzenet esetén a céltranszputernek megfelelő hardver csatornán továbbítja az üzenetet a következő transzputer számára. A bemenő csatornákon érkező üzeneteket szétválogatja oly módon, hogy a nem neki szóló üzeneteket továbbítja a transzputer lánc következő elemének, míg a neki szóló üzeneteket átadja a beazonosított célprocessznek. A kommunikációs réteg alkalmazása jelentősen egyszerűsíti a leképezési problémát, de jelentősen növeli a transzputer rendszer kommunikációval kapcsolatos adminisztrációs idejét csökkentve a soktranszputeres rendszerrel elérhető programvégrehajtási sebességnövekedést.

5. A 2. GENERÁCIÓS TRANSZPUTER CSALÁD

Az 1. generációs transzputerek alapvetően a szomszédsági algoritmusok megvalósítása esetén használhatók ki optimálisan. Meg kell jegyezni, hogy tervezőik

eredetileg nem is szándékozták általános algoritmusok megvalósítására tervezni a transzputert, de a felhasználók felismerve a transzputerekben rejlő lassú, de általános kommunikáció esélyét azonnal elkezdtek a transzputereket, mint általános célú párhuzamos számítógépeket alkalmazni. Ehhez azonban az 1. generációs transzputerek kommunikációs rendszere nem volt elég gyors és így nagy csalódást okozott a felhasználóknak.

Ennek megfelelően a 2. generációs transzputerek kidolgozásánál az egyik legfontosabb célkitűzés az volt, hogy kommunikációs rendszerüknek ki kell elégíteni az általános célú számítógépekkel szemben támasztott igényeket. Ehhez először is a transzputerek összekötési filozófiáját változtatták meg [inmos91].

Az új T9000-es transzputer mellé bevezettek egy speciális kommunikációs chipet, a C104-es útvonalkereső áramkört. A C104 valójában nem más, mint egy hardverben megvalósított kommunikációs réteg. Ezek után a T9000 és a C104 együtt alkot egy ún. feldolgozó egységet oly módon, hogy a T9000 négy linkje a hozzátartozó C104-hez van kötve. A transzputerek korábbi pont-pont direkt összekötése helyett az így létrehozott feldolgozó egységek vannak egymáshoz kötve a bennük elhelyezett útvonalkereső áramkörök segítségével, vagyis nem a T9000-ek vannak összekötve, hanem a C104-ek!

A T9000-en belül egy Virtuális Csatorna Processzor (VCP) gondoskodik arról, hogy a T9000 négy hardver linkjére tetszőleges számú szoftver csatornát lehessen leképezni. Ráadásul a VCP, mint egy önálló processzor van kiképezve, amely nem lassítja a T9000 aritmetikai és logikai feldolgozási sebességét. Így módon az 1. generációs transzputerek leképezési problémája teljesen megszűnik anélkül, hogy ez a legkevésbé is csökkentené a soktranszputeres rendszer feldolgozási kapacitását. Mindezen hardver támogatások eredményeképpen a 4 linken keresztül a teljes átviteli sebesség összesen 80 Mbyte/sec.

A T9000 természetesen további jelentős újításokat is tartalmaz, melyek nemcsak az 1. generációs transzputerekhez képest, de jelenlegi versenytársaihoz képest is komoly előnyöket rejtenek. Így például a belső sínrendszer helyett alkalmazott crossbar, az 5-szintű utasítás pipeline az utasítás csoportosítóval, valamint a 3-szintű cache rendszer lehetővé teszi a 200 MIPS és 25 MFLOPS csúcsteljesítmény elérését transzputerenként.

A Parsytec cég 1992-ben jelentette be GigaCube nevű szuperszámítógépét, amely maximális kiépítésben 16K T9000-es processzorból áll. A GigaCube gép érdekessége, hogy az összekötési topológia egy 3-dimenziós térrácsnak felel meg.

6. A HTUG CÉLJA ÉS PROGRAMJA

A HTUG-ot (Hungarian Transputer User Group) 1991-ben az NJSZT keretein belül hoztuk létre a WoTUG (World Transputer User Group) magyarországi szekciójaként. A HTUG elsődleges célja szakmai fórum biztosítása mindazoknak, akik transzputerek alkalmazásával, illetve általában párhuzamos számítási rendszerek alkalmazásával, kutatásával, fejlesztésével, oktatásával foglalkoznak, vagy a jövőben foglalkozni kívánnak.

A szakmai fórumnak több lehetséges formáját egymás mellett kívánjuk alkalmazni a minél szélesebb körű tájékoztatás és részvételi lehetőség érdekében. Ilyen lehetséges formák a következők:

- a. egyedi előadások tartása
- b. ankétok szervezése
- c. szakmai bemutató napok
- d. nyári iskola
- e. workshop

Az eddig megvalósult és szervezés alatt álló programjaink is ezt a széleskörű tevékenységet reprezentálják. Példaként megemlítjük a BME Folyamatszabályozási tanszékének transzputeres tevékenységét bemutató ankétunkat, melynek fő témája transzputerek alkalmazása a robotvezérlésben és a képfeldolgozásban volt.

Erősíteni kívánjuk kapcsolatainkat külföldi társszervezetekkel és hasonló területen dolgozó kutatókkal. Így például 1992. október 8-10-én az Osztrák Párhuzamos Számítási Központtal közösen szervezünk workshop-ot Győrben "First Austrian-Hungarian Workshop on Transputer Applications" címmel, melyre az osztrák és magyar alkalmazók, kutatók mellett szlovák és lengyel előadókat is szeretnénk meghívni.

Szeretnénk, ha a HTUG tényleg országos szervezatként működne és tevékenysége nem korlátozódna a fővárosra. Több vidéki rendezvényt is tervezünk, melyre megintcsak példa a fenti workshop színhelyének megválasztása.

Különösen fontosnak tartjuk, hogy a transzputeres és ezen keresztül a párhuzamos számítási rendszerek technológiája az oktatásban is tért hódítson. Ennek érdekében tevékenyen támogatjuk a Kandó Kálmán Műszaki Főiskola székesfehérvári Informatika Intézetében és a Miskolci Egyetem Informatika Intézetében induló transzputeres oktatás előkészítését. Pályadíjat tüztünk ki az ÖTGY pályázat országos döntőjébe jutott transzputer vagy párhuzamos számítási témájú dolgozatai számára. Felméréseink szerint Magyarországon jelenleg viszonylag kevés cég, intézet, főiskola, egyetem foglalkozik transzputerekkel, vagy azzal kapcsolatos tevékenységgel. Ugyanakkor a fejlett országok számítástechnikai ipara rohamléptekkel halad a párhuzamos számítási rendszerek és ezen belül a transzputerék

irányába. Reméljük, hogy szervezett népszerűsítő és szakmai tevékenységünk eredményeképpen egyre több helyen figyelnek fel erre a rendkívül fontos új esz-
közre és irányzatra, és kezdik alkalmazni majd az ipari gyakorlatban is.

REFERENCIÁK

- [Hoar78] C.A.R. Hoare: Communicating Sequential Processes, Communications of the ACM, 1978. Aug.
- [Hoar85] C.A.R. Hoare: Communicating Sequential Processes, Prentice-Hall International, 1985.
- [Dijk75] E.W. Dijkstra: Guarded commands, nondeterminacy, and formal derivation of programs, Communications of the ACM, 1975. Aug.
- [inmos88] inmos: OCCAM 2 Reference Manual, Prentice-Hall International, 1988.
- [inmos89] inmos: The Transputer Development and iq Systems Databook, 1989.
- [inmos91] inmos: The T9000 Transputer Product Overview Manual, 1991
- [KaFF90] P. Kacsuk, I. Futó and Sz. Ferenczi: Implementing CS-Prolog on a Communicating Process Architecture, Journal of Microcomputer Applications, Vol.13, (1990) pp. 19-41.
- [FoTa90] I. Foster and S. Taylor: Strand: New Concepts in Parallel Programming, Prentice-Hall, Englewood Cliffs, N.J., 1990



Kacsuk Péter villamosmérnök

a műszaki tudományok kandidátusa, a KFKI-MSZKI tudományos főmunkatársa, a KFKI-MSZKI-ban a BME Villamosmérnöki Karával közösen megalakított Párhuzamos Számítási és Módszertani Központ vezetője. 1976 óta foglalkozik párhuzamos és multiprocesszoros rendszerekkel, 1984 óta fő kutatási területe a Prolog párhuzamos implementálása.

1986/87-ben és 90/91-ben a londoni egyetem Queen Mary and Westfield College-ának meghívására az 1024 processzoros DAP számítógépen és egy soktranszputeres rendszeren implementálta a Prolog nyelvet. 1991-ben a linzi egyetem vendégprofesszora az Osztrák Párhuzamos Számítási Központ meghívására. Az 1991-ben megalakult Hungarian Transputer User Group elnöke.



Ferenczi Szabolcs

a Kandó Kálmán Műszaki Főiskola Számítás-technikai Rendszerek szakán végzett 1983-ban. Jelenleg az ALL Számítástudományi Kutató Fejlesztő Kiszövetkezet munkatársa. Érdeklődési területe a párhuzamos és konkurrens programozás, a programozási nyelvek és rendszerek párhuzamosítása, valamint a számítástechnika és a filozófia kapcsolata.

Korábban a CS-Prolog sokprocesszoros implementációjában vett részt. Az 1991-ben megalakult Hungarian Transputer User Group titkára.

PÁRHUZAMOS SZÁMÍTÁSI RENDSZEREK FEJLŐDÉSÉNEK NEMZETKÖZI TRENDJEI ÉS VÁRHATÓ HATÁSUK A MAGYARORSZÁGI INFORMÁCIÓTECHNOLÓGIÁRA

Kacsuk Péter - Erényi István
KFKI Mérés és Számítástechnikai Kutatóintézet
Párhuzamos Számítási és Módszertani Központ
1525 Bp. Pf. 49.

1. BEVEZETÉS

Párhuzamos számítási rendszernek nevezünk minden olyan számítógépet, melyben egyenél több processzor alkalmazható egy adott program végrehajtására. Az ilyen párhuzamos számítógépek frontáttörése világszerte megtörtént, sorra jelennek meg az un. személyi szuperszámítógépek, melyek ára lassan összemérhetővé válik a hagyományos munkaállomásokéval. Egyes becslések szerint az évtized közepére már csak sokprocesszoros számítógépek kerülnek piacra.

A sokprocesszoros számítógépeknek ezt a frontáttörését elsősorban a félvezető technológia rohamos fejlődése hozta magával és ennek következtében ezen számítógépek programozási technológiája messze elmaradt a hardver technológia mögött. A párhuzamos számítógépek programozása számos új nehézséget vet fel (programok párhuzamos szervezése, programok optimálist megközelítő szétosztása a processzortérben, hibakeresés párhuzamos programokban, stb.), melyek megoldása még várat magára, de világszerte intenzív kutatás tárgya.

Az előadásban egyrészt felvázoljuk azokat a legfontosabb trendeket, amelyek a párhuzamos számítógépek várható fejlődési irányait leginkább meghatározzák a jövőben, másrészt beszámolunk egy olyan hazai kezdeményezésről, amely a hazai számítástechnikai társadalmat kívánja a fenti irányzatok gyors és hatékony befogadására felkészíteni.

2. PÁRHUZAMOS SZÁMÍTÓGÉP ARCHITEKTÚRÁK

Nagyon nehéz és hálátlan feladat a nagyteljesítményű párhuzamos számítási rendszerek fejlődését előre jelezni, hiszen egy előre nem látható technológiai változás vagy architekturális újítás a jelenleg látható fejlődési irányokat teljesen megváltoztathatja. A fejlődést meghatározó ma látható leglényegesebb tényezőkről szólunk az alábbiakban.

A nagybonyolultságú integrált áramkörök (VLSI) alacsony gyártási (sokszorozási) költségeinek köszönhetően igen erős az az irány, hogy nagyszámú, azonos processzorból (processzorok térbeli multiplikálásával) készítsenek párhuzamos számítógépeket (mint pl.: DAP, Connection Machine, Intel Touchstone gépei, szisztolikus tömbprocesszorok, Transputer típusu processzorokból felépülő gépek, stb.) [Hock87], [Jess87].

Ugyancsak gazdaságossági szempont a párhuzamos gépeknek a programokkal való ellátása. A programfejlesztés, írás és hibakeresés eszközeiről és problémáiról később még szólunk. Itt most a már meglévő szekvenciális programok párhuzamos gépekre történő áttelepítésének a kérdésére kívánunk utalni. Nagyon sok esetben, - ilyen pl.: az azonos típusú processzorokból felépített (homogén) gépek kategóriája is -, a meglévő szoftvert nehéz (vagy lehetetlen) átfordítani úgy, hogy az a párhuzamos gépen hatékonyan futtatható legyen [Har186].

A homogén felépítésű gépekben mind a feldolgozási, mind a tárolási képességet egyenletesen osztják szét a processzor térben. Ezekben a gépekben a feldolgozás és tárolás mellett megjelenik egy újabb kritikus művelet, a processzorok közötti kommunikáció. A lényeges és meghatározó különbség ennek megfelelően egy három elemű vektorral jellemezhető: $v(p,s,c)$, ahol p az alkalmazott processzorok feldolgozási képességét jelöli, s az egyes processzorok által közvetlenül elérhető ún. saját memória kapacitását, és c a processzorok közötti kommunikáció topológiájára és annak áteresztő képességére jellemző értéket jelöl. Az így bevezetett vektor egyértelműen jelölheti ki a különböző párhuzamos számítógépek helyét egy háromdimenziós térben, mely a fenti gépek összehasonlítására egy szemléletes módszert biztosít.

Igen nagyszámú (néhány ezer vagy tízezer) processzor esetén lehetetlen teljes összeköttetést kialakítani, így a kommunikációs topológia létrehozása csak kompromisszumok alapján lehetséges. Ha a kommunikációs hálózat topológiája rögzített, akkor a rendszer csak bizonyos feladatokhoz használható hatékonyan. Alternatívát jelenthet a programozhatóan átkonfigurálható hálózatok kialakítása, ennek hátránya azonban, hogy további programozási feladatokat ró a felhasználóra, másrészt az átkonfigurálás természetesen a futási időt is megnöveli. Néhány sokprocesszoros konfigurációról az alábbiakban adunk egy-egy mondatos értékelést.

Tömbprocesszorok:

szomszédsági kommunikáció hatékony, (egyszerre mozognak az adatok, a kommunikáció is párhuzamos),

Szisztolikus tömbök:

hatékony kommunikáció szomszédsági műveletekre terjed ki, a processzállással átlapoltan történik,

Közös memóriás multiprocesszorok:

hatékony cache protokollokkal elérhető, hogy viszonylag ritkán kell adatblokkokat cserélni a cache és a közös memória között,

Elosztott memóriás, kommunikáló multiprocesszorok:

egyszerű "store and forward" technika helyett hatékonyabb útvonalkeresésre van szükség (pl. wormhole módszer),

Adatfolyam (dataflow) gépek:

eddiggi sikertelenségük oka rossz hatékonyságú processzállási és adatmozgatási arányban keresendő. Új irányok keresését jelenti a makrodataflow és a hibrid dataflow elv.

A számítógépek teljesítőképességének növelésére irányuló másik eddigiekben is nagyon sikeres technika a pipeline feldolgozás, a pipeline felépítésű processzor egységek működési módja. Sikeres implementálására példa: Cray computer, X MP gép, stb. Ezekben a viszonylag kis-számú processzor mindegyike a közös osztott tárolót használja kommunikációs célra is. Kedveltségüket a tudományos számításokban gyakran jelentkező vektor műveletekhez való alkalmasságukon túlmenően nem kis mértékben a jól felépített vektorizáló kompájlereknek is köszönhetik, melyekkel a meglévő soros programok könnyen átfordíthatók a számítógépek részére [Hock87].

A modern félvezető technológia lehetővé teszi, hogy az algoritmusok egy széles csoportjára közvetlenül algoritmust realizáló ún. algoritmus-orientált felépítésű feldolgozó egységeket hozzunk létre [LSILogic]. Igaz, az egység így csak egy speciális célra használható, de a tudományos számítások és feldolgozások számos terén elég sok szabvány algoritmust használnak. Ez a közvetlen hardver implementálást gazdaságossági szempontok alapján is egyre indokoltabbá teszi (pl.: kép- és jelfeldolgozás terén).

3. KONKURENS ÉS PÁRHUZAMOS PROGRAMOZÁSI NYELVEK TRENDJEI

A jelenlegi szakirodalomban a konkurens és párhuzamos programozás fogalma nem teljesen tisztázott, gyakran keverednek és összemosódnak. Ezért első lépésként e két fogalmat kell pontosítani és helyére tenni.

Sokprocesszoros környezetben ha az eredeti programot úgy bontjuk egyidejűleg futó részprogramokra, hogy azok egymással nem kommunikálnak és erőforrásokért nem versenyeznek, akkor az eredetileg determinisztikusan futó program a párhuzamos dekomponálás után is determinisztikus marad. A sokprocesszoros rendszerek ilyen jellegű programozását a továbbiakban párhuzamos programozásnak nevezzük.

Ha azonban az eredeti programunkat olyan egyszerű részprogramokra bontjuk fel, amelyek az eredeti program feladatát úgy oldják meg, hogy miközben egyidejűleg futnak, egymással kommunikálnak is, esetleg erőforrásokért versenyeznek, akkor a kommunikáció és versenyhelyzetek következtében a konkurens programok nemdeterminisztikussá válhatnak, még akkor is ha az eredeti program determinisztikus volt. Ha a párhuzamos dekomponálás során szándékosan hozunk létre nemdeterminisztikus programot, akkor konkurens programozásról beszélünk, egyébként párhuzamos programozásról. A zavart a szakirodalomban feltehetőleg az okozza, hogy a párhuzamos programok hibás dekomponálás esetén nemdeterminisztikusan viselkedhetnek és ezért a konkurens programokkal megegyező tulajdonságokat mutatnak, ennek következtében gyakran ezeket is konkurens programoknak nevezik.

Ha azonban megvizsgáljuk a konkurens és párhuzamos programok alkalmazásának célját és megjelenési formáját a programozók felé, akkor világossá válik, miért nem célszerű a (hibás) párhuzamos programokat konkurens programoknak nevezni. A párhuzamos programozás célja minden esetben az eredeti program felgyorsítása a párhuzamos dekomponálás segítségével. Ideális kompájlereket feltételezve ez úgy történik meg, hogy a programozó észre sem veszi, hogy programja egy- vagy többprocesszoros környezetben fut. A vektorizáló FORTRAN kompájlerek például teljesen eltakarják a programozó elől az alkalmazott párhuzamos számítógép architektúráját. Valóban, például egy fizikust nem az érdekli, hogyan dekomponálja programját párhuzamos számítógépekre, hanem hogy korábban megírt programja egyre gyorsabban fusson függetlenül attól, hogy milyen az alkalmazott számítógép architektúrája.

Konkurens programozás esetén azonban a programozó nagyon is tudatában akar lenni, hogy programját milyen módon dekomponálja, mivel az egyes részfeladatoknak (processzeknek) kitüntetett szerepe lehet. A konkurens programozás tipikus alkalmazási területei az operációs rendszerek, valós-idejű folyamatirányító rendszerek, objektum orientált szimuláció, kommunikációs hálózatok, stb. Egy folyamatirányító rendszerben például a gyártási folyamat különböző elemeihez különálló felügyelő processzeket célszerű rendelni, amelyek a felügyelt rendszer változásaira egymás közötti kommunikáció révén reagálnak.

A párhuzamos és konkurens programozásnak ez a fajta megkülönböztetése segít állást foglalni a sokprocesszoros számítógépek programozásáról régóta folyó vitában is. Két alapvető nézet ütközik ebben a vitában. Az első szerint a párhuzamosítási lehetőségeket teljes egészében el kell takarni a programozók elől, míg a másik tábor ennek ellenkezőjét vallja és arra hivatkozik, hogy egy explicit párhuzamos algoritmus sokkal gyorsabb lehet, mint egy szekvenciális algoritmus, amelyet egy intelligens kompájlér fordít le egy párhuzamos gépre. A vita jelenleg is folyik, de néhány trend világosan látszik.

3.1 Konkurens programozás

A konkurens programozást elsősorban a közös memóriás és az elosztott memóriás sokprocesszoros gépek programozásában alkalmazzák. Mivel a konkurens programozást a rendszerprogramozás területén alkalmazzák, itt egyértelműen olyan nyelvekre van szükség, amelyek lehetővé teszik a programozó számára a program dekomponálását, továbbá eszközöket biztosítanak a dekomponálás során létrehozott processzerek kommunikációjára és szinkronizációjára.

A program dekomponálás eszköze általában a strukturálatlan FORK/JOIN és a strukturált COBEGIN/COEND processz kreálás. A párhuzamosság szempontjából korszerű nyelvekben (Occam) már egyértelműen az utóbbi használata dominál, de sajnos még mindig jelennek meg olyan újkeletű nyelvek (és ezek általában C bázisúak), amelyek még mindig lehetővé teszik a meglehetősen veszélyes FORK/JOIN használatát.

A programok kommunikációjának és szinkronizációjának szervezéséhez eredetileg a strukturálatlan szemaforokat vezették be, amelyeket később egyre nagyobb strukturáltságú eszközök követtek. A konkurens programozás szempontjából korszerűnek tekinthető nyelvek az alábbi eszközöket alkalmazzák:

- feltételes kritikus terület (Edison [Brin81])
- csatorna (Occam 2 [Inmo88], Strand [Fost90], Parallel C [3lld])
- monitor (Concurrent Pascal [Brin75], Modula-2 [Wirt82])
- távoli eljáráshívás (Ada [Dod81], DP [Brin78])

A monitor és távoli eljáráshíváson alapuló konkurens nyelvek előfutárai a sokprocesszoros objektum alapú nyelveknek [Anno90], amelyeket kifejezetten sokprocesszoros gépek objektum orientált programozására fejlesztettek ki. Tekintettel arra, hogy az öröklési tulajdonság sokprocesszoros megvalósítása még meg nem oldott probléma, ezek a nyelvek az öröklést nem tartalmazzák és ezért megkülönböztetésül nem objektum orientált, hanem objektum alapú nyelveknek nevezik őket.

3.2 Párhuzamos programozás

A párhuzamos programozás célja, mint láttuk elsősorban a számításigényes programok végrehajtásának jelentős felgyorsítása. Itt két fő alkalmazási területre kell megkülönböztetni:

- numerikus problémák (tudományos és műszaki számítások)
- numerikus problémák (műszaki alkalmazások)
- szimbolikus problémák (mesterséges intelligencia)

A tudományos és műszaki számítások területén az uralkodó programozási nyelv a FORTRAN, amelyet általában nem számítástechnikusok használnak. Ennek megfelelően itt olyan kompájlerok szükségesek, amelyek teljesen eltakarják a párhuzamos számítógép architektúrát a programozók elől. Ennek köszönhetően máig egyedülálló népszerűségüket a vektorprocesszorok, mivel ezekre több mint egy évtizede jó minőségű vektorizáló kompájlerok állnak rendelkezésre. Ugyanakkor minden olyan párhuzamos számítógép, amely arra kényszerítené a FORTRAN programozókat, hogy más programozási nyelvet használjanak, vagy akárcsal módosított FORTRAN-ban programozzanak, eleve kudarcra van ítélve [Berg88]. Ennek megfelelően jelenleg is intenzív kutatás tárgya FORTRAN kompájlerok készítése különböző típusú párhuzamos gépekre. Az utóbbi időben sikerült hatékony FORTRAN kompájlerokat kifejleszteni közös memórián keresztül kommunikáló

vektorprocesszorok számára. Ez azért különösen nagy jelentőségű előrelépés, mert a legújabb szuper-mikroprocesszorok legtöbbje már tartalmaz vektor művelet végző egységeket, és így az ezekből felépített sokprocesszoros gépek az új FORTRAN kompájler hatékonyan alkalmazhatók.

A műszaki alkalmazások területén tipikus numerikus problémák például a kép- és jelfeldolgozás. Ezekre és a hasonló műszaki alkalmazásokra általában nem általános célú számítógépet, hanem valamilyen speciális célszámítógépet használnak. Ezek tehát egyedi gépek, melyek programozása is történhet egyedi eszközökkel. Itt az elsődleges szempont a műszaki paraméterek kielégítése és ennek érdekében sok esetben valóban a párhuzamos algoritmusok használata szükséges. Ez az a terület tehát, ahol az explicit párhuzamosságot kínáló nyelvek a leginkább létjogosultak. Nem véletlen, hogy az Occam nyelv, amely egy speciális párhuzamos célnyelvnek tekinthető éppen ezen a területen vált népszerűvé, míg a tudományos számítások területén a FORTRAN azonnal visszaszorította.

A mesterséges intelligencia területén a Lisp és a Prolog, ill. ezek ötvözetei a legnépszerűbb nyelvek, de ezek alkalmazása nem annyira kizáró jellegű, mint a FORTRAN-é a numerikus számításokban. Itt is jelentős kutatások folynak ezen nyelvek párhuzamos implementációja körül. E nyelvek számára a legadekvátabb párhuzamos architektúráknak a dataflow és redukciós számítógépeket tekintették, de ezek első prototípusai csalódást keltettek. Kiderült, hogy a processzorközi kommunikációra fordított idő túlságosan nagy ezekben a géptípusokban és ezért teljesítményük messze elmarad a kívánatostól, azaz a processzor számmal arányos teljesítménynövekedést nem képesek produkálni. Ennek megfelelően újabban az un. hibrid dataflow gépeket kutatják [Gao91], amelyek alapvetően Neumann-elvű sokprocesszoros gépként működnek és csak akkor kapcsolnak át dataflow üzemmódba, ha ezt a programban rejlő inherens párhuzamosság megkívánja. Ezzel egyidejűleg intenzíven kutatják Lisp és Prolog kompájlerek készítését Neumann-elvű sokprocesszoros gépekre is. Ezen a területen ért el jelentős eredményt a nemzetközi Gigalips projekt [Lusk90].

Ugyanakkor mivel ez a terület viszonylag fiatalabb, mint a numerikus számításoké, az itteni felhasználók nyitottabbnak tűnnek egy eredendően párhuzamos mesterséges intelligencia nyelv befogadására. Ennek megfelelően komoly kutatások folynak ilyen nyelvek kialakítására. Ezen törekvések egyik legsikeresebb terméke a Strand88 nyelv [Post90], amely egy Prolog-szerű párhuzamos nyelvi környezetbe képes korábban kifejlesztett szekvenciális eljárásokat befogadni.

4. KONKURENS ÉS PÁRHUZAMOS PROGRAMOZÁSI MÓDSZERTAN

A konkurens és párhuzamos programok fejlesztésének módszertana messze elmaradt a párhuzamos számítógépek hardver fejlesztési módszertanától. Ezen a területen erőteljes kutatások folynak és itt néhány kezdeti eredményről kívánunk beszámolni.

Mint láttuk a konkurens programok eredendően nemdeterminisztikusak, ezért belövésük nagyságrendekkel nehezebb, mint a szekvenciális programoké. Ebből következik, hogy ezen programokat lényegesen gondosabban kell megírni, mint szekvenciális társaikat, vagyis olyan programozási módszertanra van szükség, amely minimalizálja a programozás közben elkövethető hibákat. Ez az oka annak, hogy a korszerű konkurens programozási nyelvek magasszintű, strukturált szinkronizációs és kommunikációs eszközöket kínálnak a programozó számára. Ezek egyik előnye, hogy a kompájler már fordítási időben képes kiszűrni számos szinkronizációs hibát. Másik előnyük, hogy a processzek szinkronizációját a program jól definiált pontjaira koncentrálják és így szinkronizációs hiba esetén elegendő a program idevágó részeit analizálni.

Konkurrens programok belövésére a hagyományos programbelövési módok, mint a lépésenkénti végrehajtás módszere, vagy a töréspontok alkalmazása nem lehetséges, mivel ezek mind a program determinisztikus jellegén alapulnak. A determinisztikus viselkedés jellemzője, hogy hibás program esetén is igaz, hogy akárhányszor lefuttatva a hibás programot ugyanazzal a bemenő értékkombinációval mindig ugyanaz a kimenő értékkombináció keletkezik. Mivel ez a feltétel konkurrens programokra nem teljesül, ezért a hibás lefutás nem reprodukálható és így a hiba oka kísérleti módszerekkel nem határozható meg.

Ezen a problémán elvben segíthetne egy olyan kimerítő adatgyűjtés, amely a program minden állapotváltozását regisztrálná futás közben. Ez azonban már viszonylag kis programok esetén is olyan hatalmas adatmennyiséget jelentene, amelynek tárolása és feldolgozása is túlságosan nagy problémát okozna.

A fenti probléma megoldására javasolták az ún. visszajátszási technika bevezetését [Lebl87], amely a strukturált szinkronizációs eszközök kiterjesztésén alapul. A módszer lényege, hogy az állapotváltozásokról az adatgyűjtés csak a szinkronizációs eszközök használatakor történik meg, mégpedig oly módon, hogy ezen információ alapján a program ismételten úgy lefutatható, hogy a processzek egymáshoz viszonyított sebessége (szinkronizációja) pontos másolata lesz az előző futásának.

Párhuzamos programok írásakor két dologra kell figyelni. Egyrészt el kell kerülni a nemdeterminisztikus viselkedést eredményező programozási hibákat. Ezek elsősorban közös változók versenyszerű írásának és olvasásának révén állnak elő. Az ilyen jellegű hibák kiszűrésére sok esetben a forrásprogram analízise is elegendő és így a nemdeterminisztikus programbelövés sok esetben elkerülhető [Emra92].

A másik szempont párhuzamos programok készítésénél a processzorok jó hatásfokú kihasználásának biztosítása. Ez egyrészt azt jelenti, hogy az algoritmus ne tartalmazzon túl sok kommunikációt, ami a processzorok hasznos feldolgozási idejét csökkenti, másrészt a processzorok terhelése legyen közel egyenletes. Az egyenletlen processzor terhelés jelentős teljesítmény csökkenést okoz a párhuzamos rendszerekben.

Ennek megfelelően párhuzamos programok fejlesztéséhez olyan mérő programok szükségesek, amelyek lehetővé teszik a processzorok terhelésének, továbbá a kommunikációban eltöltött processzor időnek a mérését. Ezek a mérőprogramok általában futási időben adnak vizuális információt a párhuzamos programok futási jellemzőiről és így nagy mértékben segítik a programozót a hatékony párhuzamos algoritmus kifejlesztésében [Bemm91].

5. PÁRHUZAMOS SZÁMÍTÁSI ÉS MÓDSZERTANI KÖZPONT

A párhuzamos számítógépek elterjedése és alkalmazása olyan hatalmas léptékkel folyik világszerte, hogy önálló kutatásukra, fejlesztésükre, oktatásukra és gyakorlati alkalmazásba vitelük támogatására a világ fejlett országaiban sorra alakulnak a párhuzamos számítási központok egyetemi, területi és országos szinteken. Hazánk európai felzárkózása az információtechnológiában nem képzelhető el ennek a területnek aktív művelése nélkül. Ennek a felismerésnek a jegyében hoztuk létre a KFKI Méréstechnikai és Számítástechnikai Kutatóintézetben belül a BME Villamosmérnöki Karának együttműködésével a Párhuzamos Számítási és Módszertani Központot (PSZMK), mely a tervek szerint a következő feladatokat látja el:

1. Párhuzamos számítási technológia fejlesztése és kutatása tudományos iskola jelleggel graduális, posztgraduális képzésben résztvevő hallgatók bevonásával.

2. Párhuzamos számítógép kultúra meghonosítása és oktatása.

3. Párhuzamos számítási rendszerek alkalmazásba vitelének támogatása.

Olyan know-how központot szeretnénk kialakítani, amely képes a párhuzamos számítógépek gyakorlati alkalmazásában felmerülő problémák megválaszolására és a párhuzamos számítási rendszerekkel kapcsolatos információk széles körű terjesztésére. Ennek érdekében szándékunkban áll néhány tipikus párhuzamos számítógép beszerzése, ill. hálózatot keresztüli elérésének biztosítása. A fenti célok elérése érdekében szívesen együttműködnénk minden olyan hazai és külföldi intézménnyel, amely a párhuzamos számítási rendszerek kutatásában, hazai oktatásában vagy terjesztésében aktív szerepet kíván játszani.

HIVATKOZÁSOK

[Anno90] J.K. Annot and P.A.M. den Haan, POOL and DOOM: The Object-Oriented Approach, In Parallel Computers: Object-Oriented, Functional, Logic, (ed. P.C. Treleaven), Wiley, 1990, 47-79

[Bemm91] T. Bemmmerl and A. Bode, An Integrated Environment for Programming Distributed Memory Multiprocessors, Proc. of Conf. on Distributed Memory Computers, 1991, 130-142

[Berg88] D. Bergmark, Parallel Programming Languages for Scientists, CONPAR'88 conf., Manchester, 1988, vol. B, 1-10

[Brin75] P. Brinch Hansen, The Programming Language Concurrent Pascal, IEEE Trans. on Software Eng., vol. 2, no. 6, 1975, 151-164

[Brin78] P. Brinch Hansen, Distributed Processes: A Concurrent Programming Concept, Commun. ACM, vol. 21, no. 11, 1978, 934-941

[Brin81] P. Brinch Hansen, Edison: A multiprocessor language, Software Practice and Experience, vol. 11, no. 4, 1981, 325-361

[Dod81] US Dep. of Defense, Programming Language Ada: Referenc Manual, vol. 106, Lectures Notes in Computer Science, Springer-Verlag, 1981

[Emr92] P.A. Emrath et al, Detecting Nondeterminacy in Parallel Programs, IEEE Software, no 1, 1992, 69-77

[Fost90] I. Foster and S. Taylor, Strand New Concepts in Parallel Programming, Prentice Hall, 1990

[Gao91] G.R. Gao et al., Towards and Efficient Hybrid Dataflow Architecture Model, PARLE'92 conf., 355-371

[Inno88] INMOS Ltd, occam 2 Reference Manual, Prentice Hall, 1988

[Harl86] Harland, D.M., Concurrency and Programming Languages, Ellis Horwood, 1986

[Hock87] Hockney, R.W., Supercomputers - Past, Present and Future, in Parallel Processing State of the Art Report, Pergamon Infotech Ltd, 1987, 51-57

[Jess87] Jesshope, J.C., A Dynamic Load-Balanced Active-Data Model of Parallel Processing for Vision, in Proc. BSC Workshop on Parallel Processing and Vision, Oxford University, 1987

[Lebl87] T.J. Leblanc and J.M. Mellor-Crummey, Debugging Parallel Programs with Instant Replay, IEEE Trans. on Computers, vol. C-36, no. 4, 1987, 471-481

[LSILogic] LSI Logic #L64000 Series Signal and Image Processing VLSI Circuits Data sheets

[Lusk90] E. Lusk et al., The Aurora OR-Parallel Prolog System, New Generation Computing, vol. 7, no. 2,3, 1990, 243-271

[Wirt82] N. Wirth, Programming in Modula-2, Springer-Verlag, 1982

[3ltd] 3L Ltd, Parallel C User Guide, 1988



Erényi István villamosmérnök
a műszaki tudományok kandidátusa, a KFKI
Mérés- és Számítástechnikai Kutató Intézet
Informatika Főosztály vezetője. Több mint
éve foglalkozik a multiprocesszoros rendsz
ekkel és speciális célú, elsősorban képf
dolgozás terén alkalmazott párhuzamos fel
dolgozó egységek kérdéseivel. Mellékfogl
kozású oktatóként résztvesz a BMR villamos
mérnök és informatika szakos hallgatók ké
zésében. 1986-88 között a Florida Intern
tional University-n vendégprofesszori áll
vállalt.

Fejlett kommunikációs lehetőségek hatása az emberi döntéshozatalra

Biró Miklós

MTA SzTAKI

1. Bevezetés

Az új hardver és szoftver technológiák által kínált lehetőségek általában nagy hatást gyakorolnak a gazdasági fejlődésre és mindennapi életünkre. Itt, e témakörön belül, a fejlett kommunikációs lehetőségek hatását elemezzük az emberi döntéshozatalra.

Különleges jelentőséget kölcsönöznek ennek a területnek a közelmúlt és a közeljövő gazdasági, politikai változásai. Ide tartoznak például a Kelet-Európában zajló változások, az Európai Közösség egységes belső piacának létrehozása és a környezetvédelem szerepének növekedése.

Ezek a tényezők növekvő igényt támasztanak mind a fejlett kommunikációs lehetőségek biztosításának, mind a fokozottabb versenykörnyezetben jelentkező döntési helyzetek támogatásának irányába. A két irány metszéspontjában helyezkedik el a csoportos döntéstámogatás.

A következő szakaszban röviden összefoglaljuk a csoportos döntéstámogató rendszerek előnyös hatásait az emberi döntéshozatali folyamatra. Ezután olyan matematikai, pszichológiai, vagy szociológiai alapú tényezőket tárgyalunk, amelyek a fejlett kommunikációs eszközöket használó csoportos döntéstámogató eszközök elterjedését hátrányosan befolyásolják. Végül összefoglalunk néhány alkalmazást, ahol a fejlett kommunikációs eszközök a fentiek ellenére is jelentős hatékonyság javulást eredményezhetnek.

2. Előnyös hatások

E területen a legalaposabb elemzéseket az Arizonai Egyetem elektronikus értekezleti rendszerével (EMS=Electronic Meeting System) végezték. A kísérletek először laboratóriumi környezetben folytak, majd számos egyetemre és céghez is telepítették a PLEXSYS elnevezésű rendszer továbbfejlesztéseit. Ezek alapvetően egyazon helyen és időben lefolytatott értekezletek támogatását szolgálták. Mivel minderről az irodalomban bőséges leírás található

[Nunamaker, J.F. Jr., Applegate, L.M., Konsynski, B.R. (1988)], [Nunamaker, J.F. Dennis, A.R., Valacich, J.S., Vogel, D.R., George, J.F. (1991)], [Martz, W.B., Vogel, D. Nunamaker, J.F. Jr. (1992)], itt nem megyünk a részletekbe, csupán az alapeszköz felvázolása mellett felsorolunk néhány előnyös hatást.

Az alapeszközök négy osztályba sorolhatók: gondolat generálás, gondolat rendszer rangsorolás, döntés megfogalmazás és értékelés. A gondolat generálás legfontosabb eszköz elektronikus agyfuttatás (talán nem rossz fordítása a brainstorming szónak). A gondolat rendszerezés során a korábban generált nagyobb mennyiségű információból kikristályosított alternatíva halmaz áll elő. A rangsorolás támogatása különböző szavazási több kritériumú döntési eljárások biztosításával történik. A döntés megfogalmazásához értékeléséhez kidolgozásra került egy az érdekelt feleket azonosító és érdekeltégi szintek feltáró (Stakeholder Identification and Assumption Surfacing) eszköz.

Az előnyös hatások közül csak néhányat sorolunk fel, melyek egyébként elektronikus támogatás hiányában a csoportos döntési folyamatot gátló tényezőkként jelentkeznek.

-Hozzászólási idő korlát: EMS használatakor a gondolatok átadása különböző gépeken és időben is történhet.

-Gondolat feltárási akadályok: Elektronikus támogatás hiányában gondolatok veszhetnek el, valaki nem kap időben szót és elfelejti, vagy már nem is tartja elég eredetinek mondanivalóját. Gyakori eset, hogy valaki annyira összpontosít az előadott szövegre, hogy nem tud közben időt fordítani a saját gondolataira. Az előbbi eset párja az, amikor valaki szóhoz jutásáig saját mondanivalója fejbentartására összpontosít, így az előadót is követni tudja, és újabb gondolatai sem születnek.

-Memória korlátok: Mások hozzászólása elektronikus memória hiányában könnyen elfelejthető.

-Gátlások: Az EMS-ben magvalósítható névtelen hozzászólási lehetőség felszabadítja az udvariasság, megtorlási fenyegetettség, vagy önbizalom hiánya miatt elfolytott gondolatokat.

-Potyázás: Az EMS a névtelen hozzászólások miatt egyrészt növelheti egyes résztvevők passzivitását, másrészt azonban passzivitás csökkentő hatása is van, mivel a szóhoz jutni nem kell megküzdeni.

-Kognitív tehetetlenség: Szemtől szembe beszélgető partnerek tartózkodnak attól, hogy a beszélgetés gondolati irányába nem közvetlenül illeszkedő megjegyzéssel zavart okozzanak, ezért nehezen tudnak egy gondolatkörből kitörni.

-Csevegés: Az EMS kiküszöböli a feladattal nem összefüggő beszélgetés hatékonyság rontó hatását. Valamilyen szintű személyes kapcsolattartás azonban nyilván fontos a csoportos döntési folyamatban. Ebben komoly szerepe lehet a multimédia eszközöknek.

-Elnyomás: Az EMS kizárja a tekintély vagy hangerő alapú befolyásolást és időkiszájtítást.

-Információ özön: EMS esetén ugyan lényegesen megnő a feldolgozandó információ mennyisége, a feldolgozás azonban az elektronikus memória miatt könnyebben végezhető el.

-Koordinációs problémák: Az EMS kellően rendszerezi az értekezlet folyamatát ahhoz hogy elkerülhető legyen az egy helyben forgás vagy az elhamarkodott döntés.

-Információ hiány: Az EMS-en keresztül könnyen elérhetőek a szükséges információk.

-Felületes feladat elemzés: Az EMS ösztönzi és támogatja a feladat alapos elemzését, így elkerülhetőek a felületes megértésből eredő viták.

3. Hátrányos tényezők

3.1. A szociális választási problémák megoldhatatlansága

Egy szociális választási problémát elméletileg tekinthetünk egy speciális több kritériumú döntési feladatnak, ahol a kritérium rendszer egy-egy részét képezik az egyes szavazók kritériumai. Könnyen látható azonban, hogy itt egy szokásos súlyozott átlagoláson alapuló mechanikus rendszer nem alkalmazható, mivel ebben az esetben szociális konszenzusra van szükség, szemben a kizárólag élettelen kritériumok esetével, amelyek sosem tiltakoznak [Bíró,M., Csáki,P., Vermes,M. (1991)].

A csoportos döntéshozatal legegyszerűbb és egyben klasszikus esete a választás szava útján. Közismert, hogy alapvető nehézség merül fel, ha a jelöltek egyike sem kapja meg szavazatok abszolút többségét. Ilyenkor második fordulót szoktak tartani, amelyben már csak a két legtöbb szavazatot elért jelölt vesz részt. Könnyen előfordulhat azonban, hogy a harmadik jelölt nagyobb mértékben felelne meg a többségnek, mint az előbbi módszer megválasztható jelöltek bármelyike.

Természetes megoldásként kínálkozik az a módszer, hogy a szavazók ne csak egy jelölt adjanak meg, hanem az összes jelöltet rangsorolják. Probléma azonban ezután is maradt. Lássunk egy egyszerű példát.

9 szavazó: $\{s_1, \dots, s_9\}$, 3 jelölt: $\{j_1, j_2, j_3\}$.

s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
j_1	j_1	j_1	j_1	j_2	j_2	j_2	j_3	j_3
j_3	j_3	j_3	j_3	j_3	j_3	j_3	j_2	j_2
j_2	j_2	j_2	j_2	j_1	j_1	j_1	j_1	j_1

Jól látható, hogy a fenti két fordulás szavazási módszer esetén j_2 lenne a végső győztes, míg fölényesen győzne külön-külön mind j_1 -el, mind j_2 -vel szemben.

Az előbbi példa esetén sikeresen alkalmazható az egyszerű többségi szabály, mely szerint csoportos rangsorban egy jelölt akkor előz meg egy másikat, ha a többség rangsorban megelőzi. Ez a módszer vezet Marquis de Condorcet (1785) szavazási paradoxonához, mely a következő példa illusztrál.

3 szavazó, 3 jelölt.

s_1	s_2	s_3
j_1	j_2	j_3
j_2	j_3	j_1
j_3	j_1	j_2

Az egyszerű többségi szabály azonnal ellentmondáshoz vezet, hiszen az általa előállított reláció nem tranzitív. A többség j_1 -et j_2 elé, j_2 -t j_3 elé, de ugyanakkor j_3 -at j_1 elé sorolja.

Egy másik, ugyancsak régi szabály Jean-Charles de Borda (1781) nevéhez fűződik. E szerint egy jelölt csoportos rangját úgy kapjuk meg, hogy az összes szavazóra összegezzük az adott jelölt alá rangsorolt jelöltek számát. E szabály szerint az utóbbi 3 jelölt mindegyike azonos (3) súlyú. Ez megfelel az adott esetben várt eredménynek. Könnyen ellenőrizhető, hogy a Borda szabály az első példa esetén is elfogadható eredményt ad. A teljesség kedvéért azonban lássunk egy példát, ahol a Borda szabály mond csődöt.

5 szavazó, 6 jelölt.

s1	s2	s3	s4	s5
j1	j1	j1	j1	j2
j2	j2	j2	j2	j3
j3	j3	j3	j3	j4
j4	j4	j4	j4	j5
j5	j5	j5	j5	j6
j6	j6	j6	j6	j1

A Borda szabály szerint e szavazás nyertese 21-es szúllyal j2. A j1 jelölt súlya csak 20, holott j1-et ötből négy szavazó a legjobbnak ítélte.

A fenti esetekhez ad matematikai megközelítést [Arrow, K.J. (1951)]. Formálisan bizonyított Lehetetlenségi Tételének lényege, hogy nem létezik olyan szabály, amely minden szavazási variációra kielégít néhány természetesnek elfogadható követelményt (axiómát).

3.2. Talleyrand herceg döntéselőkészítő módszere

Napoleon külügyminisztere a diplomáciatörténet kiemelkedő alakja. Egy, az ellenséges szövetségekkel folytatandó tárgyalás előtt a császár kifejtette, hogy tárgyalási pozícióinak javítása érdekében szükséges még egy csata megnyerése. Talleyrand azonban finoman közölte vele, hogy erre semmi szükség, hiszen őrá bízták a megegyezési lehetőségek kidolgozását. Módszerének lényege, hogy jól megválasztott alternatívák célszerű sorrendben történő tálalása esetén nagy pontossággal előre látható a döntés [Talleyrand-Périgord, Charles Maurice de, Prince de Bénévent (1899)].

A módszer alátámasztásához vizsgáljuk meg hogyan is gondolkodik az ember például egy tartós fogyasztási cikk megvásárlásakor. Először is igényt tartunk egy megfelelő választékra.

Tegyük fel, hogy öt különböző márka áll rendelkezésre különböző árakon, de a termék minőségéről semmi mást nem tudunk. A legdrágább biztosan túl drága. A legvalószínűleg gyenge minőségű. A második legolcsóbbnak is lehetnek gyenge minőségűek. Választásunk tehát vagy a második vagy a harmadik lehetőségre fog esni.

Az előbbi megfigyelések valójában gyakran megfelelnek a hétköznapi tapasztalatoknak, és könnyen vezethetnek mindenféle döntéstámogató eszköz elvetéséhez akkor is, amikor használata indokolt lenne.

3.3. Braes paradoxona

A [Braess, D. (1968)] cikkben leírt paradoxon egy olyan közlekedési környezetet mutat, ahol egy új úttösszeköttetés bevezetése rontja az úthálózat áteresztő képességét. Ez sokszor megfelel hétköznapi tapasztalatainknak is.

Ez a paradoxon a kommunikációs lehetőségek hatásának egy olyan példája, ahol egy (tehát csoportos szempontokat figyelmen kívül hagyó) optimális döntések együttese magadöntési helyzetnek olyan mértékű megváltozását idézi elő, amely az egyéni döntés ellenkezőjét teszi optimálissá.

3.4. A technológia gátló hatása

Mind a [Nunamaker, J.F. Jr., Applegate, L.M., Konsynski, B.R. (1988)] cikkben jelzett, a hétköznapi tapasztalatok azt mutatják, hogy kis csoportok esetén a beszélgető partnerrel élkedő technológia gátló hatása túlnôhet az előnyein. A fent leírt információfeldolgozó előnyök általában csak nagyobb csoportok esetén jelentkeznek. A gátló hatás csökken olyan stílusú értekezlet levezetésével, ahol csak egy személy kezeli a számítógépet, a képernyőt viszont mindenki látja, így elektronikus táblaként játszik szerepet.

Egy másik tényező, amely eleve szükségessé teheti a technológia igénybevételét, a tudáson alapuló módszerek bevetése [Biró, M., Maros, I. (1991)], [Biró, M.; Bodnyai, Bor, A.; Knuth, E.; Kovács, L. (1992)].

4. Fejlett kommunikációs lehetőségek kihasználására kész helyzetek

Az alábbiakban felsorolt néhány alkalmazás már ma is többnyire valóság, a fejlett kommunikációs lehetőségek elterjedése azonban szélesebb körű felhasználást eredményezhet.

-Emberek és eszközök rugalmas kihasználása építkezéseken.

Egy fejlett kommunikációs lehetőségekre építő projekt vezetés lehetővé teszi emberek és eszközök jobb kihasználását a külső vagy belső feltételek dinamikus változása esetén is. A kihasználható lehetőségek közé tartoznak telekonferencia eszközök, multimédia adatbázisok technikai és tervezési adatokkal, kapcsolatteremtés a megfelelő szakértővel beleértve esetleg egy problémás terület képét is.

-Bonyolult specifikációnak megfelelő termékek megrendelése.

A bonyolult specifikációk a gyakorlatban félreértelmezéseket és költséges hibákat eredményeznek. Fejlett kommunikáció esetén az eladó hatékonyan, és részletekbe menően egyeztetheti a rendelőt saját adatbázisai, a megrendelő és esetleg további szakértők között.

-Táv árverés.

Ehhez multimédia katalógus, valamint valós idejű audio-video kapcsolattartás szükséges.

-Elektronikus bemutatóterem.

Egyes üzletekben már ma is működnek olyan rendszerek, ahol a vevő számítógépes katalógus segítségével választhatja ki a neki megfelelő árut. Az ilyen katalógusok elérése történhetne azonban távolról is, hálózaton keresztül. Ennek további előnye az lenne, hogy a kiskereskedők saját adatbázisukat egyszerre több nagykereskedő adatbázisával köthetnék össze.

-Távoli diagnózis és tanácsadás.

Bonyolult berendezések és folyamatok diagnosztizálását kamerák és érzékelők segítségével egy szakértő távolról is végezheti. Ezen alkalmazás analogonja a távoli orvosi diagnózis és tanácsadás. Különösen kritikus eset lehet például egy műtét, ahol a beteg felnyitása után derül ki, hogy arra a szakorvosra lenne szükség, aki éppen a negyterületű kórház ellenkező szögletében van.

Hivatkozások

- [Arrow,K.J. (1951)] Social Choice and Individual Values. Wiley, New York.
- [Biró,M., Csáki,P., Vermes,M. (1991)] WINGDSS Group Decision Support System MS-Windows. In: Proceedings of the Second Conference on Artificial Intelligence, Budapest, Hungary, pp.253-274.
- [Biró,M., Maros,I. (1991)] Deep Knowledge for Group Decision Support. MTA Sci Report 42/1991.
- [Biró,M.; Bodroghy,E.; Bor,A.; Knuth,E.; Kovács,L. (1992)] The Design of DDM Distributed NEgotiation Support Shell. Proceedings of the IFIP WG 8.3 Conference "Decision Support Systems: Experiences and Expectations". North-Holland, (to appear)
- [Braess,D. (1968)]. Über ein Paradoxon aus der Verkehrsplanung. Unternehmensforschung 12,pp.258-268.
- [Nunamaker,J.F. Jr., Applegate,L.M., Konsynski,B.R. (1988)] Computer-Aided Deliberative Model Management and Group Decision Support. Operations Research, 36,6,pp.826-841
- [Nunamaker,J.F. Jr., Dennis,A.R., Valacich,J.S., Vogel,D.R., George,J.F. (1991)] Electronic Meeting Systems to Support Group Work. CACM, 34,7,pp.40-61.
- [Martz,W.B., Vogel,D.R., Nunamaker,J.F. Jr. (1992)] Electronic Meeting Systems, Reflections from the field. Decision Support Systems, 8,pp141-158.
- [Murchland,J.D. (1970)] Braess's Paradox of Traffic Flow. Transportation Research, 4,pp.394.
- [Talleyrand-Périgord,Charles Maurice de, Prince de Bénévent (1899)] Lettres Inédites Talleyrand à Napoléon, 1800-09. Perrin and Co, Paris.
- [Woolsey,R.E.D. (1991)] The Fifth Column: La Méthode de Charles-Maurice de Talleyrand Maximized Acceptance with Optimized Agendas. Interfaces, 21,6,pp.103-105.



Bíró Miklós - matematikus

1977-től dolgozik az MTA SZTAKI-ban. Egyetemi doktori fokozatát 1983-ban védte meg. 1978 óta operációkutatást és diszkrét programozást tanít az ELTE TTK-n. 1985-87. között a New Jersey Rutgers Állami Egyetemen volt tanársegéd, 85-87. között pedig a City University os New Yorkon, majd a St.Johns egyetemen dolgozott vendégprofesszorként. Kutatási területe a kombinatorikus optimalizálás, az ember-gép interakciós technika, valamint a döntés- és tárgyalástámogatás.

HIPERTEXT ÉS MULTIMÉDIA RENDSZEREK

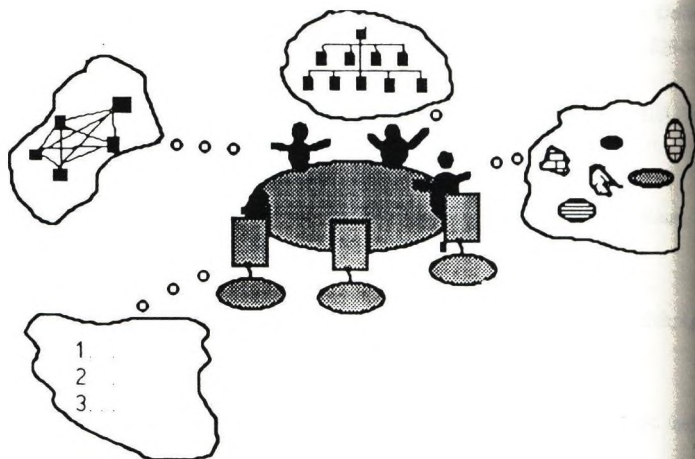
Remzsó Tibor

MTA SZTAKI

1132 Budapest, Victor Hugo u. 18-22.

1. Bevezetés

Az emberi gondolkodás többdimenziós jellege és a szokásos szövegek lineáris volta közötti különbség okozza azt, hogy a megszokott dokumentumok sokszor nehézkesen kezelhetők. Az emberi gondolkodás sokféle aspektussal, több dimenzióban működik. Memóriánk asszociatív, azaz különböző dolgokat kapcsolhatunk össze teljesen előre meghatározhatatlan módokon és sorrendben. Előfordul olyan eset és memória illetve gondolkodási típus is, amelyre a szó-ábrák asszociációja a legjellemzőbb. Léteznek olyan emberek is, akiknek a memóriájuk és gondolkodásuk teljesen strukturáltnak és hierarchikusnak tekinthető. A feladatunk az, hogy olyan információforrást hozzunk létre, amely képes a különbségek áthidalására, pontosabban képes arra, hogy egy környezetben belül ki tudjuk elégíteni többféle – az előzők szerint lényegesen különböző – alkalmazó információs igényeit és feldolgozási feladatait.



A hipertext típusú rendszereket és a kiterjesztésüknek tekinthető ún. hipermedia rendszereket éppen erre a célra lehet felhasználni.

2. Hipertext rendszerek

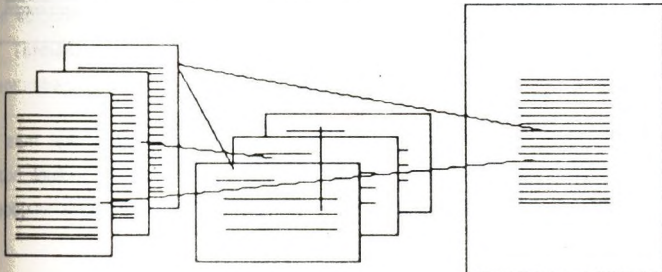
A hipertext rendszerek legfontosabb jellemzőjeként azt tekinthetjük, hogy lehetővé teszik egy szövegben azt, hogy a számítógépen tárolt szöveg valamely pontjához hozzákapszoljunk ugyanabban a szövegben egy másik részt, vagy egy másik szöveg egy darabját, ilymódon, ezen szerkezet mentén nemlineáris kapcsolat jön létre szövegrészek között, amelyet gombok (button) segítségével igen gyorsan bejárhatunk. A hipertext egyaránt tekinthető a szerzők eszközeként és az olvasók médiájaként. A hipertext segítségével szerzők vagy szerzői csoportok képesek egymással gondolataikat megosztani, létező szövegeket megjegyzésekkel ellátni, lábjegyzet szerű képződményeket létrehozni, amelyek lehetővé teszik a későbbi olvasóknak azt, hogy bibliográfiai adatokkal kiegészítve tekinthessék az adott szöveget. Az olvasók az illető szöveget a kialakított szerkezeteken, keresztreferenciákon, megjegyzéseken keresztül, rendezett, de nemlineáris módon olvashatják el.

Egy hipertext három alapvető jellegzetességgel rendelkezik:

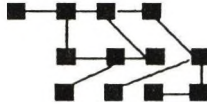
1. **csomópontok halmaza**, amely csomópontokon szöveg vagy grafika helyezkedhet el,
2. **szoftver módszertan**, amely lehetővé teszi a csomópontok egymáshoz való kapcsolását megfelelő szerkezetek szerint
3. **Interface eszközkészlet**, amely lehetővé teszi a fenti szerkezetek kialakítását a szövegekben megfelelő gombok segítségével, és könnyen manipulálhatóvá teszi a szövegeket és más médiákat különféle ablakok felhasználásával.

A **node** egy olyan darabja a hipertext hálózatnak, ahol szöveg vagy más média helyezkedik el. Egyes szoftver kialakításokban egy csomópont egy mondat, másokban egy csomóponthoz komplett szövegdarabok, fejezetek, vagy könyvek rendelhetők.

Különféle alkalmazásokban a **node** megjelenítése a képernyőn kártya (pl Apple Macintosh HyperCard), lap, ablak (Guide) formában történik.



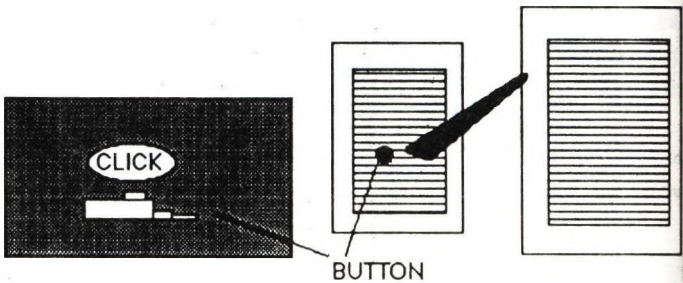
A **link** olyan eszköz, amely a hipertext rendszerekben alkalmas arra, hogy a fentiekben kialakított csomópontokat számítógéppel támogatott módon összekössön, ilyen módon lehetővé váljon a csomópontok hálózatának gyors és könnyű bejárása.



A **link** kialakítására néhány példa:

- a dokumentumok belső szervezése, ekkor ugyanazon dokumentum két részét kapcsolja össze egy szerkezettel;
- különböző dokumentumok közötti kapcsolat kialakítása;
- ún. pop-up ablakokon keresztül megjegyzések közölhetők a dokumentum olvasójával;
- egy dokumentum tartalomjegyzéke és indextáblázata is ilyen szerkezetnek tekinthető;
- egy dokumentum egy egységének tartalomjegyzéke.

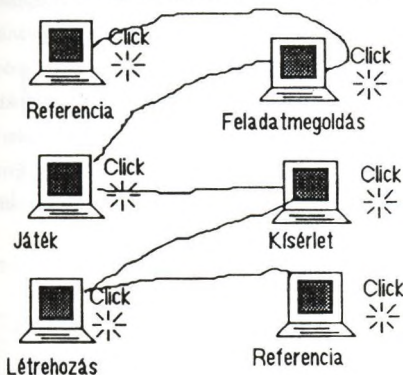
A **gombok** speciális helyek egy hipertextben vagy más médián, amelyek lehetővé teszik a felhasználónak azt, hogy az illető helyről egy szerkezet mentén eljussanak a link által kijelölt helyre ugyanazon, vagy más dokumentumban. A gomb megnyomása a rendszer kialakításától függően vagy egy egér nyomógombjának megnyomását vagy egy billentyű megnyomását jelenti. A gomb a kapcsolatoknak a felhasználó által látható megjelenési formájaként tekinthető.



Nézzük most meg azt, hogy mi különbözteti meg a hipertext rendszereket más szöveges adatbázis rendszerektől. Ahhoz ugyanis, hogy ilyen hipertext rendszert létre tudjunk hozni, emellett gondot kell fordítanunk. A fentiekben elmondottak mellett még három lényeges tulajdonságuk van a hipertext rendszereknek.

A gyors végigbongészés lehetősége

A hipertext rendszerek egyik legfontosabb előnye a szövegekben és dokumentumokban való gyors böngészés és navigálás lehetőségének biztosítása. Szükséges, hogy a felhasználó a szövegen illetőleg dokumentumon gyorsan és könnyen végighaladhasson. Ez azt jelenti, hogy a felhasználói interface kialakítása során biztosítani kell a rendszerben létrehozott kapcsolatokon való egyszerű végigjárás lehetőségét oly módon, hogy pl. elegendő legyen néhány nyomás az egér - mint manipulátor - gombján vagy néhány billentyű lenyomása a klaviatúrán.



Nemlineáris szövegek kezelése

Égyes hipertext szerzők a hipertextek legnagyobb előnyeként említik, hogy *lehetőség van a dokumentumok részleges rendezettségű tárolására és használatára, illetve lehetőség van nemlineáris dokumentumok kezelésére* is. Ez elvezet az ún. *szabad asszociációs gondolatok* tárolásának lehetőségéhez is. *Ilyen struktúra mellett lehetőség van arra, hogy pl egy irodalmi történetnek vagy más dokumentumnak többféle véget vagy továbbhaladási irányt is megfigeltethessünk.* Előnyös lehet ez a tulajdonság akkor, ha pl. egy kisebb csoport rendszertervezésre használja a hipertext rendszert, ilyen esetben számos terv-variáció tárolható a tervezés különböző szakaszaiban, egészen a döntésig, ahol általában csak egy lehetőség marad. Természetesen lehetőség van alternatív megoldások tárolására is a végső dokumentumban.

Két szerkezet-alakító opció biztosítása

Személyre szabható hipertext

Sok hipertext rendszer olyan, amelyben lehetőség van a hipertext adatbázisnak a személynél megvalósuló igényekhez való illesztésére.

Az általunk létrehozott hipertext, hipermédia fejlesztő rendszer is ilyen, lehetőséget nyújt a felhasználók számára az általuk szükségesnek látott változtatások végrehajtására a már létrehozott dokumentum részben az eredeti szerző alkotása, ugyanakkor többé vagy kevésbé magán viselheti használójának kézjegyét is, ez pedig óriási módon megnöveli a dokumentum felhasználhatóságának lehetőségeit. Rendszerünkben különféle pontok természetesen megtiltható illetőleg korlátozható – jelszavak, belépési kulcsok segítségével a felhasználó beavatkozási, pontosabban módosítási lehetősége. Erre elsősorban amiatt van szükség, hogy a dokumentum illetőleg hipertext rendszer az eredetileg kitűzött célnak megfelelő életciklusa során valamilyen szinten legalábbis meg tudjon felelni.

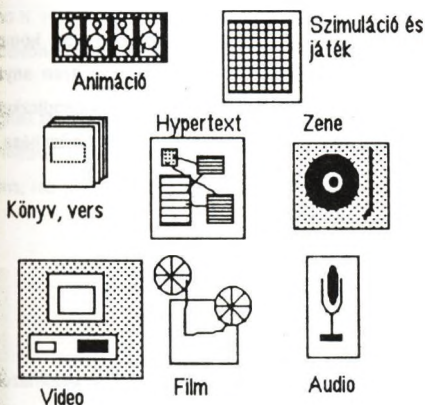
Csak olvasható hipertext

Az előbbiekből következik, hogy léteznek olyan hipertext alkalmazások, ahol lehetetlen az olvasók által történő utólagos beleírás. Ilyenek pl. a rendszerdokumentációk (pl. a programrendszer dokumentációja), oktatóprogramok, stb. Ezekben az esetekben a hipertext létrehozójának szándékaival ellentétes lenne az, hogy utólag az olvasó beleírhatson az anyagba. (Elképzelt pl., hogy egy operációs rendszer leírásának ilyen átírása komoly hibák forrása lehetne, stb.) Természetesen ilyen esetekben lehetőséget kell biztosítani a felhasználó számára arra, hogy esetleges megjegyzéseikkel, tapasztalataikkal a hipertext anyag kibocsátás felé jelezhessenek az újabb verzió kibocsátásának érdekében.

A fenti read-only jelleg értelemszerűen érvényes a CD-ROM alkalmazásokra, ezek esetében fizikailag is elképzelhetetlen az utólagos beleírás. Egyebekben egy CD-ROM létrehozásának költsége kb 2.00 \$, ezért a szokásos eljárás valóban az ilyen rendszerek esetében, hogy a változtatásokat időszakonként kibocsátott újabb CD-ROM verziókkal követik. (Amennyiben a fejlesztés során addig jut egy hipertext kidolgozója, akkor ilyen CD-ROM-ok létrehozásának számos jelentkező cég van Európában és az USA-ban egyaránt. Ez különösen a lexikális, könyvtári, lexikális, történelmi adatok tárolása esetén lehet nálunk méltánylandó szempont.)

1. HIPERMÉDIA

A hipermédia a **hipertext típusú alkalmazási kör olyan kiterjesztését** jelenti, amelyben a szöveg mellett **video, különféle illusztrációk, diagramok, hang, animáció és természetesen különféle számítógépes grafikák** is szerepelnek. Egy szerző lehetőséget kap arra, hogy szövegek, gráfok, diagramok, fotók, video, zene, hangeffektusok és más médiák között hipertext-szerű, számítógéppel segített kapcsolatokat hozzon létre. Ez az ember-gép kapcsolatok közt hihetetlen mértékben kitágítja.



Az ilyen típusú rendszerek kialakításának szoftver és hardver eszközei széles körben rendelkezésre állnak a különféle számítógéptípusokra (Apple Macintosh, IBM PC, workstation, stb.) már a hazai számítástechnikai piacon is.

4. Hipermédia rendszerek a gyakorlatban

Lássuk ezután, hogyan hozható létre és hogyan működik egy általunk elképzelt hipermédia rendszer.

Legyen a példánk egy termékről készített demo:

1. A potenciális felhasználói először egy HyperCard-ban elkészített demot nézhet meg, amelyben az illető termék legfontosabb jellemzői megtalálhatók, innen kiválaszthatja, hogy mi az a dolog, amit még részletesebben meg szeretne ismerni.

2. Amennyiben lehetőség van rá, akkor részletesebb demók is megtekinthetők videolemeze felvett film, animáció vagy diasorozat segítségével. (Erre jó példa a Hyperban elkészített *Bird Anatomy* című oktató anyag, amelyhez egy videodisc is csatlakoztatva a repülés technikája ugyanis természetesen így lényegesen jobban bemutatható.)

3. Az animációhoz hangban elmondott és zenével kísért információk is csatolhatók. Az információ nemcsak a számítógép diszkjén, hanem a videolemezen is tárolható. Természetesen lehetőség van arra, hogy a szöveg vagy zene minden eleméhez a dokumentumok bármely részéhez hozzákapcsolható legyen hipermedia linken keresztül.

4. Végezetül pedig korlátozott mértékben lehetőség van komplett filmek, fotók tárolására is ugyanazon a lemezen. Természetesen az ilyen anyagok tárolása igen nagy helyet igényel, de figyelembe véve a CD-lemezek és videodiszkek nagy tárolókapacitását (600 MByte és felett) nem kizárt ilyen komplett anyagok elkészítése sem.

5. A hipertext rendszerek dimenziói

Információszolgáltatás

Egy hipertext rendszert azzal is jellemezhetünk, hogy mekkora alkalmazói kört lát el információval. Ezzel a szemponttal egy hipertext rendszer kiszolgálhat információt egy felhasználót, egy csoportot, vagy akár egy felhasználói csoportot, valamint felhasználóknak egy egész hálózatát.

Módok

A mód azt az eljárást jelenti, ahogyan és amilyen feladatra egy hipertextet fel tudunk használni.

Ennek fajtái :

Használati módok : végigpörgetés, tréning, help, prezentáció, tanuló és analízis mód, referenciamód.

Authoring mód, amelyben létrehozuk a hipertext rendszert.

Szerkesztő mód, amelyben használjuk és módosítjuk a hipertextet.

Adminisztrációs mód.

A nálunk működő integrált multimédia fejlesztőrendszer mindezen módokat képes kiszolgálni információkkal és eszközökkel.

8. Irodalom

1. Campbell, B., Goodman J.M.: HAM: a general purpose HYPERTEXT abstract machine
Proc. of *HYPERTEXT'87*, Chapel Hill, NC. 1987, pp 45-96.
2. Engelbart, D.C.: Authorship provisions in augment.
IEEE 1984 *COMPCON Proceedings*, Springer, 1984., pp 75-102.
3. Engelbart, D.C., Hooper, K.: The augmentation system framework.
In: S. Ambron and K. Hooper (eds.), *Interactive Multimedia*.
Reading, MA: Addison-Wesley, 1988., pp 11-34.
4. Parsaye K., Chignell M., Khoshafian S., Wong H.: Intelligent Databases- Object-Oriented,
Deductive Hypermedia Technologies
John Wiley & Sons, Inc., 1989., 479 p.



Remszó Tibor - villamosmérnök

1976-tól dolgozik az MTA SZTAKI-ban, jelenleg tudományos főmunkatársként. 1978-ban szakmérnöki oklevelet, 1979-ben doktori címet szerzett a BME Villamosmérnöki Karán. 1989-ben adatbáziskezelés témakörben. E témában 1989-ben megvédte kandidátusi disszertációját.

Kutatási témái: adatbázis kezelés, ember-gép kapcsolati eszközök és technikák, vizuális adatkezelés, számítógépes hálózatok, heterogén rendszerek, hipertext, multimédia rendszerek.

INTERVALLUM ALAPÚ TUDÁSREPREZENTÁCIÓ

Csáki Zsuzsanna

Rendszer- és Irányításméleti Kutató Laboratórium, MTA SzTAKI

1518 Budapest, Pf. 63.

e-mail:h681csa@ella.hu

1. BEVEZETÉS

A mesterséges intelligencia gyakorlati alkalmazásainál az első és legfontosabb feladat megfelelő tudásreprezentáció kiválasztása. Ekkor a legfontosabb kérdés az, hogy létezik-e adott rendszerről elég elérhető szaktudás vagy pedig a hiányos ismeretek túlsúlya jellemző a rendszerre. Ez utóbbi esetben valamelyik hiányos tudáson alapuló reprezentációs módszer kiválasztani [3]. Ezek a módszerek a hiányos tudás alapvető megközelítésében az interpretációjában különböznek attól függően, hogy milyen típusú kérdésekre akarnak választ adni. Természetesen a kérdések típusai nem mindig egyértelműek. Az interpretáció alapvető tudásreprezentációs ág létezik:

- A hiányos tudást lefedik a lehetséges válaszok egy minimális halmazával. (pl. halmazelmélet, melynek speciális esetei az intervallum módszerek)
- Az adott kérdésre egzakt választ adnak, de ehhez egy hihetőségi mértéket is csatolnak (pl. valószínűségelmélet, hihetőségi mértékek a szakértői rendszerekben)

Az intervallum módszerek esetén a vizsgált rendszer változóinak értelmezési tartományát értékek készlete a valós számtest feletti intervallumok halmazának egy-egy részhalmaza közötti kapcsolatot pedig az ilyen típusú intervallumokon értelmezett művelet segítségével írják le. Ezek a műveletek egyrészt a hagyományos algebrai műveletek (pl. összeadás, szorzás, hatványozás, stb.), másrészt olyan speciális műveletek, melyek egy intervallumon valamilyen tulajdonságát mérik (pl. egy intervallum szélességét). E leírási mód előnye, hogy megtartja a megszokott (valós számokon alapuló) matematikai struktúrák leírási formáját, és könnyen értelmezhetővé válnak az intervallumokon alapuló formulák, és a bennük található bizonytalanság jól szemléltethető. Ugyanakkor e jó oldalnak komoly ára van, mivel algebrai tulajdonságok, mint pl. a disztributivitás, hiányoznak. Ezért gyakran valamilyen speciális közelítő algoritmusokat használnak, ill. speciális normálformákkal dolgoznak.

Mind e hátrányok ellenére az intervallum alapú reprezentációk újra és újra előtérbe kerülnek különböző típusú tudások leírásánál, tipikus példa erre a mesterséges intelligencia ún. kvalitatív szimulációs módszerénél alkalmazott tudásreprezentáció [6,7,13], vagy az irányítás elméletében "unknown, but bounded" típusú rendszerleírása [8]. Mindkét esetben a fizikai rendszer leírása nagyon szemléletes, viszont e modelleken alapuló optimális megoldó algoritmusok

következtetési módszerek kifejlesztése komoly problémákat okoz [2,4,11,12;8], komoly matematikai háttérrel igényel.

A jelen cikk első része bemutatja az intervallum alapú módszerek helyét a különböző bizonytalan tudásleíró módszerek között. Majd összefoglalja az intervallum módszerek jelölés módszerét és eddigi legfontosabb eredményeit, hangsúlyozva e módszerek főbb nehézségeit, valamint e hátrányok feloldásának lehetséges módjait. Végül a cikk bemutatja e módszer alkalmazását rendszerelméleti problémák megfogalmazására.

2. A BIZONYTALAN TUDÁS KÜLÖNBÖZŐ REPRESENTÁCIÓI

Mint már a bevezetésben említettük alapvetően kétféle megközelítési mód létezik a bizonytalan tudás leírására:

- A módszerek egy része a hiányos tudást a lehetséges válaszok egy minimális halmazával fedik le (pl. többértékű logikák; halmazelmélet, melynek speciális esetei az intervallum módszerek; kvalitatív következtetések)
- Másrésztük az adott kérdésre egzakt választ keresnek, de ehhez egy hihetőségi mértéket is csatolnak. (pl. hihetőségi mértékek a szakértői rendszerekben, valószínűségelmélet, Bayes alapú következtetés, hibaszámítási módszerek)

A fenti tudásrepresentációs ágakon belül a bizonytalan tudás megközelítési módjai elég különbözőek. A következőkben e különböző módszerek főbb tulajdonságait foglaljuk össze.

1. A többértékű logikákban [3] a fő feladat állítások igazságértékének megállapítása ismert igazságértékű állításokon alapuló következtetések (modus ponens, rezolúció, stb.) segítségével. Ezen belül az egyik legelterjedtebb technika a bizonytalan tudás 3-értékű ("igaz", "hamis", "nem-ismert") logikával történő leírása [10]. E módszer nagy hátránya az, hogy a bizonytalan ("nem-ismert") tudás gyorsan "továbböröklődik", ezáltal az adott rendszer leírásánál hirtelen megnő a csak bizonytalan tudással leírható részek nagysága.
2. Az általánosított halmazelmélet [3] alapjaiban elég közel áll a többértékű logikákhoz. Ebben a témakörben a legjelentősebb megközelítés az ún. *Fuzzy-elmélet*, melyet Zadeh 1965-ben fejlesztett ki [14]. Itt az ún. részvételi függvények játszanak fő szerepet, melyek egy elem egy adott tulajdonsághalmazhoz tartozásának mértékét adják meg.
3. Az intervallum módszerek esetén minden egyes információt egy zárt intervallum jellemez, és a következtetés nem más, mint intervallum alapú műveletek egymásutání végrehajtása. Itt a fő problémát az intervallum aritmetika gyengesége okozza (lásd 3. fejezet), ahol az eredmény intervallumot sokkal szélesebbnek kapjuk, mint amilyen az valójában [9].
4. A kvalitatív következtetést B. Kuipers, J. de Kleer, K. Forbus fejlesztette ki 1984-ben [13]. A módszer fő célkitűzése a környező világ egyfajta reprezentálása, úgy hogy a nem lényeges

- részinformációkat elhanyagolják, míg nagy súlyt helyeznek a lényegileg különböző részviselkedések megkülönböztetésére és legfontosabb tulajdonságaik megmagyarázására.
5. A szabályalapú szakértői rendszerekben alkalmazott *hihetőségi mértékek* [10] a következő eredményének "valószínűségét" jellemzik. Bizonytalansági mértékek adhatók a tudás tényeihez, ill. szabályaihoz, stb.
 6. A *valószínűségelmélet* általánosan (nemcsak a mesterséges intelligencia területén) alkalmazott módszer bizonytalan tudás leírására. Mivel e módszer matematikai alapjai messze legkidolgozottabbak, ez a megközelítés különböző kérdések megválaszolásához elég sok algoritmust tartalmaz.
 7. A *Bayes alapú következtetés* a valószínűségelmélet részeként a feltételes valószínűség kérdéskörével foglalkozik. E megközelítés előnye, hogy módszert ad bizonytalan tudás elérhető, ismert információkkal történő becslésére.
 8. A *hibaszámítási módszereket* leginkább a numerikus analízisben használják. Ekkor lehetséges válaszok halmazát egy pontos válasz, mint középpont, és a hiba, mint sugár segítségével adják meg. Habár ezt a módszert nem használják a mesterséges intelligenciában (de alkalmazzák speciális esetekben bizonyos intervallum határok szűkítésére - lásd 3. fejezet vége), probléma megközelítése nagyon hasonlít a előző három módszerre, mert itt is pontos választ kapunk, és egy számítási hibát, mely mint egyfajta bizonytalansági mérték értelmezhető.

3. AZ INTERVALLUM ARITMETIKA

A következőkben az intervallum aritmetika legfontosabb jelöléseinek és eredményeinek összefoglalása található [9] alapján. E fejezeten végig jelentsen $f: \mathbf{R} \rightarrow \mathbf{R}$ egy valós függvény. Először az intervallum függvények kétféle megközelítési módját vázoljuk fel.

1.a Definíció. Tegyük fel, hogy az f valós függvényt csak véges számú pontban ismerjük, de tudjuk róla, hogy monoton növekvő vagy monoton csökkenő függvény és $L = \{l_i; i = 0, 1, \dots, n\}$ jelentsen a valós számok egy növekvő sorozatát, ahol e függvényt ismertnek tételezzük fel. $f(L$ -en alapuló) *intervallum kiterjesztését*: F_L -t a következőképpen definiáljuk.

$$F_L([l_i, l_i]) = F_L(l_i) = f(l_i) \quad l_i \in L \quad \text{vagy}$$

$$F_L([l_i, l_j]) = [f(l_i), f(l_j)] \quad , \quad \text{monoton növekedést feltételezve, klb. fordítva} \quad (i < j)$$

ahol $l_i, l_j \in L, l_i < l_j$.

Megjegyzés. Legyen \mathcal{I} a zárt intervallumok halmaza, azaz:

$$\mathcal{I} = \{[a, b]; a, b \in \mathbf{R}, a \leq b\}$$

speciálisan: $\mathcal{I}_L = \{[l_i, l_j]; l_i, l_j \in L, l_i < l_j\}$

A fentiek alapján F_L egy intervallum függvény abban az értelemben, hogy \mathcal{I}_L -t \mathcal{I} -re képezi (jelölésben: $F_L: \mathcal{I}_L \rightarrow \mathcal{I}$).

1.b Definíció. Más úton is eljuthatunk az intervallum függvényekig. Ha a becsléni kívánt függvényt egy pontban sem ismerjük (vagy ez a tudás nagyon elenyésző), akkor célszerű valós, ún. boríték- függvényekkel alsó-felső korlátokat definiálni. Az f (boríték-függvényeken alapuló) intervallum kiterjesztését: $F_{I,u}$ -t a következőképpen definiáljuk.

$$F_{I,u} = [f_l, f_u], \text{ ahol}$$

$$F_{I,u} : \mathcal{I} \times \mathcal{I} \rightarrow \mathcal{I}, f_l, f_u : \mathcal{R} \times \mathcal{R} \rightarrow \mathcal{R} \text{ és } \forall x \in \mathcal{R}, f_l(x) \leq f_u(x) \quad (1b).$$

2. Definíció. Egy intervallum szélessége $w : \mathcal{I} \rightarrow \mathcal{R} : w([a, b]) = b - a$

Egy intervallum nagysága $|| : \mathcal{I} \rightarrow \mathcal{R} : ||[a, b]|| = \max(|a|, |b|)$

Intervallumok parciális rendezésén a következőt értjük:

$$[a, b] < [c, d] \text{ akkor és csak akkor } b < c$$

Két intervallum akkor és csak akkor egyenlő, ha a végpontjaik megegyeznek, azaz:

$$[a, b] = [c, d] \text{ akkor és csak akkor } a = c \text{ és } b = d.$$

A fentiekből kitűnik, hogy az intervallumokat kétféleképpen is értelmezhetjük. Egyrészt mint a valós számegegyenes egy részhalmazát, másrészt mint pontpárokat a valós számok halmazának önmagával vett direktszorzat halmazából. Az első esetben az intervallumokon végrehajtható algebrai műveleteket egy halmazzal definiáljuk, míg az utóbbi esetben a végpontokból számítjuk ki az új végpontokat. De ez utóbbi esetben előfordulhat, hogy bővebb eredmény halmazt kapunk (lásd lent 1. következmény) A következőkben a valós aritmetika intervallum kiterjesztését definiáljuk, mindkét típusú művelet megadás segítségével.

3. Definíció. Ha 'op' '+' vagy '-' vagy '*' vagy '/' jelek helyett állhat, akkor az aritmetikai műveleteket az 1. típusú definíció segítségével a következőképpen definiálhatjuk:

$$[a, b] \text{ op } [c, d] = \{x \text{ op } y \mid a \leq x \leq b, c \leq y \leq d\}$$

$$\text{kivéve: } [a, b] / [c, d] \text{ ha } 0 \in [c, d]$$

Ha feltesszük, hogy a fent említett 2. típusú művelet definiálásnál a végpontokat abszolút pontossággal ki tudjuk számítani, akkor az \mathcal{I} elemeire az ún. pontos intervallum aritmetikát vezettük be.

4. Definíció. Az intervallumokon értelmezett pontos intervallum aritmetika $+, -, *, / : \mathcal{I} \times \mathcal{I} \rightarrow \mathcal{I}$

$$[a, b] + [c, d] = [a + c, b + d]$$

$$[a, b] - [c, d] = [a - d, b - c]$$

$$[a, b] * [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$$

$$[a, b] / [c, d] = [a, b] * [1/d, 1/c]$$

1. Állítás. \mathcal{I} a fent definiált intervallum aritmetikával asszociatív, kommutatív, de nem disztributív.

Bizonyítás: Az asszociativitás és a kommutativitás nyilvánvalóan teljesül a valós szám értelmezett műveletek tulajdonságai miatt. A következőkben egy ellenpéldát mutatunk esetre, amikor ez nem teljesül.

$$[1, 2] * ([1, 2] - [1, 2]) = [1, 2] * ([-1, 1]) = [-2, 2]$$

$$[1, 2] * [1, 2] - [1, 2] * [1, 2] = [1, 4] - [1, 4] = [-3, 3]$$

De az ún. *szubdisztributivitás* itt is igaz marad. Ez a következőt jelenti: tegyük fel, hogy $I \in \mathcal{I}$, akkor: $I * J + I * K \supset I * (J + K)$

2. Állítás. A 0 és 1 valós számok az összeadás ill. a szorzás egységelemei lesznek additív és multiplikatív inverzek csak a *degenerált* intervallumokhoz, azaz a valós számok léteznek.

1. Tétel. Ha $F(X_1, X_2, \dots, X_n)$ egy *racióndús* kifejezés az X_1, X_2, \dots, X_n intervallum változóiban, azaz X_1, X_2, \dots, X_n -ből és konstans intervallumok véges halmazából intervallum aritmetikai műveletek véges kombinációjaként áll elő, akkor:

$$X'_1 \subset X_1, \dots, X'_n \subset X_n \Rightarrow F(X'_1, \dots, X'_n) \subset F(X_1, \dots, X_n)$$

minden olyan X_1, X_2, \dots, X_n halmazra, amelyre az F -beli intervallum aritmetikai műveletek értelmesek.

Bizonyítás: Ez $n=2$ -re, amikor csak egy aritmetikai művelet van, könnyen belátható. Ezt a tételt a műveletek száma szerinti indukcióval könnyen bizonyítható.

Megjegyzés: A fenti tétel azt jelenti, hogy az intervallum aritmetikai műveletek a részintervallumok tartalmazására, mint rendezésre nézve monotonok (röviden: monoton műveletek).

1. Következmény. Ha $f(x_1, \dots, x_n)$ egy valós racionális kifejezés, ahol minden x_i egyetlen egyszer fordul elő és csak az első hatványon, akkor a kifejezésnek megadott intervallum kifejezés, $F(X_1, \dots, X_n)$, intervallum végpontos számítás esetén is megadja pontos értékét, ha x_i -k az X_i tartományokból függetlenül, tetszőlegesen választhatók.

$$F(X_1, \dots, X_n) = \{f(x_1, \dots, x_n) \mid x_i \in X_i, i = 1, \dots, n\}$$

5. Definíció. Két intervallum távolsága $-d: \mathcal{I} \times \mathcal{I} \rightarrow \mathcal{R}$:

$$d([a, b], [c, d]) = \max(|a - c|, |b - d|)$$

1. Lemma. (\mathcal{I}, d) metrikus teret alkot, azaz ha $I, J, K \in \mathcal{I}$:

$$(1) d(I, J) = d(J, I)$$

$$(2) d(I, J) \geq 0, \text{ ahol egyenőlség akkor és csak akkor létezik, ha } I = J$$

$$(3) d(I, J) + d(J, K) \geq d(I, K), \text{ azaz a háromszög egyenlőtlenség teljesül.}$$

Bizonyítás: A $| \cdot |$ függvény tulajdonságából automatikusan következik.

Bonyolultabb feladatok megoldása esetén (pl. lineáris egyenletrendszer megoldása) általában az intervallum határok segítségével adják meg az algoritmusok az eredmény intervallum alsó ill. felső korlátját. Viszont ezekben az esetekben a disztributivitás hiánya miatt az egyenleteket olyan alakra kell hozni, hogy az 1. következmény feltételei teljesüljenek. Ha a tagokból kiemelhetők az azonos változók, és így megkaptuk, hogy minden intervallum változó csak egyszer és csak az első hatványon szerepel, akkor az így kapott eredmény intervallum nem lesz szélesebb a valóságos megoldásnál. Azonban gyakran nem tudjuk ezt megtenni. Ezért az általános megoldó algoritmusok valamilyen speciális normálformából (pl. központosított forma, átlagérték forma) indulnak ki, és az eredmény intervallum számításával párhuzamosan adnak hibabecslést. Az így kapott eredmény intervallumok tartakmazzák a valóságos megoldást (annál általában bővebbek), de így szűkebb halmazzt kaphatunk, mintha az adott feladatot az eredeti struktúrával oldanánk meg.

4. DINAMIKUS RENDSZEREK INTERVALLUM ALAPÚ LEÍRÁSA

A fentiek alapján vizsgáljuk meg, hogyan alkalmazható a fenti elmélet olyan rendszerek leírására, később irányítására, ahol a rendszer modell egyenleteit bizonytalan tudásrészekkel ismerjük csak. Ez tulajdonképpen a már fent említett kvalitatív szimuláció [6] vagy az "unknown but bounded" módszer [8] alkalmazását jelenti. Az előbbi esetben a bizonytalan részeket boríték-függvény alapú intervallum függvényekkel akarjuk becsülni az 1.b definíció alapján. Ugyanakkor diszkrét idejű rendszerekkel dolgozunk, mivel a lényeges rendszer viselkedéseket szeretnénk feltárni bizonytalan tudás mellett úgy, hogy a különböző diszkrét időpillanatokban különböző rendszerállapotokat - azaz az állapotváltozók különböző aktuális értékeit vagy változásuk eltérő irányait - vizsgáljuk. A következőkben Kuiperstől eltérően intervallum matematikai megközelítésben írjuk le a fenti tudásreprezentációs módszertant, azért hogy összehasonlíthassuk az "unknown but bounded" módszertannal.

6. Definíció. Diszkrét idejű rendszert, és a változóiban ill. a modellben fellépő korlátos bizonytalanságot feltételezve, a *folyamatot modellezzük* a következők szerint:

$$U: N \rightarrow \mathcal{V}^k, \quad \text{a rendszer bemenetek időfüggvénye (Jelölés: } U_1 := U(t))$$

$$X: N \rightarrow \mathcal{V}^m, \quad \text{az állapot változók időfüggvénye (Jelölés: } X_1 := X(t))$$

$$F_{1,u}: \mathcal{V}^k \times \mathcal{V}^m \rightarrow \mathcal{V}^m, \quad \text{melyet intervallum végpontokkal számolunk pl. } k=m=1 \text{ esetén:}$$

$$F([a, b], [c, d]) = [\min(f_l(a), f_l(b), f_l(c), f_l(d)), \\ \max(f_u(a), f_u(b), f_u(c), f_u(d))]$$

Ekkor a rendszer viselkedését a $t+1$. időpillanatban a következőképpen becsülhetjük:

$$X_{t+1} = F(U_t, X_t; \tau = 0 \dots t) \quad (2)$$

Abban az esetben, ha a rendszer viselkedését nemcsak becsülni, hanem befolyásolni szeretnénk, akkor a cél az, hogy olyan U_τ sorozatot válasszunk, hogy X_τ elég közel legyen X^* -hoz. Ezt az optimalizálási feladatot többféle célfüggvénnyel is megfogalmazhatjuk.

$$\begin{aligned} \text{a) } C_0 &= \min_{U_t} d(X_{t+1}, X^*) & \text{vagy} & & C_0 &= \min_{U_t} \left| (X_{t+1} - X^*) \right| \\ \text{b) } C_\infty &= \min_{U_t} \sum_{k=1}^{\infty} d(X_{t+k}, X^*) & \text{vagy} & & C_\infty &= \min_{U_t} \sum_{k=1}^{\infty} \left| (X_{t+k} - X^*) \right| \end{aligned}$$

A következőkben a fenti feladatot az "unknown but bounded" módszer intervallum megközelítésével fogalmazzuk meg. Itt adott egy hiányosan ismert problémaelem λ ($\lambda: N \rightarrow R^r$), ami jelenthet pl. egy dinamikus rendszer viselkedését pontosan jellemző időfüggvényt (a rendszer kimenő változóinak száma). Keressük λ egy függvényét ($S(\lambda)$ -t), ami értelmezhető mint a dinamikus rendszer paramétereinek értéke vagy mint egy diszkrét, csak lényeges rendszer viselkedési eltéréseket leíró függvény (ez utóbbi a kvalitatív szimuláció célja is). E utóbbi esetben $S(\lambda): N \rightarrow \mathcal{V}^m$ (ahol m az állapotváltozók száma). λ -t nem ismerjük pontosan, de feltételezhetjük, hogy ismerjük azt a K halmazt, amely őt tartalmazza (azaz pl. a rendszer struktúrája vagy foka ismert vagy feltételezett), és ezenkívül a rendszer bizonyos változóit mér tudjuk, azaz létezik $F(\lambda) \in R^r$ függvény (ahol r a mért változók száma). De az aktuális mérési eredményeit csak bizonyos hibával ($\rho(F) \in \mathcal{V}^r$) kapjuk meg, amely nem ismert, de korlátozott ("unknown, but bounded").

7. Definíció. A dinamikus rendszerek korlátos hibával megközelített leírása:

$$y = F(\lambda) + \rho, \quad \lambda \in K, \quad F: N \rightarrow R^r, \quad y \in \mathcal{V}, \quad \|\rho\| \leq \epsilon, \quad (\epsilon > 0). \quad (3)$$

Ekkor a becslési feladat nem más mint egy $\Phi: \mathcal{V}^r \rightarrow \mathcal{V}^m$ becslő függvény megkeresése úgy, hogy $\Phi(y)$ a mért adatok alapján a lehető legjobban megközelítse valamilyen intervallum normájában $S(\lambda)$ -t.

A fentiek alapján a rendszer viselkedésének becslése a $t+1$. időpillanatban a következőképpen fogalmazható meg:

$$S(\lambda) = \lambda_{t+1}, \quad \Phi(y) = \Phi(F_\tau, \rho_\tau, K, \tau = 1, \dots, t) \quad (4)$$

A (2) ill (4) formulák összevetéséből kitűnik a két módszer közti lényeges különbség az azonosság. Az első esetben bemeneti adatok bizonytalanságát (U) egy minimális lefedési intervallummal ábrázoltuk, míg a 2. esetben egy hibamértéket adtunk (ρ) egy pontos számítás eredményéhez ($F(\lambda)$). A rendszer belső állapotainak (X ill $S(\lambda)$) reprezentálása mindkét esetben intervallum lefedéssel történt.

5. ÖSSZEFOGLALÁS

A cikk összefoglalta a mostanában használt intervallum alapú tudásreprezentációk helyét a bizonytalan tudás reprezentálásának lehetséges módszerei között, valamint az intervallum alapú módszerek legalapvetőbb problémáit. Ezen túlmenően a cikk bemutatta e módszer alkalmazásait dinamikus rendszerek intervallum alapú leírására. Az egyes irányok tárgyalása korántsem lejelezett, a cikk leginkább csak felvillantja a lehetőségeket, problémákat, de a cikk korlátai túl szűkek egy részletesebb, igazán alapos tárgyaláshoz.

6. KÖSZÖNETNYILVÁNÍTÁS

A szerző ezúton is megköszöni a Magyar Hitelbank "Magyar Tudományért" alapítványának támogatását. A fenti kutatást ezenfelül az Országos Tudományos Kutatási Alap is támogatta a 2577-es szerződésen keresztül. A szerző mindezen felül köszönetet mond Hangos Katalinnak támogatásáért és értékes megjegyzéseiért.

7. HIVATKOZÁSOK

- 1 Csáki, Zs. Bayesian identification as Reasoning about incomplete knowledge of physical systems. Accepted for the ECAI'92 Conf., Wien, 1992.
- 2 Csáki, Zs., Hangos, K.M., Hallager, L. and Jørgensen, S.B. Qualitative Simulation for Start up a Distillation Column. *Prep. of the Workshop on Computer Software Structures Integrating AI/KBS Systems in Process Control*, Bergen (Norway), 1991, 81-87
- 3 Goodman, I.R. and Nguyen, H.T. Uncertainty Models for Knowledge-Based Systems. *Elsevier Science Publisher*, The Netherlands, 1985
- 4 Hangos, K.M. and Csáki, Zs. Qualitative Simulation in the Limit. To appear in *Artificial Intelligence in Engineering*, 1992
- 5 Henrici, P. Essentials of Numerical Analysis with pocket Calculator Demonstrations. *John Wiley & Sons*, 1982
- 6 Kuipers, B. Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge. *Automatica*, 1989, 25, 571-585
- 7 Kuipers, B. Higher order derivative constraints in qualitative simulation. *Artificial Intelligence*, 1991, 51, 343-379
- 8 Milanese, M. and Vicino, A. Optimal Estimation Theory for Dynamic Systems with Set Membership Uncertainty: An Overview. *Automatica*, 1991, 27, 997-1009
- 9 Moore, R.E. Interval Analysis. *Prentice-Hall Series in Automatic Computation*, 1966
- 10 Schildt, H. Artificial Intelligence using C. *Osborne McGraw Hill*, California, 1987
- 11 Struss, P. Mathematical Aspects of Qualitative Reasoning. *Artificial Intelligence in Engineering*, 1988, 3, 156-169
- 12 Struss, P. Mathematical Aspects of Qualitative Reasoning. Part two: Differential Equations. *Siemens Report INF2 ARM-7-88*, 1988, 1-36
- 13 Weld, D.S. and de Kleer, J. (eds.). Readings in Qualitative Reasoning about Physical Systems. *The Morgan Kaufmann series in Representation and Reasoning*, 1990, 1-720.
- 14 Zadeh, L.A. Fuzzy sets. *Inf. and Control*, 8, 1965, 338-353



Csáki Zsuzsanna - programtervező matematikus. 1989-től dolgozik az MTA SZTAKI-ban tudományos segédmunkatársként. Kutatási területe mesterséges intelligencia, ezen belül az az irányításelmélettel közös határterület. 1990-ben 4 hónapot ösztöndíjasként a Dániai Szaki Egyetem Kémiai Technológiai Intézetében töltött, ahol egy kvalitatív szimuláció alapuló operátori tanácsadó rendszer prototípusát készítette el.

A DODS döntésobjektíváló és dokumentáló rendszer

(Pekker Mária - Sály Zoltán (ÁSZSZ) - Pósze Lajos (Szerencsejáték Rt.)

A DODS (DöntésObjektíváló és Dokumentáló Szoftver; angolul: Decision Objectivating and Documentating System) elnevezésű, a döntési folyamat korszerűsítését szolgáló döntésobjektíváló és -dokumentáló szoftvert magában foglaló eljárásunkat többéves tapasztalatok alapján dolgoztuk ki s a múlt év végén sikeresen alkalmaztuk egy országos jelentőségű cég technológiai rekonstrukciójára vonatkozó pályázatok kiértékelésére.

Eszközünket jellemzően a DODS szoftver - egyéni vagy csoportos - versenyztető jellegű használatára terveztük, ahol versenyző pályázatok, "alternatívák" közül a "nyerő" kiválasztása a cél.

Gyakorlatunk során szerzett tapasztalataink azt mutatják, hogy a fenti döntési eljárás gyorsabban elvezet a végeredményhez, ha már a pályázati kiírásba is beépítjük az előzetesen kialakított döntési modellt.

Magyarországon, sajnos, a döntési eljárásban alkalmazott módszerek még meglehetősen elmaradottak. A fent említett cég szóbanforgó döntésénél alkalmazott módszere (ti. hogy megbízta az ÁSZSZ-t a döntési eljárás levezetésével) azonban nagy előrelépést jelent és példaértékűnek tekinthető, s remélhetően megindít egy áttörési folyamatot.

Eljárásunk alkalmazásának főbb előnyei:

A döntést objektíválja. Mint már említettük, a döntési eljárás során a DODS szoftverrel végeredményként a pályázatok rangsorát állítjuk elő, előzetesen rögzített kritériumok szerint. Az egyes kritériumok (szempontok) szerinti kiértékelésnél esetleg bekövetkezhet valamely alternatíva kizárása, az adott szempontra vonatkozóan előírt ún. kizáró feltétel teljesülése miatt. Ezeket a kizárásokat is feltüntetjük a fenti rangsor részeként.

- A döntési eljárás elemeit és azok jellemzőit és a kiértékelés összesített és részletes végeredményét a rendszerből bármikor visszadokumentálhatóvá, s ezáltal a kiértékelési eljárást (a döntés hátterét) jól ellenőrizhetővé teszi.

A döntési eljárás elemei (szereplői): az alternatívák, a szakértők és a szempontok.

A szempontok jellemzői: azok struktúrája, egymáshoz viszonyított súlya, magyarázata, a hozzájuk tartozó minősítő skálák és az azokat egységes pontrendszerre konvertáló hasznossági (utility) függvények.

A szakértők jellemzői: szöveges bemutatásuk (pl. szakmai életrajz) és végszempontonkénti kompetenciáik (táblázatosan).

Az alternatívákat is jellemezzük (szöveges bemutatás).

A döntés modellezésének lényege, hogy az egyes alternatívákra az egyes szempontoknál a skálák szerint adott minősítések eredményeit összesíteni kell, s ezért a minősítési skálaértékeket szempontonként egységes pontrendszerre kell átalakítani. Ezeket a pontszámokat összesíti - a szempontsúlyok és szakértői illetékességek figyelembe vételével - a DODS szoftver a kiértékelési eljárás során. (A szakértők természetesen az alternatívák alapos áttanulmányozása, megismerése után véglegesítik a súlyokat és a skálákat, majd végzik el a minősítést.)

Így válik az "alma" összehasonlíthatóvá a "körtevel", azaz így válnak a teljesen különböző minősítő értékek (pl. a numerikus a szöveges típusúval, a vételár a műszaki megoldások színvonalával) összemérhetővé és végső soron összesíthetővé.

A döntési eljárás alkalmazása előnyös mindazokban az esetekben, amikor

- nagyobb horderejű döntésről (pl. beruházási tender kiválasztása) van szó;
- sok pályázat (pályázó) van;
- a pályázatok terjedelmesek és bonyolult felépítésűek (még ha pályázati kiírásnak megfelelően struktúráltak is);
- s a döntéshozónak ráadásul döntését komolyan, megtámadhatatlanul meg kell indokolnia, le kell dokumentálnia mind pályázói felé (akik sok munkát

fektettek a pályázatok összeállításába és az esetleges előzetes tárgyalásokba, s esély van arra, hogy megtámadnának egy nem kellően megalapozott, az elfogultság látszatától nem mentes döntést), mind felesleges szervei felé (akik csak akkor hagyják jóvá a felterjesztett döntést, ha a sikerre bizonyos garanciákat már előre látnak).

A DODS szoftvernek azonban az előbbi, versenyeztető jellegű alkalmazásokon túlmenően **más alkalmazási területei** is vannak:

- Egy ötlet, előterjesztés, javaslat, koncepció elfogadása-elvetése (pl. elvetés egy vagy több kizáró feltétel teljesülése alapján).
- Jelenség, folyamat, eljárás figyelése, a kritikus időpont, szituáció bekövetkeztének jelzése.
- Egy adott eseménycsoportból, jelenségegyüttesből, pályázói körből előzetes szűrés végzése a további elemzés előtt (ez tk. egy prekvalifikációs fázis beépítése a döntési folyamatba).
- Több kedvezményezett között egy adott (korlátozott) erőforrás elosztása (ez több lépcsős eljárás, itt nem részletezzük). (Példa lehet erre egy nagyobb hitelkeret szétosztása hitelképesnek bizonyult pályázók között. Előzetesen a hitelképesség-vizsgálat is elvégezhető az eljárással és a szoftverrel.)
- Az előnyöst mérő szempontokból, illetve a hátrányost mérőkből külön szempontrendszer kialakításával egy **előny-hátrány mérleg** készíthető egy vagy több alternatívára - ami nem más, mint egy haszon-ráfordítás (benefit-cost) elemzés. (Példa lehet erre egy ilyen jellegű elemzés a bős-nagymarosi vízlépcsőről és más alternatív energia-termelő beruházási lehetőségekről.)

Tehát az eljárás alkalmas többek között:

- pályázó személyek (pl. felvételizők, szakértők) kiválasztására,
- telepítési alternatívák kiválasztására,
- környezeti hatások összegző értékelésének dokumentált objektíválására.

Még egyszer röviden felsorolva a döntésobjektíváló és dokumentáló eszköz alkalmazásának előnyei a következők:

Az eljárás, a kialakított döntési modell, illetve a döntés végeredménye:

- Struktúrált, áttekinthető.
- Dokumentált, ellenőrizhető.
- Helyzetváltozástól függően aktualizálható.
- Munkát takarít meg - ezzel az átfutási idő csökkenthető.
- Az eljárás modellezhető - hasonló esetekre a modell ismételt felhasználható (mint egy "tudásbázissal" feltöltött szakértői rendszer).

Az ilyen döntéseket objektíváló és dokumentáló korszerű eljárás és eszköz alkalmazása az igazgatás magasabb szintjein egyre fontosabbá válik, mivel a döntések megfelelő, bármikor reprodukálható és ellenőrizhető dokumentálása, objektivitásának, elfogulatlanságának bizonyítása csak megfelelő eszközökkel lehetséges.

Ha tehát **nehéz döntési helyzetbe** kerülnek, a következő lehetőségeket tudjuk felajánlani Önöknek mint későbbi potenciális felhasználóinknak:

1. **A döntési eljárás teljes levezetése** már a "kezdet kezdetétől", a pályázati kiírás előtt történő bekapcsolódással. Ez azt jelenti, hogy a kiértékelési szempontrendszer (döntési modellt) beépítjük már pályázati kiírásba is.

A döntési modell helyes kialakítása az eljárás sikeres alkalmazásának sarkalatos pontja, s ha ezt már a pályázati kiírásba is beépítjük, ez azzal az előnnyel jár, hogy amennyiben a szakértői zsüri úgy dönt, a kiírásnak nem megfelelő felépítésű pályázatok egy előminősítési (prekvalifikációs) fázisban kizárhatók a további értékelésből, ezáltal egy könnyebben értékelhető "shortlist" alakul ki; a teljeskörű minősítéshez szükséges **pótdatkérések** igényét azonban mindenképpen minimálisra csökkenti.

2. Elvégezhetjük csak a beérkezett pályázatok kiértékelésének levezetését is.

A döntéshozó általában a fenti két lehetőség vonatkozásában egyaránt igen nehéz helyzetben van, mert csak az általa képviselt cég céljait, pénzügyi helyzetét és lehetőségeit érintő kérdésekben illetékes. Nem tudja azonban összemérni egymással pl. egy számítástechnikai tenderrel kapcsolatban az ajánlott műszaki megoldások (pl. hardver, szoftver, adatátviteli háttérfeltételek) előnyeit-hátrányait, a felkínált szolgáltatások színvonalát stb., továbbá sem ideje, sem olyan tanácsadója, szakértője nincs, aki beleásná magát több ezer oldalnyi (ráadásul gyakran idegennyelvű) anyagba vagy a pályázók által rendelkezésére bocsátott berendezés-mintadarabok tesztelésébe (esetleg referenciahelyek megtekintésébe).

Kívánságra és előzetesen az anyag előfeldolgozását is elvállaljuk külön szolgáltatásként (ez ugyanis nem oldható meg a módszerrel, illetve a szoftverrel; gyakorlatilag elsősorban a pályázatok leltárba vételét, lefordítását, lektorálását és korrektúráját, a fordítások végső formába öntését (nyomtatás és bekötés), illetve a minősítéshez bizonyos önmagukért beszélő adatok ("automatizmus") kigyűjtését, bizonyos adatok (pl.a vételár) lenormálását (a vételár esetében egy határnap árfolyamán és bizonyos tényezők figyelembe vételével) stb.) jelenti.

A berendezés-tesztelésbe és a referenciahelyek megtekintésébe (ezeket formálisan az előfeldolgozás körébe soroltuk, valójában azonban történhetnek a döntési eljárás levezetésével párhuzamosan, sőt utána is) való bevonásunkra kevésbé s inkább csak akkor számítunk, ha a beruházás a megbízó számára többé-kevésbé idegen (pl. az egészségügyben számítástechnikai jellegű beruházás esetén; diagnosztikai berendezéseik tesztelését nyilvánvalóan nem ránk fogják bízni).

A döntési eljárásban részt vevő szakértőket a Megrendelő (is) javasolja, de nekik már mi adjuk a megbízást, s a későbbiekben mi fizetjük ki a díjaikat s végezzük el az ezzel kapcsolatos adminisztrációt.

A szakértők általában az illetékes régi területükön szakértői igazolvánnyal rendelkező független külső szakemberek, legalább

egyikük azonban célszerűen a Megrendelő képviselője (gondoljunk a megvalósítás pénzügyi, finanszírozási módozataira).

Az 1. és 2. lehetőség vonatkozásában a döntési eljárás általunk történő teljes levezetése a döntési modell felállításától a szakértői minősítésen keresztül a teljes ledokumentálásig tart. A fenti feladatok komplexitása függvényében a fenti szolgáltatások díját szakértői ülésenként határoztuk meg,

illetve egy pályázati kiértékelésnél (a pályázati anyagok előfeldolgozása nélkül) a tervezett beruházás összértékének, a beruházás méretével ellenkező arányban csökkenően 10-0,5 %-ában állapítjuk meg.

3. **Betanítással eladjuk a szoftvert és a módszert belső (vállalati, intézményi) alkalmazásra.**

Ennek díja a várható használat intenzitásától, illetve az intézmény, vállalat méretétől függően változó.

4. **Betanítással eladjuk a szoftvert és a módszert a fentiekén túlmenő, harmadik félnek végezhető további alkalmazásokra is.**

Ezzel - az eljárás know-how-jának és az azt támogató szoftvernek az eladásával - fejlett, magas színvonalú eszközt adunk a felhasználó kezébe akár további saját, akár külső alkalmazásokra.

A 3. és 4. lehetőség vonatkozásában a betanítás közös, a felhasználó által megadott területekre vonatkozó esettanulmányok készítését is magában foglalja.

5. Természetesen maga a szoftver, pontosabban annak betanult változata is megvásárolható.

- 5.1. A program egy gépre hajlékony lemezen, felhasználói kézikönyv kíséretében vagy:
- 5.2. A program egy gépre üzembehelyezéssel hajlékony lemezen, felhasználói kézikönyv kíséretében, helyszíni installálással, bemutatással, betanítással ,(esettanulmányok, mintamodellek készítése).



Pekker Mária

az Államigazgatási Számítógépes Szolgálatnál dolgozik teamvezetőként, jelenleg többek között a döntéstámogató eszközök fejlesztése és alkalmazása területén. A döntési eljárás levezetésében, és a benne foglalt szoftver alkalmazásában az utóbbi időben sikerült komoly gyakorlatot szereznie.

Adatkonverziós technológiák térinformatikai rendszerek fejlesztéséhez

Dr. Srajber Benedek
Dr. Bakó András

Államigazgatási Számítógépes Szolgálat
Budapesti Műszaki Egyetem

1. Áttekintés

Napjainkban egyre elterjedtebbé válnak azok a számítógépes alkalmazások, amelyek térképi, illetve térbeli objektumokhoz kötött információk feldolgozását jelentik. Ezekhez a CAD/CAM rendszerek valamelyikét használják. Nagyobb földrajzi, közigazgatási vagy ipari területhez kapcsolódó (integrált) térinformatikai rendszerek egyre inkább GIS (Geographic Information System), LIS (Land Information System) vagy ezekhez hasonló szoftvereket használnak.

A térinformatikai rendszerek alapját, s egyben azok fejlesztésének legszűkebb hazai keresztmetszetét a korszerű adatgyűjtő (adatkonverziós) technológiák hiánya vagy azok nehéz hozzáférhetősége (magas költség) képezi. A CAD/CAM rendszerek eléggé elterjedtek (leginkább az AUTOCAD), de az azokhoz szükséges digitalizált térképi vagy rajzi adatállományok csak korlátozottan állnak rendelkezésre. A manuális digitalizálás vagy szkenneléssel párosuló, "olcsó" megoldások nem vezetnek célra. A komplex adatkonverziós technológiák telepítése és szolgáltatási rendszerként való bevezetése teremtheti meg a térinformatikai rendszerek (területi műszaki adatbázisok, átnézeti térképre alapozott ipari-műszaki információs rendszerek, közműnyilvántartások, stb.) fejlesztésének induló feltételeit. Ezekről a technológiákról, ezek szoftver és hardver eszközeiről kívánok áttekintést adni - a teljesség igénye nélkül.

Az 1980-as években bevezetett szkennelési technológiák a digitális adatgyűjtés (térképi, műszaki és képi) teljes automatizálása felé fordították a figyelmet. Az automatikus raszter-vektor konverzió és karakter illetve szimbólum felismerés első próbálkozásai nyomban nem vezettek a várt, gyors eredményekre. Egyszerű nyilvánvalóbbá vált az utólagos elemzések és korrekció szükségessége, amelyeket a különböző GIS rendszerek igényeltek.

A szkennelési technológiát újabb technológiákkal kellett kiegészíteni, hogy sikeres megoldást érjünk el. Ilyenek voltak:

- optikai diszk technológia a nagytömegű szürkeskálájú képadatok tárolására
- nagysebességű processzorok (RISC)
- nagyfelbontású interaktív raszter és vektor grafikus munkaállomások bevezetése lokális hálózatokkal
- alakfelismerési algoritmusok finomítása (mesterséges intelligencia technikák)

A fenti technológiák célnak megfelelő működtetése egy integrált rendszerben képzelhető el. Ezt kísérelte meg számos nagynevű cég, például SMARTSCAN, INTERGRAPH, LASER SCAN, PAFEC, stb. Mindegyik sajátosan a maga módján. Egyöntetű, minden GIS számára használható megoldás így nem született. Vannak jó és kevésbé jó rendszerek, de azok megkülönböztetése igencsak megnehezíti a felhasználó helyzetét, ha választani kíván közülük.

2. Adatkonverziós technológiai fogalmak

Adatkonverziós technológián egy olyan összefüggő és konzisztens adathalmaz struktúrázását és digitalizálását értjük, amelynek célja grafikus és adatbázis információk előállítása a különböző GIS rendszerek igényeinek megfelelő formában.

A térképek, képek, rajzok és dokumentumok digitalizálása manuálisan (digitalizáló tábla) is történhet, de a térinformatikai rendszerek mai követelményei (automatizálás) a szkenneléstől történő digitalizálást (szkennelés) nélkülözhetetlenné teszik. A szkennelés eredménye egy "raszter kép" (digitális információ), amely alapját képezi az automatikus konverziós folyamatnak. A raszter kép pontok vagy képelemek (pixels) tömbjeinek egyesítéséből áll elő. (Egy 60x60 cm-es térkép vagy műszaki rajz raszter képe - 250 pont/inch felbontás esetén - 36 Mb adatból áll.)

Raszter-vektor konverzió az a folyamat, amely valamely geometriai alakzatot azonosító raszter pontokat (vagy képelemeket) elválasztja a háttértől és megfelellet azoknak egy vektor-reprezentációt. Ez a vektorizálás lehetővé teszi a grafikus adatok kezelését, editálását, tulajdonságok hozzárendelését és tárolását s egyben alapeleme a GIS-nek is. A folyamat lehet teljesen vagy részben automatizált (utóbbi interaktív, képernyőről történő vektorizálás).

Interaktív vektor-raszter editálás egy olyan nagyfelbontású színes grafikus munkahely (szoftver és hardver) használatát jelenti, amely lehetővé teszi a felhasználónak, hogy mintegy raszter kép feletti fedőréteget (overlay-t) könnyedén áttekintse és kezelje a vektoros grafikát. Ez a kulcsa minden konverziós folyamat automatizálás irányába történő terelésének, beleértve a vektorizálást, a tulajdonság-hozzárendelést, határkorrekciót, szélek illesztését és adathitelesítést.

GIS adatbázisokban egy olyan struktúrát, amely általában tetszőleges számi egyedi sajátosságokkal rendelkező, logikailag összetartozó információt tartalmaz, rétegnek (layer-nek) nevezünk.

3. Az adatkonverziós technológia hardver komponensei

- a) Nagyfelbontású sötét színskálájú szkennerek
Főbb összehasonlítási paraméterek: méret (térkép, dokumentum, rajz), felbontás, sebesség, beolvasási idő, pontosság, ár.
- b) Optikai diszkek
Kapacitás: néhány gigabájtól (GB) 50 GB-ig is terjedhet alkalmazási feladattól függően.
- c) Interaktív vektor-raszter munkaállomás új display rendszerrel (5-16 Mb belső memória a raszter képek és vektorizálás hatékony kezelésére).
- d) Nagysebességű (RISC) processzorok a raszter képek elemzése és feldolgozása GB nagyságrendű halmazokon és maga a vektorizálás

(matematikai transzformációk) 10-100-szor nagyobb sebességet kíván, mint amit az általában használt CPU-k szolgáltatni képesek.

e) Lokális hálózati rendszer

Egy nagyvolumenű adatkonverziós feldolgozás funkciói munkahelyek között osztódnak el és naponta több GB-nyi gyors adatátvitelt igényelnek.

4. Az adatkonverziós technológia szoftver komponensei

Nagy hatékonyság érhető el az alábbi szoftver környezettel (RISC processzoros munkaállomással):

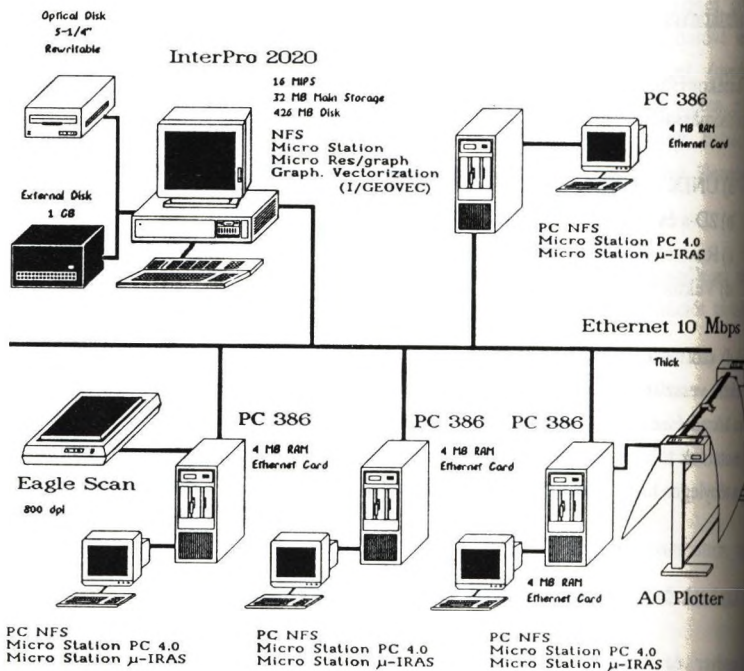
- a) UNIX operációs rendszer
- b) 2D-s és 3D-s CAD szoftver
- c) Raszter kezelő és editáló szoftver
- d) Vektorizáló szoftver

Az a)-d) szoftverek összehasonlítása, ha a nemzetközi élvonalba tartozó cégek termékeit vesszük figyelembe, az elérhető leírásokból nehezen végezhető el. Durva közelítéssel ugyanazokat a követelményeket látszanak kielégíteni. (Az árak sem mutatnak többet.) Az általános leírásokon és "demo"-kon túlmutató anyagok vagy a tényleges használat alapján állapíthatók meg a minőségi különbségek.

5. Egy konkrét adatkonverziós technológiai rendszer összeállítása.

A hardver és szoftver eszközök megválasztásánál a megoldandó feladat sajátosságait és a komponensek összehasonlító elemzését vettük alapul. A végeredmény egy Intergraph konfiguráció, amelyet a mellékelt ábra szemléltet.

INTERGRAPH



Dr. Srajber Benedek

1963-től az MTA Számítástechnikai Központnak munkatársa és operációkutatással foglalkozik. 1972-től a SOTE Számítástechnikai Központjának vezetője, oktató és kutató. 1977-től az ÁSZSZ-nél alkalmazásfejlesztési vezetőjeként dolgozik. Az utóbbi években a tevékenységének fő területe a számítógépes grafika felé fordult. A közigazgatási és iparvállalati informatikai alkalmazásfejlesztések a legfontosabb alkalmazási területek számára.

Korszerű Fulltext rendszer UNIX alatt

Herczeg István (ÁSzSz) - Mag. Peter Graf (Wien):

Egy UNIX alatti működésre átdolgozott irodai ügyviteli, illetőleg az iratarchiválás-visszakeresési folyamatba jól illeszkedő fulltext rendszert ismertetünk röviden a következőkben.

Jelentősége:

A magyarországi hivatalok többségében eddig már bevezetett ügyirat kezelést korszerűsítő műveletek (szövegszerkesztés, sokszorosítás, telefax) után ma aktuálisan korszerűsíthető az irat(szöveg) kezelés is, mégpedig annak a következő három tényezője:

- szempontok szerinti szabad keresés
- a szöveg elemzés, ugyancsak szempontok szerint és
- az archiválás, helytakarékosan, jól hozzáférhetően.

Az archiválás kulskérdése ma már nem a letárolás, a WORM diszkek elterjedése miatt, de továbbra is megoldatlan a hatékony, gyors, tartalom szerinti visszakeresés. Ezt a célt kell, hogy szolgálja az újabb fulltext fejlesztése.

A megvalósításnak többféle útja-módszere ismert:

- deskriptoros, kulcsszavas módszer
- editor-jellegű, tartalom szerinti keresés (karakter-összehasonlító módszer)
- keresési lehetőség invertált listák (szójegyzék indexálás szerint).

Előnyök-hátrányok:

- deskriptoros változat:
 - előnye a gyors keresés, de a deskriptorozás
 - * időigényes
 - * magasabb szaktudást igényel és
 - * mindenképpen maradnak hézagok
- tartalom szerinti keresés karakter összehasonlítással:
 - * minden szóra, szövegrészre kereshet
 - * gyors a betöltés, de
 - * nagyobb állományoknál nagy a keresési időigény
- invertált listás szövegindexelés

előzőkhöz képest több a tárolási idő, de jellemzői

- * a rövid keresési idők
- * bármilyen szövegösszefüggésre lehet keresni
- * viszonylag kis indexállományok

A megvalósult invertált listákkal dolgozó rendszer

- 3-szintű felépítéssel rendelkezik, az ügyviteli sajátosságoknak megfelelően adatbázis, dokumentum(irat), mező,
- ajánlott irányelvek
 - * 1 dokumentum: max. 3-5 oldal legyen
 - * legalább egy cím feltüntetése célszerű az első mezőben
- keresési opciók
 - * szabad szövegű
 - * mező szerinti
 - * csonkoltra, hasonlóra keresés
 - * logikai kapcsolatok (ÉS, VAGY, NEM, egyenlő, kisebb, nagyobb)
- szöveg input-output ...
 - * Scanner-OCR
 - * többféle szövegszerkeztő
 - * adatbázis kezelők
 - * táblázat kezelők

- saját szövegszerkesztő
- tetszés szerinti karakterkészlet
- irodai segéd funkciók
 - * körlevél
 - * formulák
 - * etikett
 - * lista készítés
- kommunikáció más szoftverekkel
- három védelmi szint (adatbázis létesítés, ezen belül keresés, módosítás)
- kötőszavak letilthatósága
- deskriptorok alkalmazása
- szinoníma szótár

Referenciák az eddigi DOS-változatra:

- Német nyelvterületen 1988 óta több, mint 180
- Magyarországon a bevezetés időpontja 1991. május, eddig alkalmazva 5, bevezetés alatt több 10 helyen

A fulltext rendszer kapcsolódása a UNIX-hoz.

Mivel a UNIX eredetileg szövegfeldolgozásra készült, amit alátámaszt az, hogy a legtöbb szabványos eszköz szöveges jellegű file-ok kezelésére íródott, egy ilyen rendszerbe jól integrálható egy fulltext eszköz. Az ismétlődő feladatok végrehajtására kisebb parancsnyelv állhat rendelkezésre, mint az **awk**-nál. Jól használható parancsláncba szervezve a meglévő eszközök és lehetőségek felhasználásával (**sed**, **awk**, **nroff**, **troff**, input-output átirányítás). Interaktív megjelenésével segítheti a felhasználót a kényelmes kezelés útján. Implementálása a szabványokhoz igazodva történik, s így hordozhatósága a kis mértékben eltérő UNIX rendszerek közt egyszerű feladattá válik. Az opcionális Xwindows interface kihasználása rendkívül hatékonyá teszi a programot a megfelelő környezetben.



Herceg István

INFRASTRUKTÚRA ("SZ VI.")

Elnök: Lajtha György



1952-ben villamosmérnöki oklevelet szerzett, és a Posta kiserületi Állomáson kezdett dolgozni. 1986-ban áthelyezték Postavezérgazgatóságra, jelenleg a Magyar Távközlés c. újság felelős szerkesztője. Kandidátusi disszertációját az átviteli technika, a doktorit pedig a hálózattervezés témaköréből írta. 1992-ben Széchényi díjat kapott.

HELYI HÁLÓZATOK

Előadó: dr. Nagy Ákos igazgató, X-Byte Számítástechnikai Kft.

Szokolay Tamás igazgatóh., X-Byte Számítástechnikai Kft.

Helyi hálózatok szabványosítása

Hálózati topológiák

- Csillag (Star)

- Sín (Bus)

- Gyűrű (Ring)

- Kombinált topológiák

Hálózat-hozzáférési eljárások (protokollok)

- Token Passing

- CSMA/CD

Hálózati kábelek

- Rézkábelek

- Koaxiális és twinaxiális kábelek

- Sodrott érpárú kábelek

- árnyékolt sodrott érpárú kábelek

- árnyékolatlan sodrott érpárú kábelek

- Optikai kábelek

- Multimód rendszerű

- Monomod rendszerű

Hálózat típusok

- 3 Com Ether Linc és Ether Linc Plus (10 Mb/s)

- IBM PC Network (2 Mb/s)

- Standard Microsystem ArcNet (2,5 Mb/s)

- IBM Token Ring (4, 16 Mb/s)

hogyan lehet választani ?

Bevezetés

Egyre több olyan feladattal, problémával találjuk magunkat, amelyek számítógépek nélkül nem megoldhatók. A számítógépek iránt mutatkozó egyre nagyobb érdeklődés következménye az ilyen gépekből felépített számítógép-hálózatok iránti igény növekedése. A legnagyobb jövőt a szakirodalom kategórián belül az ún. **helyi hálózatok**-nak jósolja. A szakirodalomban leggyakrabban az angol megnevezésből (*Local Network*) képzett betűszó **LAN** szerepel. Nagyobb távolságok (50-60 km) a viszonylag új típusú hálózat a **Metropolitan Network (MAN)** jöhet szóba. Ezt a hálózatot a szakirodalom T3-nak említi. Ugyancsak nagyobb távolságok áthidalására (20-300 km) alkalmas a **Wide Area Network (WAN)**, amelyről T1 megnevezéssel lehet olvasni.

Ennek az előadásnak elsősorban az a célja, hogy a hálózatokról adjon rövid ismertetést és megkönnyítse a választást.

Mit tekinthetünk helyi hálózatnak ?

Helyi hálózatnak nevezzük azt az önálló (helyi) feldolgozó képességgel és erőforráskészlettel (processor, memória) rendelkező számítógépek összekapcsolásával létrehozott hálózatot, amelyben egyes munkállomások konkurens módon érhetik el a hálózat kifejező gépein (file server) tárolt, ill. azok környezetében elhelyezett erőforrásokat (nyomtatók, file-ok). A hálózatban belüli kapcsolatok valamilyen adatátviteli közegen speciális hozzáférési eljárások (adatátviteli protokoll) segítségével valósítják meg. Az egymástól legtávolabb levő gépek közötti távolság általában nem több, mint 10 km-nél.

Helyi hálózatok szabványosítása

A helyi hálózatok kialakulásának kezdetén a legnagyobb problémát jelentette, hogy nem lehetett a különböző típusú hálózatok hálózati rendszerre (internetworking) összekapcsolni, illetve egyik hálózaton kifejlesztett alkalmazói programok nem voltak másikon változtatás nélkül futtathatók.

A szabványosítási törekvéseket először a hálózati hardver szintjén próbálták megoldani, ezt azonban nem volt megvalósítható.

...t nagymértékben korlátozta volna a felhasználói igényekhez való rugalmas alkalmazkodást. A szabványosítás software útja volt járható.

Az adatátvitel struktúrája helyi hálózatokban az adatátviteli rendszerek nyílt architektúráját leíró klasszikus, 7 rétegű OSI (Open System Interconnection - nyílt rendszerek összekapcsolása) modell felépítését követi. (A Nemzetközi Szabványügyi Szervezet ISO - javaslat alapján.)

7. Alkalmazói réteg (Application)
6. Megjelenítési réteg (Presentation)
5. Együttl működési réteg (Session)
4. Szállítási réteg (Transport)
3. Hálózati réteg (Network)
2. Adatkapcsolási réteg (Data Link Control)
1. Fizikai réteg (Physical)

A 7 rétegű OSI modell

Az OSI-modell egyes rétegeinek fontosabb funkciói:

Az alkalmazói réteg teszi hozzáférhetővé a hálózati szolgáltatásokat a felhasználók, a felhasználói programok számára.

A megjelenítési réteg határozza meg az alkalmazói program adatstruktúráját, végrehajtja az átvitelhez szükséges konverziót.

A együttműködési réteg gondoskodik az együttműködés szervezéséről, szinkronizálásáról

A szállítási réteg vezérli a teljes adatsere folyamatot (végponttól végpontig).

A hálózati réteg végzi az összeköttetés létrehozását, fenntartását és bontását.

Az adatkapcsolási réteg feladata, hogy felfedje, esetleg javítsa a fizikai rétegben az átvitel során keletkező hibákat.

A fizikai réteg a hálózati hardware tulajdonságaihoz alkalmazkodva vezérli a tényleges adatátvitelt.

Az OSI modell sajátossága, hogy az egyes rétegek kizárólag a velük szomszédos rétegekkel állnak kapcsolatban (hierarchikus felépítésű), a felsőbb rétegek az alattuk levők felhasználásával működnek. A beépített szabványos interface biztosítja a rendszer hálózati hardware függetlenségét (nyílt az architektúra).

Hálózati topológiák

A LAN topológia határozza meg a hálózatban az elrendeződést és kábelezt. Alapvetően három típusát lehet megkülönböztetni: **Bus**, **Ring** (csillag, soros vagy sín, gyűrű). Kialakítható felépítésű is, ezt lehetne hópehelynek nevezni.

A csillag topológiánál minden egyes adatvégberendezéstől kábel megy a fileserver-hez, vagy a csillagelosztóhoz (HUB). Ez az eljárásnak nagy a kábelszükséglete, de ugyanakkor hibakeresés egyszerű. Egy állomás meghibásodása nem befolyásolja a hálózat működését. A csillag topológia olyan esetben ideális, ha a munkahelyek egymástól távol helyezkednek el.

A soros hálózatnál az állomások egy közös kábelre vannak felfűzve. Ennek a módszernek előnye, hogy kisebb a kábelszükséglete, ugyanakkor a kábelhibát nehezebb lokalizálni. Ha a hálózat valahol megszakad a teljes hálózat üzemképtelenné válhat. Ott célszerű használni, ahol a munkaállomások csoportosan, közel egymáshoz helyezkednek el.

A gyűrű topológiánál a munkaállomások logikailag gyűrűbe, azaz fizikailag csillagba vannak kapcsolva. A hálózat fizikailag kialakítása rendkívüli hibatűrőséget biztosít. Kialakítása könnyen bővíthető, módosítható.

Hálózat-hozzáférési eljárások (protokollok)

A protokoll, vagy másképpen a hálózat-hozzáférési eljárás legfontosabb feladata, hogy adatvesztésmentes átvitelt valósíts meg. A protokoll határozza meg az egyes kábelhálózatban a munkaállomások elérési módját és ezzel az átviteli sebességet is.

A legjelentősebb helyi hálózati protokollok:

- token passing (adási jog továbbításos)
- CSMA/CD (ütközésfigyeléses)

A Token passing eljárás lényege, hogy a hálózatban egy rögzített időközlejtésű és időtartamú bit sorozat, a Token kering, munkahelyről munkahelyre azért, hogy az információt átvegye továbbadja. A Token tulajdonképpen "üres hely"-et jelent, amelyet információval kell (lehet) megtölteni. Csak az az állomás adhat amelyiknél az adási jog (Token) van. Ha ennek az állomásnak nincs mit adnia a Token-t továbbadja a következőnek és e

Folyamat addig folytatódik amíg nem érkezik egy olyan állomáshoz amelyik adni akar. Az üzenet, vagy adatcsomag az adott hosszúságra tördelt adatrészen kívül tartalmazza a címzett és a küldő állomás címeit, valamint néhány vezérlési információt is.

A Token Passing protokollnak a hálózat terhelés változásával szembeni érzékenysége kisebb mint a CSMA/CD protokollé, a rendszer válaszidője a hálózat oldaláról jól definiálható, ami különösen vezérlési, gyártásautomatizálási rendszerekben való alkalmazás esetén alapkövetelmény. Alkalmazási területe elsősorban gyűrű, illetve csillag topologia.

A rendszer nagy előnye a protokoll nyugtázásos rendszere. Ennek működés módja: a célállomás által sikeresen fogadott üzenet vezérlő részében a célállomás csatolóegysége az adatrész kiolvasása után egy pozitív nyugtatabitet állít be, majd az üzenetet továbbengedi a küldő állomás felé. Az üzenetcsomag most már pozitív nyugtaként a küldő állomást értesíti a sikeres adatforgalomról. Amennyiben a küldő állomás időn belül ez nem érkezik meg az adási kísérlet megismétlődik annyiszor amennyi, a rendszerben meg van határozva, ha ekkor sem jön pozitív nyugta hibaüzenetet küld az adás megismétlése céljából.

A pozitív nyugta esetén az eredeti adatküldő állomás köteles üres csatlakozást küldeni az öt logikai vagy fizikai sorrendben következő állomásnak (akkor is, ha van még adnivalója). Ez a szervezési mód teszi lehetővé, hogy egyetlen állomás se monopolizálhassa hosszú időre a hálózatot.

A CSMA/CD (ütközésfigyeléses) protokoll a másik leggyakrabban használt hálózat eljárás a helyi hálózatokban (CSMA/CD a *Carrier Sense, Multiple Access, Collosion Detection* rövidítése). Az a hálózati állomás amelyik adni akar először behallgat a hálózatba, hogy az szabad-e (Carrier Sense, azaz a csatolóegység érzékeli az üzenetcsomag vivő jelet). Ha azonos időpillanatban egyidejűleg több állomás akar adni (Multiple Access) ütközés (Collosion) történik. Az ütközés érzékelése (Collosion Detection) megszakítja az adatátvitelt. Az állomás ilyenkor vár és egy idő múlva ismét megkísérli az adást. Az egyes munkaállomások úgy vannak egymáshoz csatlakoztatva, hogy mindegyik eltérő ciklusidővel próbál a hálózatra csatlakozni. Ezzel a módszerrel az ütközés ugyan nem kerülhető el, de a CSMA/CD az ütközéseket úgy kezeli, hogy tényleges adatvesztés nem történik.

Mivel ez az eljárás véletlenszerű hozzáférést biztosít elsősorban olyan rendszerekben terjedt el, amelyekben a rendszerválasztás megváltozása nem akadályozza a hálózat rendeltetés szerinti használatát. Olyan hálózatokhoz fejlesztették ki, amelyekben egyes hálózati állomások többnyire egymással forgalmaznak, és egyetlen központi állomással. Helyi hálózatokban viszont szinte kizárólag a munkállomások és a hálózat osztott állománykezelésű vezérlő ún. server között történik az adatcsere. Ennek az következménye, hogy a server-en a munkállomások üzenetei (főleg kiszolgálási kérések) egyszerre is jelentkezhetnek, az összeakadhatnak. Ez azokban a hálózatokban okozhat különösen nagy problémát ahol csak egy puffer (átmeneti tároló) áll rendelkezésre az adási és vételi irányban. Ilyen esetekben előfordulhat, hogy a rendszer sikertelennek minősíti a kiszolgálási kérést és megismétli azt, ez a hálózat hasznos átviteli teljesítményét jelentősen csökkenti. Ennek a problémának a kiküszöbölésére a server-ben több puffert alkalmaznak.

Kábeltípusok (átvivőközegek)

Helyi hálózatokban leggyakrabban vezetékes átvitelt alkalmaznak. Előfordulhat azonban, hogy a hálózat bizonyos pontjai között a helyi adottságok miatt ez nem valósítható meg. Ilyenkor az összeköttetést vonalszerűen vagy térbe kisugárzott elektromágneses hullámok útján lehet létesíteni. (A helyi hálózatok egyike a legrégebbike, az ALOHA a Hawaii szigeteken szétszórta állomások között egy számítóközpont között rádió-összeköttetést használt.)

- Fémkábelek
 - telefonkábelek
 - koaxiális és twinaxiális kábelek
 - sodrott érpárú kábelek
 - árnyékolt sodrott érpárú kábelek
 - árnyékolatlan sodrott érpárú kábelek
- Fénykábelek (optikai kábelek)

Az optikai kábel fényárnyékoló réteggel körülvett, henger alakú metrikus elrendezésű, vékony üvegszál, ill. újabban műanyag szál. Helyi hálózatokban elsősorban gerincvezeték-ként alkalmazják. Az állomások csatlakoztatása optika-elektromos jelek alakítók segítségével valósul meg.

ennek a kábeltípusnak előnye, hogy igen nagy átviteli sebességet tesz lehetővé, és nagyobb távolságok áthidalására alkalmas mivel kicsi a jeltorzítása és elektromos zavarokkal szemben gyakorlatilag érzéketlen.

Alapvetően két típusa van: a multimode és a mono-mode rendszerű. A multimode rendszernél a fényforrás LED (Light Emitting Diode), és a fénysugarak besugárzási szöge az optikai szál tengelyéhez képest eltérő, míg a mono-mode rendszernél laserforrás a fényforrás, és a lényegesen nagyobb intenzitású fénysugarak a tengely irányban érik az optikai szálát.

Hálózattípusok

10M Ether Linc és Ether Linc Plus

10 Mb/s átviteli sebességű hálózat, a hálózat hozzáférési rendszere CSMA/CD. Maximálisan 500 m távolság hidalható át, de ismétlők segítségével ez a távolság jelentősen növelhető. A rendszer koaxiális (50 Ω -os) kábellel működik. A hálózattal max. 100 állomás kapcsolható össze, működés megfelel az IEEE 802.3 szabvány előírásainak.

10M PC Network

Hálózat topológiája logikailag sín, fizikailag csillag, hozzáférési rendszere CSMA/CD. Az átviteli sebesség 2 Mb/s. Az alkalmazott kábeltípus 93 Ω -os koaxiális kábel. A legnagyobb hidalható távolság 300 m. A hálózat egy speciális központi elemet tartalmaz, amely tulajdonképpen egy frekvencia váltó.

Standard Microsystem ARC-Net

Hálózat 93 Ω -os koaxiális kábelen üzemelő 2,5 Mb/s átviteli sebességű, token passing hozzáférési rendszerrel működő hálózat. Ismétlők nélkül maximálisan 600 m távolságra lehetnek egymástól az állomások. Kábelkoncentrátor, ún. Hub segítségével kapcsolhatók össze az egyes munkaállomások.

16M Token Ring

Az IBM által kifejlesztett speciális rendszer 4, ill. 16 Mb/s átviteli sebességet biztosít 4 eres árnyékolt csavart érpárú kábelen. A hálózat logikailag gyűrű, fizikailag csillag topológiájú. Hozzáférési eljárása token passing. A hálózatban kábelkoncentrátorok (un. MAU-k) alkalmazásával maximum 260 munkaállomás üzemeltethető.

Dr. Nagy Ákos:

Iskolai végzettsége: BME Villamosmérnöki Kar (1970.)
műszaki doktor (1973.)
Nyelvismeret: német (felsőfok), angol (középfok)
Korábbi munkahelyei: NEVIKI, SZKI, PSZTI
Jelenlegi munkahelye: X-BYTE Számítástechnikai Kft.
ügyvezető igazgatója 1986-tól
Családi állapota: nős, két gyermek apja

Szokolay Tamás:

Iskolai végzettsége: BME Gépészmérnöki Kar (1971.)
Nyelvismeret: német
Korábbi munkahelyei: Magyar Villamosművek Tröszt, PSZTI
ELMÜ
Jelenlegi munkahelye: X-BYTE Számítástechnikai Kft.
ügyvezető igazgatóhelyettes 1986-tól
Családi állapota: nős, két gyermek apja

AT&T SYSTIMAX® PDS strukturált kábelezési rendszer

Előadó: Seregdy Tamás (LIAS Kft.)

Tartalomjegyzék

AT&T SYSTIMAX® PDS

A SYSTIMAX® PDS alapelvei: termékek és szolgáltatások teljes skálája

A SYSTIMAX® PDS alapelvei: modularitás és bővíthetőség

A SYSTIMAX® PDS alapelvei: nyílt rendszer

A SYSTIMAX® PDS alapelvei: választási lehetőség a kivitelben

**A SYSTIMAX® PDS alapelvei: felhasználó általi módosítás,
konfigurálhatóság**

A SYSTIMAX® PDS fejlődési iránya

AT&T SYSTIMAX® PDS: teljes rendszer (összefoglalás)

AT&T SYSTIMAX® PDS

Az AT&T SYSTIMAX® PDS strukturált kábelezési rendszer az AT&T 1986 bevezetett terméke. A SYSTIMAX® PDS egy nagyértékű kábelezési rendszer hang és képi információk továbbítására. Tervezésénél fő szempont volt a bővíthetőség és rugalmasság valamint a rendszer beruházásvédő jellege. Ezek a szempontok, amelyek meghatározták a rendszerrel szemben támasztott igények irányvonalait, illetve ma is meghatározzák a termékfejlesztési stratégiát. Ebben a definícióban fontos kiemelni a SYSTIMAX® PDS rendszer és beruházási jellegét, mert ezek azok a jellemzők, amelyek megkülönböztetik a SYSTIMAX® PDS-t más gyártók által kínált kábelezési komponensektől, alkatrészekből berendezésorientált megoldásoktól.

A SYSTIMAX® PDS olyan épületen belüli és épületek közötti jelátviteli rendszer, amely összekapcsolhat telefonokat, adatfeldolgozó berendezéseket (terminálok, nyomtatók, pénzkidő automata (ATM), stb.), analóg és digitális telefonközpontokat, lokális (LAN) és nagy kiterjedésű (WAN) számítógéphálózatokat, biztonsági és irodai rendszereket egymással és más rendszerekkel.

A jelenlegi rendszerkörnyezet általában egymással nem kompatibilis részrendszerekből áll (több központi gép, terminálok, szövegszerkesztők, stb.). Minden részrendszer más típusú csatlakozási felületet használhat, amely nem kompatibilis a többi rendszerrel (pl. IBM 3174/3274 cluster controller 3270 Type 4 típusú terminálokkal és IBM System AS400, mindegyikük más kábelképzést igényel). Különböző csatlakozási felületek miatt a költözések és változtatások költségesek és nagyobb szakértelmet igényelnek, sőt egyes esetekben egy új kábelezési rendszer kialakítása a helyigény és egyéb strukturális megfontolások (pl. kábelcsatorna) miatt lehetetlenné válik. Berendezésváltáskor gyakori probléma, hogy a már létező kábelezési rendszert a felhasználónak teljes egészében ki kell cserélnie, ez jelentős többletköltséget eredményezhet.

A SYSTIMAX® PDS megoldja ezeket a problémákat, mivel lehetővé teszi az egymással nem kompatibilis, nem szabványos, berendezésorientált rendszer huzalozásának megvalósítását. Teszi ezt az egységes, szabványos illesztési felület biztosításával, ezáltal hosszútávú beruházásvédelmet garantál.

SYSTIMAX® PDS alapelvei: termékek és szolgáltatások teljes skálája

SYSTIMAX® PDS rendszerek kiépítéséhez valamennyi szükséges eszköz rendelkezésre áll: jelátviteli eszközök (négy árnyékoltan sodrott érpárat tartalmazó rézkábelek, üvegszálás fénykábelek, kábelcsatornában és légekábelként alkalmazható, földbe temethető és öntartó kivitelben), kábelrendező, csatlakozók, csatlakozók, adapterek, aktív jelátviteli eszközök, villám- és túlfeszültségvédelmi eszközök, járulékos eszközök (pl. szerszámok). A kábelezési eszközök széles választékához a világon mindenhol azonos színvonalú szolgáltatás társul: építészeti konzultáció, rendszertervezés, kiviteli tervek elkészítése, project management, üzemeltetési eljárás és **5 éves AT&T általi garancia**.

SYSTIMAX® PDS alapelvei: modularitás és bővíthetőség

SYSTIMAX® PDS egy csillag topológiájú kábelezési rendszer, melynek modularitását alrendszer-struktúrája biztosítja. A rendszer hat, egymáshoz kapcsolódó alrendszerből áll:

- munkahelyi alrendszer (munkahelyi aljzat és lengőkábel)
- vízszintes alrendszer (vízszintes kábelek, általában 4-érpáras, árnyékoltan sodrott érpáru rézkábelek)
- adminisztrációs alrendszer (szintenként elhelyezett és az épület központi kábelrendezője)
- felszállóági alrendszer (nagy kapacitású sodrott érpáras rézkábelek, üvegszálás fénykábelek)
- géptermi alrendszer (az épület kommunikációs rendszerének közös erőforrásait (központi számítógép, telefon alközpont) a központi kábelrendezővel összekötő kábelek)
- épületen kívüli (campus) alrendszer (túláram- és villámvédelmi eszközök, kültéri réz- és fénykábelek, kábel szerelési segédeszközök és kellékek)

Az alrendszer struktúrából adódóan az alrendszereket egymástól függetlenül is ki lehet építeni (pl. először csak az épület egy szintjén), a későbbi bővítések csak a már kiépített részekhez. A rendszerben szükséges változtatások egyszerűen megvalósíthatóak, mivel a változások csak egy alrendszert érintenek. A rendszer üzemben tartása nem igényel speciális szakismeretet, a rendszer felépítését könnyen kiképezhető.

A SYSTIMAX® PDS alapelvei: nyílt rendszer

A SYSTIMAX® PDS nyílt rendszerarchitektúrája biztosítja, hogy a különböző gyártók termékei csatlakoztathatóak a kábelezési rendszerhez. A nyílt szabványos illesztési felület és az egyes komponensek kiemelkedő minősége biztosítja. A SYSTIMAX® PDS csaknem valamennyi hang- és adatátviteli szabványt támogatja: ISDN (Integrated Services Digital Network: BRI (Basic Rate Interface), PRI (Primary Rate Interface)), RS-232 aszinkron és szinkron (SDLC, Bysync és X.25) protokollok) adatátvitel, RS-422/423 átvitel, IEEE 802.3 Ethernet lokális hálózat, Token Ring (IEEE 802.5) lokális hálózat (4 és 16 Mbps), FDDI (Fiber Distributed Data Interface) és TP-DDI (Twisted Pair Distribution Data Interface, 100 Mbps) adatátvitel sodrott érpáras médiumon, stb.

A rendszer a nem szabványos eszközök csatlakoztathatóságát adaptív segítségével biztosítja.

Az AT&T SYSTIMAX® PDS strukturált kábelezési rendszer kielégíti a legmagasabb megköveteléseket, meghaladja az EIA/TIA 568 Commercial Building Wiring Standard szabványt. A szabványos illesztési felület és az adapterek segítségével valamennyi telekommunikációs és számítástechnikai gyártó cég berendezései csatlakoztathatók a SYSTIMAX® PDS rendszerhez: digitális telefon alközpontok (AT&T, Siemens, Ericsson, Philips, Schrack, Northern Telecom, NEC, stb.), illetve AT&T, IBM, Compaq, ICL, Wang, HP, stb.

A SYSTIMAX® PDS alapelvei: választási lehetőség a kivitelben

A SYSTIMAX® PDS minden esetben a felhasználó speciális igényeinek megfelelően alakítható ki. Lehetőség van csak sodrott érpáras rézkábelekre, sodrott érpáras rézkábelre és üvegszálalás fénykábelre valamint csak fénykábelre (Fiber to the desktop) alapuló rendszer kiépítésére.

Egy adott kábelezési rendszer tervezése során a felhasználói igények, a várható kapacitásigény, az adatbiztonsági igény és a helyi strukturális lehetőségek azok a tényezők, amelyek meghatározzák az ajánlott média típusát.

SYSTIMAX® PDS alapelvei: felhasználó általi módosítás, konfigurálhatóság

Egy épületben az átkötözések során szükség lehet a kábelezési rendszer módosítására. A módosítást az épület kábelrendezőin kell végrehajtani, a módosításokat - egyszerűségükönél fogva - a felhasználó is megteheti.

Sodrott érpáras rézkábelekhöz két fajta kábelrendező áll rendelkezésre. Egyikük az ún. Cross Connect kábelrendező, amely csak szerszám segítségével, némi gyakorlat után módosítható. A rugalmasabb, kényelmesebb megoldást az ún. Patch Panel kábelrendező adja, amelyen átkötő kábelek egyszerű átcsatlakoztatásával hajtható végre az átkonfigurálás. Mindkét kábelrendező típusból a szükség szerinti mennyiség helyezhető el, számukra nincs korlátozás.

Az üvegszálalás fénykábelek az ún. LIU-ba (Lightguide Interconnect Unit) csatlakoztathatóak, amely funkcióit és rugalmasságát tekintve, a Patch Panel fénykábeles megfelelője.

Mind a rézkábeles, mind a fénykábeles kábelrendezők könnyen kezelhető, kompakt méretűek, rugalmasan átalakíthatóak. Adapterek biztosítják a fénykábelek és a sodrott érpáras rézkábelek illesztését, egy rendszerbe történő integrálásukat.

SYSTIMAX® PDS fejlődési iránya

A termékfejlesztési stratégia eredményeként várható, hogy az üvegszálalás átviteli technológia egyre inkább kiterjed a hang-, adat- és képi információk továbbítására is. Az egyre újabb és újabb adaptertípusok kifejlesztése a vezető gyártók teremtéseinek a SYSTIMAX® PDS-hez való csatlakoztatási lehetőségét teremti meg.

A SYSTIMAX® kábelezési rendszer specializálódását mutatja az, hogy a közelmúltban elkészült a SYSTIMAX® IBS (Intelligent Building System) és a SYSTIMAX® IDS (Industrial Distribution System) kábelezési rendszer is.

Az AT&T-nek, mint távközlési világcégnek a nyílt rendszerek melletti elkötelezettsége, valamint az AT&T Bell Laboratóriumban végzett kutatómunka garantálja azt, hogy a SYSTIMAX® PDS rendszer hosszútávon kielégíti a felhasználói igényeket. Ez jelenti egyrészt azt, hogy az AT&T mindig élenjáró szerepet tölt be a szabványok definiálásában ezáltal biztosítja, hogy rendszerei mindig megfelelnek a nemzetközileg elfogadott szabványoknak, másrészt a termékfejlesztési stratégiája biztosítja azt, hogy a világ vezető gyártóinak teremtéseinek csatlakoztathatóak a SYSTIMAX® PDS rendszerhez.

AT&T SYSTIMAX® PDS: teljes rendszer (összefoglalás)

Az AT&T SYSTIMAX® PDS strukturált kábelezési rendszer hang, adat, szöveg, videóinformáció továbbítására szolgál, tehát minden információátviteli igényt kielégít. Csillag topológiájából adódóan kompakt és áttekinthető, tartalmazza a kábelezési rendszer strukturájának valamennyi alrendszerét. Szabványos illesztési felületekkel rendelkező álló adapterek segítségével támogatja más eszközök és termékeinek alkalmazását, tehát nem köti a felhasználót egy szállítóhoz. A SYSTIMAX® PDS egy egyesített kábelezési rendszer, mert valamennyi jelátvitelt ugyanolyan kábelben, egy rendszerben képes továbbítani.

A fenti jellemzők miatt a SYSTIMAX® PDS alkalmazásának előnyei:

- nyílt rendszerarchitektúra
- nagy teljesítmény/ár arány
- rugalmasság, funkcionalitás
- kézbentartható bővítés
- teljesskörű szolgáltatásnyújtás



AT&T - A megfelelő választás



Seregdy Tamás - LIAS Kft.

1989-ben kapott villamosmérnöki oklevelet a BME Villamosmérnöki Karának Műszer- és Mérés-technika Szakán. Az egyetemi tanulmányok befejezése után a KFKI Számítástechnikai Csoportjához tartozó LIAS Kft.-ben helyezkedett el, ahol programtervező mérnökként dolgozott. 1990. július 1-től kereskedelmi és marketing managerként az AT&T SYSTIMAX PDS strukturált kábelezési rendszer eladásával foglalkozott. 1992. január 1-től kereskedelmi igazgatóként irányítja a LIAS Kft. kereskedelmi tevékenységét. Nős, 1 gyermekje.

Bertáné - Czigány - dr.Kántor - dr.Koós
PKI Távközlési Intézet

A vezeték nélküli távközlő rendszerek tervezésének információ-ellátása terepadatbázisok segítségével.

Bevezetés

A speciális hazai körülmények miatt a földmérési-térképészeti szakágazat a mai napig nem tudta digitális (domborzati) térképpel ellátni a felhasználókat. Ezért kezdődött el '70-ben a távközlési igényeket kielégítő topográfiai adatbázisok (DTM 200, DTM 3000 és DTM 1000) kidolgozása intézetünkben, amelynek eredményeként már 1978 óta számításként működnek a tervezést segítő -- topográfiai adatbázist alkalmazó -- programok. E munka állapotáról, haladásáról, az eredményekről és a kapcsolódó kérdésekről eddig a NJSZT mindegyik kongresszusán hangzott el beszámoló.

A húsz év előtti kezdet célkitűzése -- a rendelkezésre álló számítástechnikai előrelépések által meghatározottan -- csak "nagygépes" batch feldolgozás formájában megvalósítható számítógépes térkép lehetett. A hardware környezet az évek során a DEC PDP-11, majd a BULL GENEVEYELL 2200-tól az ESzR R-30 és R-36-os típuson át a TPA 11/440-ig és a színes CRT-képernyős személyi számítógépekig fejlődött. Ennek kedvező hatása volt a feladat egyre "szebb" kivitelű megoldása, de szükségessé tette a gépek korszerűbbre cserélését és a gépcseréket kísérő (részleges) újakezdeéseket, amelyeket erőnktől tellő új hardverfejlesztéssel igyekeztünk megvalósítani. A számítógépes megoldásba így beépített munka jelentősen csökkent az emberi munka-igényt a vezeték nélküli távközlő hálózat tervezésnél, döntően a "szellemi favágás" jellegű tevékenység területén.

Már a számítógépes térkép önmagában nem teljesen elégíti ki a felhasználókat, még a támogatást szeretnék kapni. Munkájukat sokkal kényelmesebbé tenné, ha a már működő URH, illetve mikrohullámú állomásokról és összeköttetésekről is könnyen, esetleg automatikusan tájékoztatást kapnának, hogy az adott antennatornyok állomásokból adódó költségsökkentést és a többi rádiórendszerrel várható kölcsönös interferencia minimalizálását is segítsék a tervezést támogató programok.

Ezen kívül a felhasználók elvárják (illetve elvárnák), hogy az adatbázisban abszolút adatok legyenek. A napra kész topográfiai adatok biztosítása mindenhol gond, de különösen hazánkban, ahol az utóbbi 40 év alatt két országos (polgári) térképmű jelent meg, a térkép alapján történő update 20...30 éves "naprakészséget" tud csak biztosítani. A domborzati adatoknál az ország döntő többségére még elfogadható is lenne, hiszen külszíni művelésű bányá van csak. De a hullámterjedés útjában akadályként szereplő erdők és főleg az épületek szaporodása dinamikus, ezeknél a 20...30 éves állapot nem kielégítő. A probléma megoldására kísérletek folytak és folynak, hogy a műtávérzékelés útján beszerezhető adatokat felhasználjuk az adatok aktualizálásához.

FEDETTSÉGI TÉRKÉP



Fedettségi kódok:
Sz 4 Uiz Bok Lig Erd ut Fal Kis Nag



MARI 41

ÁRNYÉKTÉRKÉP



Az árnyékterkép kódok max. nagyság

0 3 9 27 81 150 225

1. ábra Árnyékterkép a képernyőn. A képernyő baloldali felén a fedettségi kód alapján szerkesztett domborzati térkép látható, jobb oldalon az ellátási (árnyék) térkép, amelynek lépésközeit a felhasználáshoz alkalmazhatjuk tudjuk meghatározni a program futtatásakor.

A terepadatbázisok

A hazai távközlési tervezés munkáihoz az 1970-es évek első felében az 1972-ben készült, 1:25 000 méretarányú, fekete-barna színnyomású, koordináta-háló nélküli polgári térkép állt rendelkezésre. Ennek a térképnek a kiadásával egyidőben az 1951-ben kiadott 1:50 000 méretarányú, négyszínnyomású, koordináta-hálós katonai térképet be kellett egészíteni. A katonai térképészeti intézet (ma a M.H. Tóth Ágoston Térképészeti Intézet) a polgári térképszelvényekre átszerkesztette a katonai térképek koordináta-hálójait és ennek segítségével lehetett a terepadatokat helyes térbeli pozíciójukkal a térképbe vinni. A gazdasági lehetőségek szabták meg, hogy az elképzelt 100, 200 vagy 300 méteres (közelítő) méretű hálózatból éppen a 200 méterest sikerült megvalósítani. Így a megvalósításra kerülő adatbázis a **DTM 200**.

Az akkor rendelkezésre álló tömegtároló eszköz az ESzR gépeken szokásos 7.25 MB-os kapacitású lemezcsomag volt. Ennek a fizikai tárolási lehetőségéhez, a 3625 byte-os kapacitású track-jéhez igazítottuk a térképi adatok tárolási rendszerét, hogy a lehető legjobb tárolást érjük el. Minden egyes terepelemre három adatot tároltunk:

- a terepelemen belül előforduló legnagyobb (balti tengerszint feletti) **magasság** méterben,
- a terepelemen belüli legnagyobb magasság-különbség 5 méteres lépésközű **d-kód** formájában,
- a terepelem felszínére jellemző **fedettségi kód**.

A tengerszint feletti magasság ábrázolásához 4 decimális jegy szükséges, a két kódhoz egy-egy decimális jegy kell, így egy adatelem 6 decimális jeggyel jellemezhető. A diszk kapacitásának jó kihasználása érdekében a 6 decimális jegyet 3 byte-os bináris számmá kódoltuk. Ennek végrehajtása után a diszk 30 szomszédos track-jén 37 km X 37 km méretű összefüggő terület adatai helyezkedtek el.

A hazai polgári topográfiai térképek nem tartalmazzák a terepadatokat az országhatáron kívül, ott csak a sima fehér papírt "ábrázolják". Az országhatár közelében is a határolás összeköttetéseket tervezni, sőt a határtól távolabb eső telephelyekre is vizsgálni kell a szomszédos országok rádiórendszereivel való kölcsönös zavartatás lehetőségét. Ennek megvalósítása a határ túloldalán lévő terep ismerete nélkül csak nagyon sematikus módon lehetséges, ezért készítettünk egy lényegesen durvább felbontású, de Európa 44 %-ára vonatkozó adatbázist is, amelynek jele **DTM 3000**, összhangban a terepelem közelítőleg 3 X 3 km méretével.

A terepelemek felületének leírásához, a térkép síkrajzi információjának társított használt fedettségi kódok a következő táblázatban ismertetjük:

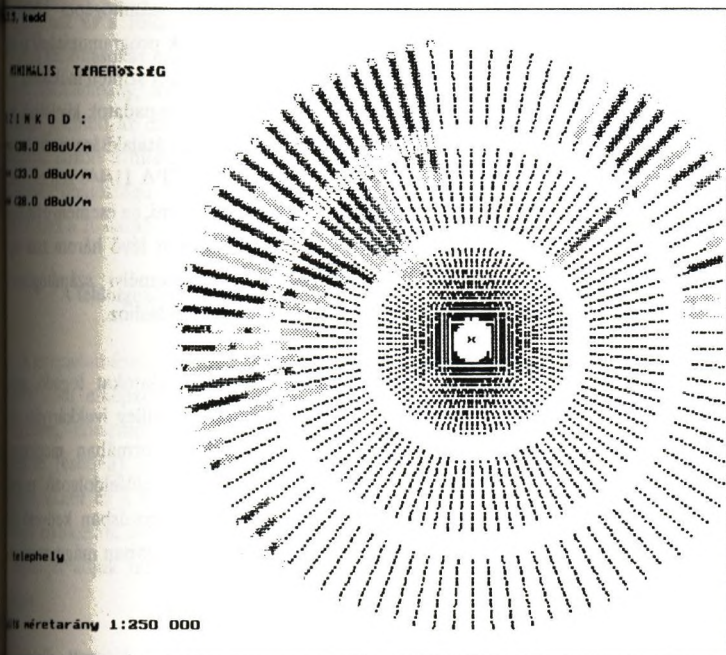
<u>fedettségi kód</u>	<u>a terepelem felszíne</u>
1	mező, rét, szántóföld
2	víz, vizenyős terület
3	bokros, bozótos terület
4	alacsony fák, gyümölcsös
5	magas fák, erdőség
6	épített út, országút
7	falu jellegű beépítettség
8	kisvárosi (elővárosi) jellegű beépítettség
9	nagyvárosi jellegű beépítettség

Ennek szervezési módja, adat-tartalma stb. egyezik a hazai területi adatbázissal. Itt jegyezzük meg, hogy a tervezett "tökéletes kompatibilitás" olyan jól sikerült, hogy végül az egyik adatbázis oszlopfolytonos, a másik sorfolytonos tárolású lett!

Később a DTM 200 adatainak számítógépes konverziójával előállítottuk a harmadik adatbázist is, a **DTM 1000** jelűt. Ennek minden egyes terepeleme 5 X 5 méteres DTM 200 terepelem adatainak összevonásával készült. A 25 darab (maximális) méretű terület átlagát képeztük, a terepegyenetlenséget - 25 méteres lépésközzel - újra meghatároztuk. A fedettségi kódszám összevonása érdekesebb feladat volt! Itt nem lehet egyszerűen az átlagképzéssel eredményt elérni, mert egy rét (F=1) és egy nagyváros (F=9) terepelem átlaga erdő (F=5) lenne. A huszonöt input terepelemre hisztogramot készítettünk, hogy az egyes fedettségi kódok előfordulási gyakorisága milyen, az azonos jellegűeket (növények, illetve építmények) átlagoltuk és a leggyakrabban előforduló jellegű kódot rendeltük az output terepelemhez. Abban az esetben, ha egy output terepelemre több növények, épületek és a vizes területek aránya közelítőleg azonos, valamelyiket önkényesen előnyben kell részesíteni. Ha a döntésnél pl. a víz súlytényezőjét túl kicsire választjuk, a víz elterjedésének ábrázolása a környezettől függően szaggatott lesz.

Az adatbázisok ilyen felépítése az ESzR gépek világában jól bevált, az ED01

DOS 2.2 operációs rendszerrel működő R-30 gépről áttelepítve a rendszert az OS/VS1 operációs rendszerrel üzemelő R-36 gépre az adatállományok - a más kapacitású diszkeket kivéve - átváltás nélkül - változatlan formában működtek tovább.



2. ábra. Térerősség-térkép a telephely körül 96 irányban

A számításokat Okumura és társai cikkében leírt algoritmus szerint végeztük.

Az RSX-11M-PLUS operációs rendszerrel működő TPA 11/440 gépre való telepítésnél a DTM 200 adatbázist alaposan átszerveztük a hatékony hozzáférés érdekében. Mindhárom adatbázis DOS 3.xx operációs rendszerrel működő AT személyi számítógépre telepítésénél a felépítést csak annyiban kellett módosítani, hogy a nagy állományt 21...45...92 file-ra bontottuk a kényelmesebb elérés érdekében. A file-okon a felépítés változatlan maradhatott.

Az adatkezelő programok.

Az adatkezelést és az adatok feldolgozását különválasztva oldottuk meg az adatkezelés mind az ESzR gépeken, mind a személyi számítógépes környezetben. A szubrutinokkal történik, hogy a különböző távközlési feladatok programozásánál ne kelljen az adatvisszanyeréssel újra és újra foglalkozni. A szubrutinok két szinten működnek. Az első szint (DTMKSZ) végzi a koordináták alapján a szükséges terepadatok kijelölését, a második szint (DTMKML) feladata az adatok visszanyerése és átalakítása a diszkenet formából a programok által igényelt H, D, F struktúrába. A TPA 11/440 gép esetében a korlátozott TASKméret miatt - más megoldást kellett kidolgozni, az eseményjelző feladatot az operatív memóriában kialakított régiók útján kapcsolatban lévő három task végzi az adatok kijelölését, visszanyerését és feldolgozását. A személyi számítógépes környezetben a feldolgozásoknál ismét vissza lehetett térni a szubrutinos megoldáshoz.

Az adatkezelő programok közé soroljuk az input adatokat fogadó, az adatokat transzformációkat végző programokat. A fix formátumú, eredetileg lyukkártyára rögzített igény-file tartalmazza az emberi szem számára szokásos formában megfogalmazott feladatot, a vizsgálandó telephelyek koordinátáit stb. Az előfeldolgozó program megengedett adatmegadási formákból egységes, a gépi feldolgozásban kedvező formátumú adatállományt hoz létre a mágneses tárolóközegen (az R-30 világában mágnesszalagra, már mágneslemezen).

Itt kell utalnunk arra, hogy az adatbázis kialakításakor létezett, illetőleg megjelent hazai polgári topográfiai térképek koordináta-rendszere speciális problémát vetett fel. Az 1972-ben megjelent 1:25000 méretarányú térképmű egyáltalán nem tartalmazott koordinátahálót, koordináta-értékeket. Az azóta kiadásra került EOTR térképek az egységes országos vetület szerinti koordinátákat tartalmazzák. Ugyanakkor egyes belföldi alkalmazásoknál, valamint a nemzetközi egyeztetéseknél felmerül a koordinátában (Greenwichtől mért hosszúság, az egyenlítőtől mért szélesség) szereplő az adatok.

Az 1972-es kiadású térképen megkeresett pontok numerikus jellemzőit "szelvényhatáros síkkoordináta" szolgál; az adott térképszelvény számát és a szelvény térképnymatának bal alsó sarkától mm-ben mért távolságot használtuk koordináta helyettesítőként. Az EOTR térképeken fel van tüntetve az EOVSZ koordináta, tehát az adatok számítására megadására van lehetőség. Ugyanakkor a szelvényhatáros síkkoordináta, az EOVSZ

koordináta és a földrajzi koordináta közötti átszámítást, illetve ezek és az adatbázis belső tárolási rendszere közötti átszámítást biztosítani kellett.

A szelvényhatáros síkkoordináta esetén az 1154 darab térképszelvény négy végpontjának és a földrajzi koordinátáknak összekapcsolására szolgáló adatállomány, valamint a számításokat ennek segítségével elvégző szubrutin oldotta meg a tájékoztatót. Mind az adatállomány, mind a program **szigorúan titkos** minősítésű volt, ami bizonyos mértékű bizalmas jellegűt jelentett az ezzel kapcsolatos munkában. Az EOVS koordináták és a földrajzi koordináták között a híradástechnikai tervezés, a DTM 200 adatbázis terepelem-mérete és a megadott pontosságú átszámítást zárt alakban, harmadfokú polinommal meg lehetett oldani; itt csak a 20 darab együttható lett szigorúan titkos.

A feldolgozó programok

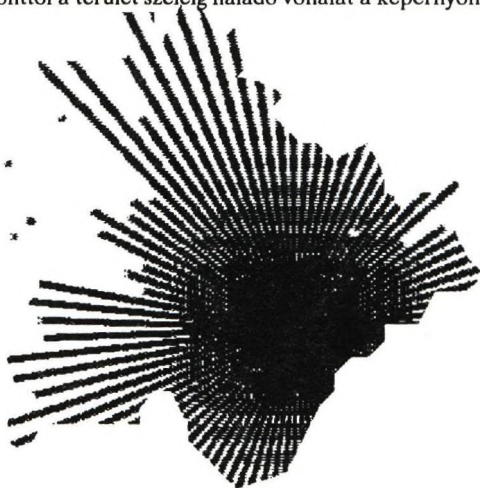
A terepadatbázisok felhasználását a híradástechnikai, távközlési tervezésre többféle módon lehet elképzelni, de minden formának alapvető művelete egy terepszakasz (terepmetszet, hosszmetesz) adatainak visszanyerése és azon az átlátási lehetőségek, mint az (esetleg) előforduló reflexiós helyek vizsgálata. Ha egyetlen rádiószakaszt vizsgálunk, akkor ez a teljes feladat; ha valamilyen összetett munkát végzünk, akkor ezt meg kell ismételjük ciklikusan. Árnyszerű eltérést jelent csupán, hogy ennek a terepszakasznak a végpontját adjuk meg, vagy az egyik végpontot, az irányszöveget és a másik végpont koordinátáit.

A tizennégy évi üzemserű felhasználás tapasztalatai alapján látszik, hogy a felhasználók a feldolgozásokat a tervezéshez és a Frekvenciagazdálkodási Intézet feladatai elnyeréséhez igénylik. A második eset az egyszerűbb, hiszen ehhez általában az összeköttetésre vonatkozó terepmetszet, valamint a telephelyek körül 12 irányban (30 fokos lépésközönként) előírt távolság készített metszet-sorozat gépi elkészítése szükséges.

A tervezésnél többféle számítógépes output iránt mutatkozik igény. A legegyszerűbb az az eset, amikor adva vannak a telephelyek (kirendeltségek stb) és azokat kell hálózatra kötni. Ennél a feladatnál két kérdés adódik: melyik állomásokat lehet (célszerű) összekötni és milyen magas antenna-árbóc biztosítja a kívánt pontosságú összeköttetést. Mind a két kérdésre oly módon kapunk választ, hogy a programmal megvizsgáljuk az adott telephelyek közötti terepmetszeteket és azon értékeljük a szükséges antenna-magasságot.

Bonyolultabb eset az, ha valamely terület "minden" pontjának rádiós elérési körzetét kell meghatározni, például egy körzeti villanyszerelő diszpécser rádióösszeköttetése vagy terület rádió-, illetve TV műsorral való ellátása esetén. Ebben az esetben sokszögvonallal kijelöljük a vizsgálandó terület határát, majd a központi telephely és a sokszögvonallal kijelölt terület közötti távolság alapján a vizsgálatot; ilyenkor általában csak árnyéktérképet szoktunk végezni, mert a sok irány bonyolultabb számítás esetén megnevelti a feldolgozás időigényét. Az árnyékoltsági vizsgálatnál gyakran értékelési programmal az árnyék "mélységét", hogy milyen magas antennaárbóccal lehet kinyitni az árnyékból; ilyen esetben az árnyék mélységét (néhány fokozatot tartalmazó színkódokkal) ábrázoljuk a képernyőn látható vagy rajzgéppel készülő térképen.

Az árnyéktérkép-szerkesztés gyorsabb futású program, de az eredmény, az azonos pontatlanabb közelítése a várható télerősségeknek. Ezért szoktunk télerősség-számítási programot is alkalmazni valamely terület ellátási vizsgálatához. Ebben az esetben is egy sokszögvonallal határolja a vizsgálat tárgyát képező területet, amelynek határpontjait rendre összeköti a program a központi telephellyel és az adott algoritmus (Okumura módszer) szerint számítja ki a terület egyes pontjain várható télerősséget. Általában egy vagy néhány télerősségszint átlépését figyeli a program és ennek megfelelően színezi a középponttól a terület széléig haladó vonalat a képernyőn, illetve a papíron.



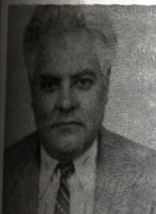
3. ábra. Télerősség-térkép az ellátandó terület ábrázolásával.

Budapest területének ellátási télerősség-térképe a város konturvonalával határolva.

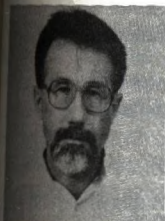
Ís szerzők a PKI Távközlési Intézet munkatársai:



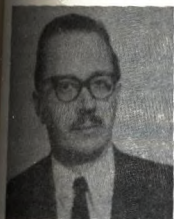
Berta Sándorné tudományos munkatárs



Czigány Sebestyén műszaki főmérnök



dr. Kántor Csaba fejlesztési főmérnök



dr. Koós Árpád tudományos főmunkatárs

ORSZÁGOS X.25 BÁZISÚ IBUSZ RENDSZER MEGOLDÁSAI ÉS TECHNOLÓGIÁI PASZTALATAI

Farkas Gábor, Ivánka Gabriella, Leporisz György SzKI Rt.
Boráros András, Madaras Gábor IBUSZ Rt.

1. Bevezetés

1989. elején az IBUSZ szerződést kötött az SzKI-val egy, az IBUSZ átvilágításán számított gépes rendszerjavaslat megfogalmazására. Többszöri átdolgozás után a végleges rendszerjavaslat 1990. közepén került elfogadásra, utána kezdődtek el a kivitelezési munkák. Az IBISZ (IBUSZ Information System) 1991. június 10-e óta üzemszerűen működik.

A cél egy olyan számítógépes információs rendszer létrehozása volt, amely biztosítja az ország egész területén elhelyezett több száz munkaállomásról számítógépes ismeret nem rendelkező felhasználók igénybe vehessék az IBUSZ üzleti tevékenységét támogató számítógépes szolgáltatásokat.

A rendszer kialakításának főbb szempontjai a következők voltak:

- szabványos, általánosan forgalmazott, professzionális hardware és software komponensek használata,
- szabványos telekommunikációs felületek alkalmazása,
- maximális (banki elvárásoknak is megfelelő) megbízhatóságra tervezés (hibatűrőképesség, tranzakció orientált feldolgozás),
- kommunikáció-kímélő megoldások alkalmazása a magyarországi kommunikációs infrastruktúra elégtelensége miatt,
- osztott adatbázis létrehozása a telekommunikációs terhelés csökkentésére,
- a feladat bonyolultságához képest mérsékelt ár,
- a szolgáltatások használatában egységes technológia alkalmazása,
- nyitottság a bővíthetőségre, új elemeknek a rendszerbe integrálására.

2. A stratégiai elemek jellemzői

Kialakítottunk egy, a fenti szempontoknak megfelelő, általánosan alkalmazható stratégiai eszközbázist, amelynek felosztása a következő:

- RENDSZERPLATFORM, amely az építkezés típusa elemeit, a rendszerben szereplő interface-ek megoldásai működési elveket, a nyitott építkezési technológiát foglalja magában.
- SYSTEM PANEL-ek, az integrált szolgáltatások kialakításának olyan saját fejlesztésű eszközei, amelyek transzportábilisak a különböző rendszerek létrehozásához.
- FEJLESZTÉSI, INTEGRÁCIÓS TOOL-ok, a rendszer elemek összekapcsolásának, az alkalmazások fejlesztésének, az integrált feladatok elvégzésének transzportábilis eszközei.

A rendszer tartalmaz egy általános célú, transzportábilis "ALAPRENDSZER"-t és egy az adott feladatra jellemző, specializált "ALKALMAZÓI RENDSZER"-t.

3. A rendszerplatform eszközbázisa

- Munkaállomás: Wearnes 286 PC (ALR szabvány)
operációs rendszer: MS-DOS v 3.30
- Server: ALR 386-os PC
Központi gép: Olivetti LSX 4240-es fault tolerant gép
processzor: 4 db M68030 (duplexált)
memória: 32 Mbyte (duplexált)

- Mérték: 1.5 Gbyte (duplexált)
- Operációs rendszer: VOS v 10.0
- LAN: NOVELL v 2.15, v 2.2, v 3.11
- X.25: kapcsolóközpont és XENIX alapú felügyelő gép (SZTAKI)
- X.25 PC kártya a COMFE gépekbe (SZTAKI)
- X.25 GATEWAY (SzKI)
- Modem: Bullett ETECH 2400 és 9600 Bps
- Adatbázis: ORACLE RDBMS v 6.0 az LSX gépen
- NOVELL SQL a szervereken
- Cípper v 5.01 a munkaállomásokon és a szervereken

Az alaprendszer architektúrája

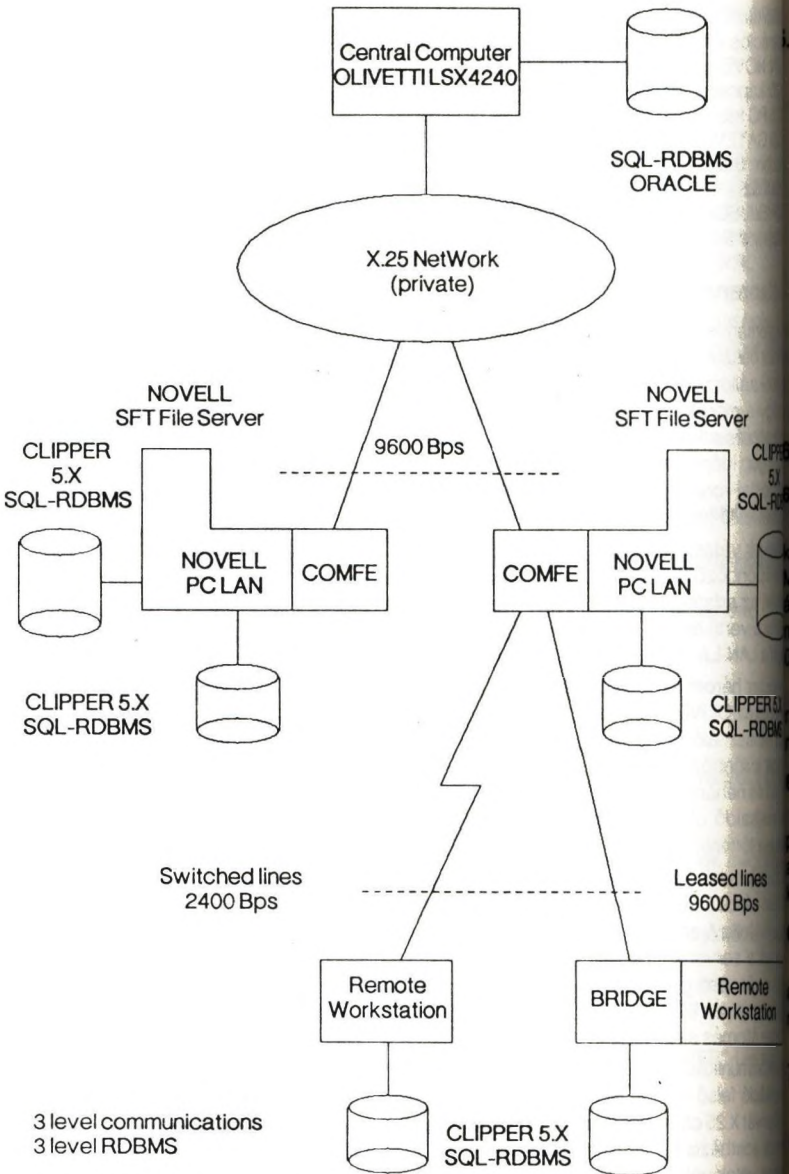
A hagyományos felépítésű rendszer esetén (intelligencia nélküli terminálok, centralizált hálózati összeköttetések, központi számítógép, megközelítőleg annyi bemeneti csatorna, mint munkaállomás) a több száz munkaállomás számítástechnikai igényeit csak egy megfelelően nagy méretezett számítógép tudja biztosítani. Másik megoldás, hogy csökkentjük a központi gép és a kommunikációs utak terhelését. Mi ez utóbbit választottuk, nemcsak azért, mert a gép kiépítettségét (és árát!) radikálisan csökkenteni lehetett, hanem azért is, mert a hálózati vonalak minősége az első megoldásból adódó rendkívül nagy mértékű terhelés ellenére nem kellő hatékonysággal biztosítja a megfelelő szintű telekommunikációt.

Az általunk választott modellben az intelligencia és az adatbázis egyaránt el van osztva a rendszer különböző komponensei között. Az 1. ábrán követhető végig a rendszer felépítése, a központi szerver, az adatbázis és a kommunikáció három szintje. Itt szeretnénk megjegyezni, hogy a rendszer a vertikális és -horizontális kapcsolatok létrejöttét egyaránt támogatja, ezen tulajdonságát a LAN-LAN kapcsolatokban használjuk fel.

A rendszer három alkotó szintje a munkaállomások és a rendszertechnikailag egyenrangú terminálok és az OLIVETTI számítógép. A munkaállomáson történik az ügyfelek kiszolgálása. Itt a felhasználó-barát felületű alkalmazói programok, amelyek lekérdezik és módosítják a központi szerver különböző szintjein lévő adatbázist az ügyfél kívánságainak megfelelően. Az OLIVETTI számítógép és néhány kiegészített LAN napközben a munkaállomásokról érkező kéréseket előfeldolgozza, munkaidő után pedig az esti feldolgozó programok futtatása, a központi listák készítése történik.

Az adatbázis három szintje megfelel a rendszer előbbi szintjeinek. A munkaállomásokon a felhasználó által változó (leíró, informatív adatokat tartalmazó), illetve a helyileg szükséges adatok tárolása történik. A szervereken több munkaállomás közös adatállományai és ideiglenes állományok (pl. a szerveren keresztül valósul meg az adatállományok és az új program verziók tárolása a központi gépről a munkaállomásra) találhatóak. A központi gépen azok az adatok tárolódnak, amelyek elérhetőségét, illetve mindig az aktuálisan legfrissebb állapotát biztosítani minden állomás számára.

A telekommunikációs rendszer három szintjéből a középső a már említett helyi hálózat. A telekommunikáció felső szintje egy a CCITT 1980, 1985. ajánlásai szerint bérelt postai vonalon működő privát X.25 csomagkapcsolt hálózat, amelyen keresztül történik a LAN-ok egymás- és a központi adatbázis kezelő számítógép közötti adatforgalom. A hálózat létrehozása annak érdekében történt, hogy bármikor könnyen megvalósítható legyen a csatlakozás a központi X.25 szolgáltató hálózathoz (nyilvános, műholdas: VSAT stb). Az alsó szintet pedig a hálózathoz távoli munkaállomások aszinkron vonalon (bérelt vagy kapcsolt telefonvonalak) csatlakoztatása jelenti.



Typical processing is the Transaction control based on RDBMS

COMFE = COMMUNICATION FRONT END for NOVELL NetWare

az alapszerver jellemző szolgáltatásai

SERVER/CLIENT tranzakció feldolgozás

Üzenetküldési rendszer

File to file transzfer

A központi gépről ill. bármely LAN-ról letöltés a szerverekre és a munkaállomásokra:

adatbázis tábla

program

Diagnosztika:

X.25 felügyelő rendszer

Server és munkaállomás háttértárolóinak karbantartása a központi ill. bármely kijelölt LAN-ból

Novell nyomkövető rendszer

Figyelés, kijelzés a központi gépen

Összeváltás, összehasonlítás a munkaállomáson

rendszerplatform fejlesztett komponensei

munkaállomásokon

A munkaállomások üzemeltetése számítógépes kultúrával, fegyelemmel nem rendelkező munkahelyeken történik, meg kellett oldani illetéktelen felhasználás elleni védelmet. Kellott a gépek BIOS-át annak érdekében, hogy csak az általunk előállított speciális floppy-kat fogadja el, egyéb floppy-kat ne tudjon olvasni, se a rendszert betölteni róluik. (A készített BIOS természetesen kompatibilis az eredetivel.) Így idáig sikerült vírusmentesen betölteni a rendszert.

A munkaállomás tárrezidens komponensei BLEND, LANCOM, ROUTER, UNIUSER, KERET a rendszerépítő technológia modulkészletét alkotják, szolgálják az integrációt és az üzemeltetés követelményeinek kielégítését.

szervereken

Az üzleti igények kielégítésére ún. Value Added Process-eket (VAP) alkalmazunk. Ilyen munkák végzik például a munkaállomások és a szerverek adatbázisainak frissítését, ill. a munkaszközök programok letöltésére, file-transzferre és üzenetek küldésére, fogadására a távkezelés feladatainak elvégzésére.

központi gépen

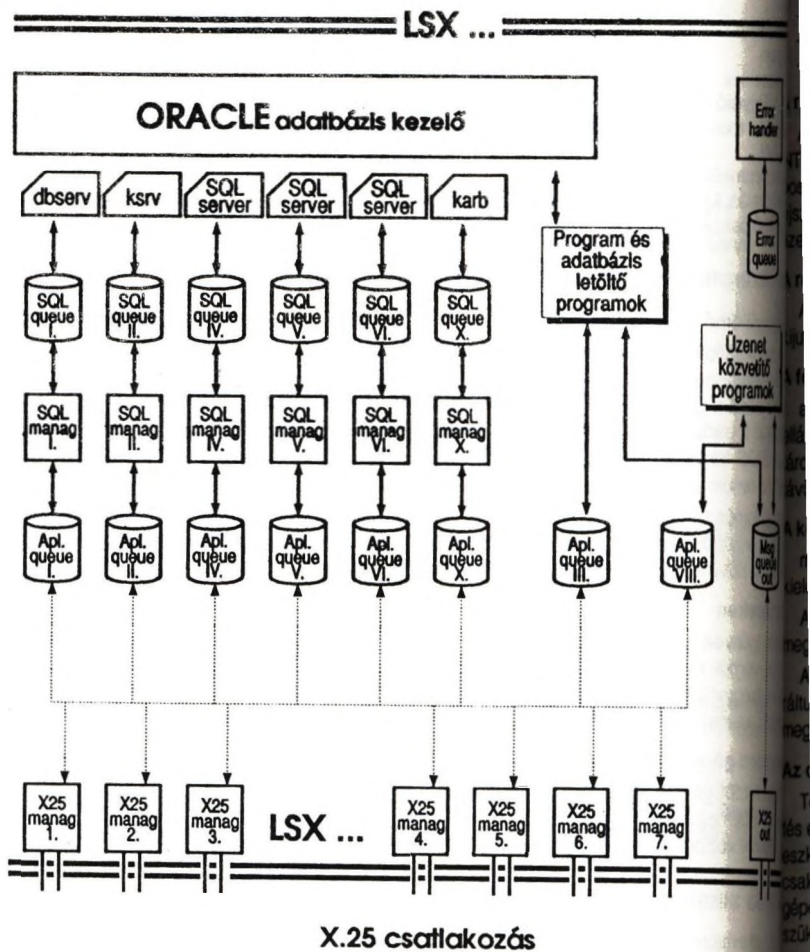
A központi LSX gépen egyrészt a SERVER/CLIENT technikán és az X.25 kommunikáción alapuló ORACLE hozzáférés komponensei, másrészt a rendszer üzemviteléhez kötődő programok futnak. A központi gép programjaival architektúráját a 2. ábra szemlélteti.

az IBISz alkalmazói rendszere

Az IBISz alkalmazói rendszerén az IBUSZ Rt. ill. 1991. július óta az IBUSZ BANK Rt. üzletágai az alkalmazói igényeiket kielégítő software rendszer egészét értjük.

A rendszer kezeli a devizaszámla kezelő rendszert, majd a kiutazási rendszer készült el. E két rendszer használatával egyidőben folyamatosan készülnek a további alkalmazások. (PI. üzletág, szállodai helyfoglaló rendszer, banki felhasználás számára számlavezetés, kassza, határidős betét, stb.) Az IBISz rendszer főbb jellemzőit és a rendszer országos elterjedését a 3. ábra mutatja be.

IBISZ LSX gép SW komponensei



X.25 csatlakozás

2. ábra

IBISz üzemeltetési tapasztalatai

IBISz üzemeltetési tapasztalatait néhány - véleményünk szerint meghatározó témakör-
szegreghetjük

- rendszer üzembehelyezése

- rendszer védelme az illegális beavatkozástól (behatolástól)

- a folytonos üzem kérdései

- a kommunikáció megbízhatósága

- az országos rendszer üzemeltethetősége Budapestről a Dob utcából

Rendszer üzembehelyezése

IBISz üzembehelyezésének kritikus voltát az jelentette, hogy az IBUSZ on-line DATAPO-
minál orientált alkalmazói rendszerét kellett párhuzamos üzem nélkül ugyanazokat a
vonalakat használó LAN-okra mindössze kb. egy hét alatt lecserélni. Az átállást három
éves próba előzte meg. A tényleges átállást 1992. június 1-9 között végeztük el, az
szertű irodai működés 1992. június 10-én reggel hét órakor indult meg.

Rendszervédelem

alkalmazások csak védett floppy-t olvasnak el, az alkalmazásokból az MS-DOS-ba
nem lehet, csak az alkalmazás által megengedett billentyű-kóddok juthatnak a PC-kbe.

Állandó üzem

erősen tolerant OLIVETTI LSX gép ill. a szünetmentes áramforrással és éjszakai árammal
NOVELL Server és X.25 Gateway biztosítja. Így az IBISz rendszer működését megha-
zó elemek éjjel-nappal működnek, biztosítva ezzel a folyamat Dob utcából történő on-line
üzemeltethetőségét.

Kommunikáció

megbízhatósága a vártnál jóval magasabb szintű, jóllehet néhány vonal minősége nem
föld.

üzemképességet az X.25 és az MNP 5 hibavédős modemek használata lényegesen
vevél.

UTAZÁS '92 kiállításától kezdődően a MATÁV SAT-NET Kft. műholdas rendszerét integ-
4 ábra) az IBISz néhány irodájába, a kommunikáció megbízhatósága ugrásszerűen
t, így a banki igényeket is messzemenően kielégíti.

Országos rendszer üzemeltethetősége

területtel a 100 irodára igen komoly feladatot jelent. Kidolgoztuk az egy helyről üzemelte-
közökeit (DOWNLOAD, UPLOAD, háttértár és PC-s programok távkarbantartása). Ezen
közök biztosítják az adatok és háttértárolók távkarbantartását, programok cseréjét, így
gép meghibásodása esetén kell személyesen a helyszínre menni. Tekintettel az ALR
használására és a kiszállítás előtti egy hetes "égetés"-re a személyes helyszíni jelenlét
képessége a rendszer méretéhez képest elenyésző mértékű.

IBISz rendszer rendelkezik azokkal az eszközökkel, amelyek lehetővé teszik, hogy a
szemben újabb és újabb elemek, illetve alkalmazások jelenhessenek meg az üzemszerű
és megzavarása nélkül. A rendszer alkalmas tehát a piaci igények által diktált változá-
sors, az üzemet nem hátráltató követésére.

IBUSZ-Rt. és IBUSZ BANK Rt. országos információs rendszere (IBISz)

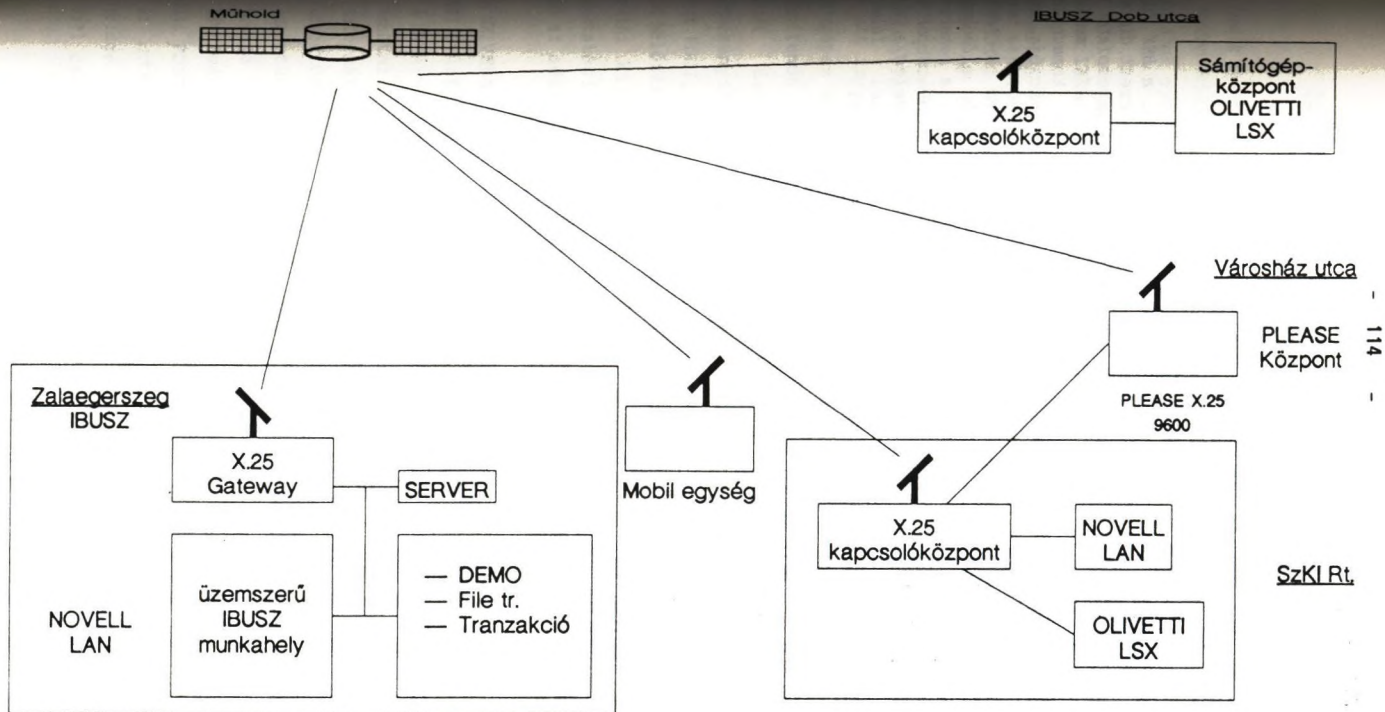


Főbb jellemzők:

- 1 db Olivetti LSX 4240* fault be-
központi gép,
- kb. 100 db iroda az ország egész
rületén,
- ebből: kb. 50 db NOVELL LAN-
szer, 50 db távoli munkaállomá-
s,
- összesen kb. 400 db munkáló-
állomás,
- X.25 privát adatkommunikációs
rendszer,
- kb. 100 kommunikációs kapu
MODEM-mei,
- On-line tranzakciófeldolgozás,
— Message és Download rendszer,
— LSX-LAN és LAN-LAN kapcsolás,
— SQL és Client-server adatkom-
munikáció,
— ORACLE bázisú osztott adat-
kezelés.

Legnagyobb működő országos on-line tranzakciós rendszer az SzKI Rt.-IBUSZ BANK Rt. együttműködés eredménye.

3. Ábra



SzKI Rt. MATÁV szolgáltató: SAT-NET, PLEASE és IBUSZ Rt. együttműködésben

4. Ábra



Leporisz György

1945-ben született Budapesten, a BME Villamosmérnöki Karán 1971-ben szerzett diplomát. Már az egyetemi évei alatt kapcsolatba került az SZKI-val, azóta is az SZKI munkatársa. A számítástechnika és a kommunikáció C&C OSI elvek szerinti integrációja, rendszeres szerteknikai problémáinak megoldása jelentős szakmai működésének meghatározó területei. A C&C témában 1973 óta tart előadásokat, cikkeit jelentetett meg, ill. a lehetőségek szerint az ISO egy-egy munkacsoportjának munkájában is részt vett. Jelenleg a C&C területén a rendszerintegrálási feladatok összefogását, irányítását végzi.

Debreceni Universitas Adatátviteli Hálózata

Terdik György, Dr. Agócs László, Dombrádi Zsolt, Gál Zoltán,
Herdon Miklós, Korcsolay Zsolt, Kovács György, Dr. Krausz
László, Sági György, Dr. Székely Géza

Universitas 4010 Debrecen, Pf. 58.

Bevezetés

Korszerű felsőfokú oktatás és kutatás alapvető objektív feltétele a fejlett információs rendszer. Hiába oktatnak a legkiválóbb szakemberek a legmodernebb eszközökkel, ha az új technológiai információkhoz való hozzáférés lassú, vagy akadózik. A fejlett információs rendszer 3 eleme: korszerű kábelhálózat, számítógépes adathálózat, integrált könyvtári rendszer.

A telepítendő rendszer az Universitas tagintézményeit és a Posta Hálózatát összekötő sokeres optikai gerincvezetékéből és a tagintézmények optikai gerinchálózatából áll. Az intézményközi hálózat FDDI, a belső hálózatok Ethernet hálózatok lesznek. A külső hálózatot mind adat, mind telefon forgalom lebonyolítására használni kívánjuk. Arra számítunk, hogy a későbbiekben más felsőoktatási intézmények és közgyűjtemények is csatlakozni kívánjanak az adathálózathoz.

Az UNIVERSITAS FDDI gerinchálózat vázlatos terve

FDDI gyűrű

A MATÁV-tól indul, ahová a szálakat optikai rendezőre kell bekötni. A gyűrűn levő bekötési sorrend a következő: Református Teológia, Atommagkutató Intézet, Debreceni Orvostudományi Egyetem, Kossuth Lajos Tudományegyetem, Agrártudományi Egyetem és vissza a MATÁV-hoz. A gyűrű hossza kb. 10 kilométer, az intézmények közötti maximális távolság 2800-3000 méter

Ethernet FDDI csatlakoztatás

Az intézmények helyi LAN-jait az FDDI gerinchálózathoz bridge teljesítményű routereken (broutereken) keresztül kell csatlakoztatni.

Felhasználás

Az Universitas egyetemein a hálózatba bekötött PC-kről terminálokról a hálózatra kötött erőforrások szolgáltatásait lehessen elérni. A szolgáltatások teljesség igénye nélkül a következők lesznek: könyvtár szolgáltatások karakteres és grafikus módban való képtovábbítás, applikációs szerver szolgáltatás, disztribúciós szerver szolgáltatás, adatbázis szolgáltatások, központi információs rendszer, news server, elektronikus levelezés.

Hálózati management

Az Universitas FDDI hálózatát és a KLTE helyi Ethernet hálózatát a KLTE Informatikai és Számítóközpontja fogja menedzselni.

3. ATOMKI optikai kábeles számítógépes gerinchálózata

A Bem téren az ATOMKI, a KLTE Kisárletti Fizikai Intézet és a DOTE Orvosi Vegytani Intézete székel. A kialakítandó hálózatnak biztosítani kell, a társintézményeknek az ATOMKI intézményükkel való kapcsolatát, az ATOMKI-nak a Universitas tagintézményeivel való összeköttetését és az ATOMKI és az Internet összekapcsolását.

Az ATOMKI belső hálózatán Internet szolgáltatásokat (remote login, network virtual terminal, file transfer, electronic mail) kívánunk kialakítani az intézetben levő munkaállomásokra, és NFS szolgáltatásokat a ciklotron elhelyezendő Unix szerveren. DECnet szolgáltatásokat kívánunk biztosítani a KLTE VAX gépén és a VIII. épületben levő mikroVAXon. Az I. épületben levő Novell szerver IPX-TCP/IP gatewayként kívánjuk használni. IPX protokollal nem akarunk a rendszerre installálni. Az Internet nemzetközi szolgáltatásait kívánjuk igénybe venni az országos hálózaton keresztül, annak beindulásáig X.25 vonalon. Az ATOMKI-ben jelenleg van X.25 végpont.

A Bem téren kialakítandó csomópontnak kettős csillag topológiája van. Az egyik csillag főleg az intézményi forgalmat bonyolítja, a másik az ATOMKI belső hálózata.

Az FDDI node-ban kialakítandó rendszer routeren keresztül csatlakozzon a városi FDDI gyűrű üvegszállaira, biztosítva

4 üvegszálás Ethernet vonal és 1 X.25 vonal fogadását. Legyen megoldva a gateway funkció és az Ethernet portok elválasztása (bridgelése) portonként külön-külön. A központ port szinten legyen vezérelhető. A csomópontnak kártya rendszerben bővíthetőnek kell lenni.

A ciklotron központban kialakítandó rendszer routeren fogadja a bejövő központból az optikai jelet és ágaztassa el a fent felsorolt 6 optikai végpontpont felé és szolgáljon ki 1 helyi koax Ethernet szegmenst. A központ port szinten legyen vezérelhető. Legyen megoldva az Ethernet portok elválasztása (bridgelése) portonként külön-külön.

Hét optikai végpont (Kis. Fiz., ATOMKI I., II., VI., VIII., IX. és XII. épület) fogadja az optikai kábelt fibre optic transcieveren, és biztosítja 3-3 koax szegmens kiszolgálását. Legyen lehetőség végpontonként további 2-2 kártya bővítésre (a bővítési lehetőség tartalmazza a vezérelhetőséget). A hideglaborban és az Orvosi Vegytan Intézetben levő központ fogadja az optikai kábelt fibre optic transcieveren, és biztosítsa 1-1 koax szegmens kiszolgálását.

Az optikai kábel a központok között 10, egyébként 6 szálás lesz. A jelen kiépítés mind a 10, mind a 6 szálás kábelen két szálat használ

NOTE optikai kábeles számítógépes gerinchálózata

A hálózati koncepció

A DOTE ethernet rendszerű optikai számítógéphálózata az elméleti intézetek alhálózatait, a klinikák alhálózatait és a gazdasági hivatal alhálózatait hivatott egy hálózatba integrálni.

Az adatbiztonság ill. a felesleges adatmozgás megakadályozása érdekében az említett három csoport egységeinek egymástól való leválasztását beépített hidakkal kell biztosítani. Elképzeléseink szerint az egyetem területén egy ethernet optikai gerinc szegmens fut körbe, amely gerincet egy beépített ethernet hid zárt gyűrűvé alakít. Ezen a körgyűrűn helyezkednek el az elméleti intézetek, a klinikák, a gazdasági hivatal épületeit

csillagszerűen összefogó csomópontok. A hálózatba kötött épületek területi elhelyezkedését figyelembe véve a koncepciónkban hat híddal leválasztott csomópontot jelöltünk ki. Minden kialakított csomópontból csillag alakzatban optikai szálal ethernet szegmensek futnak. Az egyes épületekben elhelyezett végberendezésekig, a végberendezések minden végpontnál egységesen négy épület belüli koax-ethernet szegmens csatlakozási lehetőséget biztosítanak a helyi hálózatok számára.

Az említett csomópontokon kívül a körgyűrűre csatlakozó hálózat még közvetlenül - hid közbeiktatása nélkül - az egyetemi könyvtár és a kollégiumok épületei is.

4.2 Követelmények

A DOTE belső hálózatát egy központi optikai kábeles ethernet gyűrű, és erre hidakkal kapcsolódó csillag topológiájú alhálózatok alkossák, moduláris felépítésű aktív elemekkel. A gyűrű csatlakozzék a városi hálózatra, a későbbiekben pedig maga is legyen átépíthető FDDI rendszerűre.

A kábelezés optikai kábellel történjen, a gyűrűn 8 eres csatlakozó részeken 4 eres kivitelben, aminek a fele min. legyen tartalék.

Adatbiztonsági okokból és a felesleges adatforgalom elkerülése érdekében a hálózatnak a koncepcióban megjelölt része hidakkal legyen leválasztható a gyűrűről illetve városi hálózatról. A központi gyűrű híddal kapcsolódó önmagához és hiba (pl. szálszakadás) esetén automatikusan zárjon le. A hálózat nagy kiterjedése és bonyolultsága miatt legyen központilag, szoftveresen is menedzselhető. A hálózati szoftver multiprotokolos legyen, támogassa a TCP/IP, DECnet, X.25 és további protokollokat.

5. DATE optikai kábeles számítógépes gerinchálózata

5.1. Topológia

- a., A DATE lokális hálózata az épületek elhelyezkedése miatt ETHERNET csillag topológiájú optikai kábelezéssel készüljön el az Intézmény egyes épületei között.
- b., Tegye lehetővé az Intézmény lokális hálózatából

városi FDDI-ra történő csatlakozást.

- c., Adjon lehetőséget a közeljövőben várható nagyobb teljesítményű számítógépek, workstationok számára az FDDI-ra történő közvetlen csatlakozásra a központi és az Informatikai Központ épületéből.
- d., Router funkcióval biztosítsa az Intézmény számára X.25 WAN hálózat, valamint X.21, RS232 terminálvonalak kezelését.

1. Kábelezés

Az épületek közötti kábelezés optikai kábellel történjen rágcsálóvédelemmel kivitelben.

1. Hálózati menedzselés és protokollok

- a., A hálózat fizikai kiterjedése és a csomópontok nagy száma miatt a helyi hálózat menedzselhető kell legyen.
- b., A hálózati szoftvernek multiprotokolosnak kell lennie (TCP/IP, DECnet, X.25, esetleg más protokollok is).

Református Teológiai Akadémia és Nagykönyvtár optikai kábeles számítógépes gerinchálózata

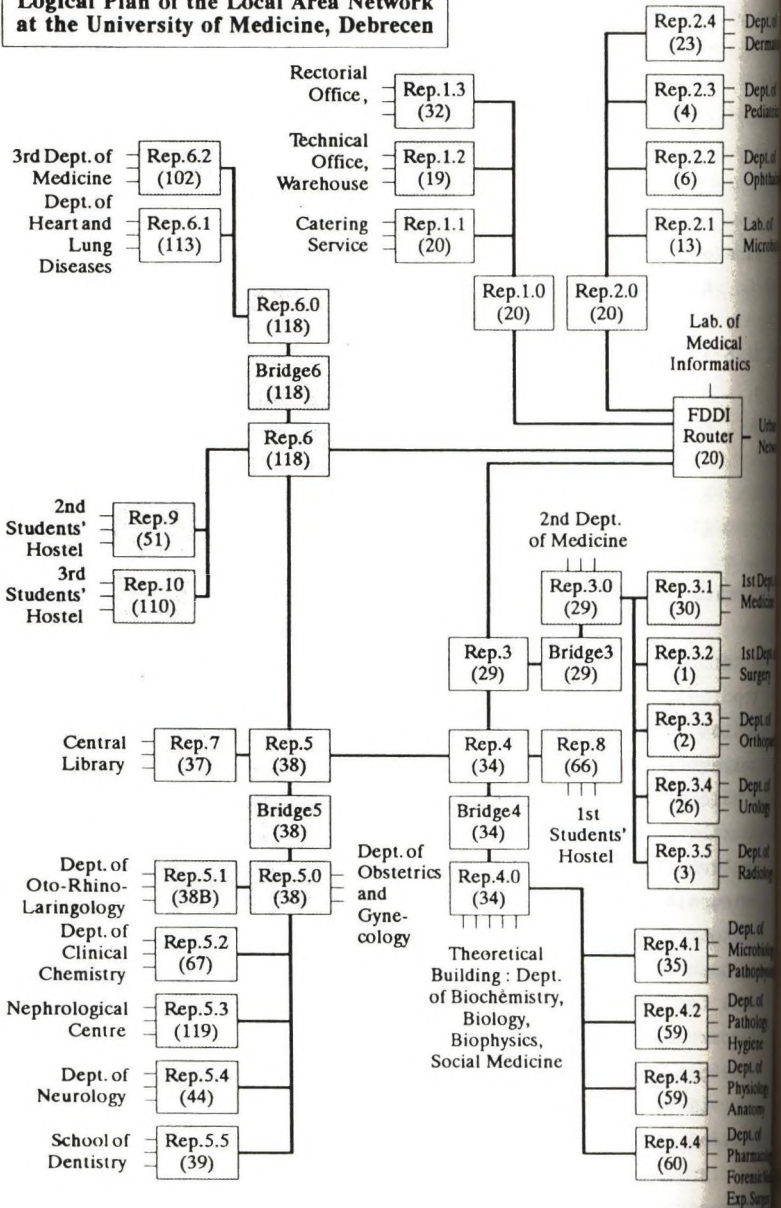
A Református Teológiai Akadémia és Nagykönyvtár épületének topológiai és fizikai sajátosságait figyelembe véve az épülő belső hálózatnak a következő feltételeket kell teljesítenie:

A hálózat az intézmény informatikai szükségletének kiszolgálására készül. Lehetővé kell tennie az intézmény már meglévő (MikroVAXII) és tervezett számítógépeinek egységes integrálását az UNIVERSITAS számítógépes hálózatába.

Biztosítani kell a könyvtári rendszer hatékony felhasználásával a könyvtári munka gyorsítását. Lehetővé kell tennie az erőforrások (printer server, file server) megosztásával az irodai munka javítását. A városi és az országos hálózatba történő bekapcsolásnak javítania kell más könyvtárakkal (UNIVERSITAS és Országos Széchenyi Könyvtár) történő adat-cserét.

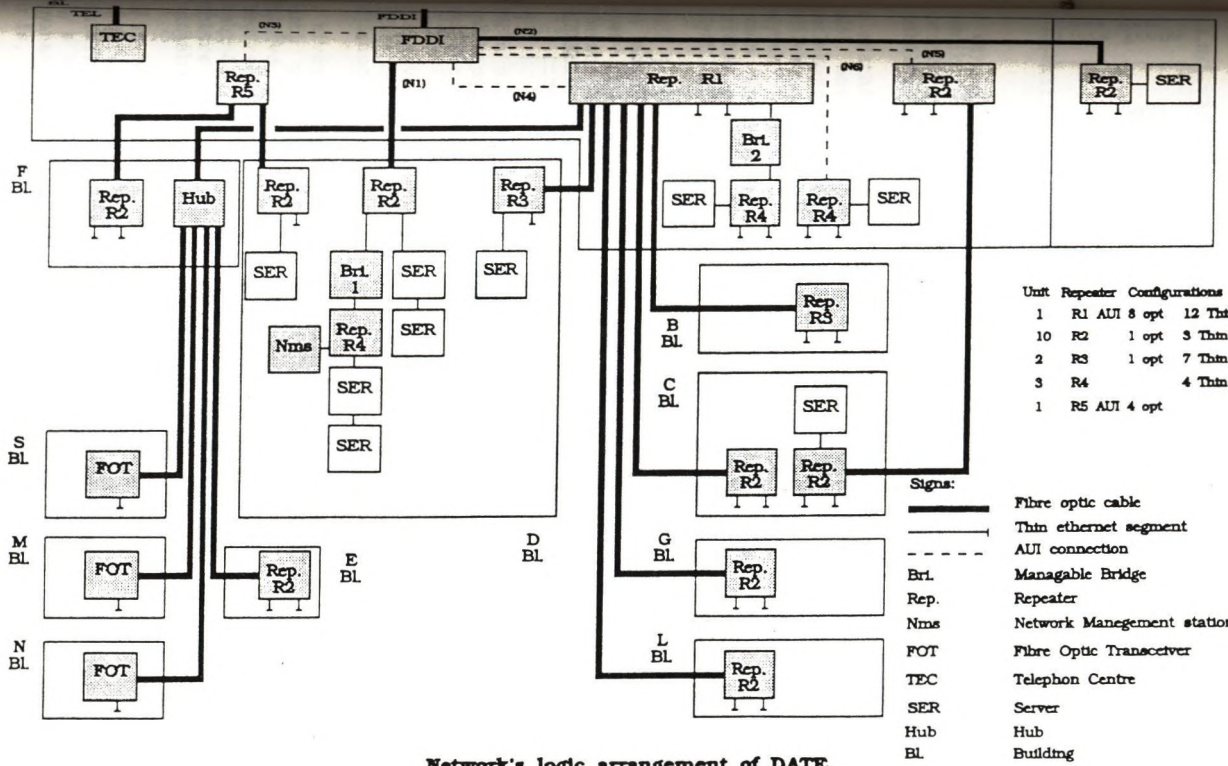
Az intézmény épülete csaknem szabályos négyzet alakú folyosóval rendelkezik. Az intézmény lokális hálózata ETHERNET csillag topológiájú, amelynek lehetővé kell tennie a városi hálózat elérését.

Logical Plan of the Local Area Network at the University of Medicine, Debrecen



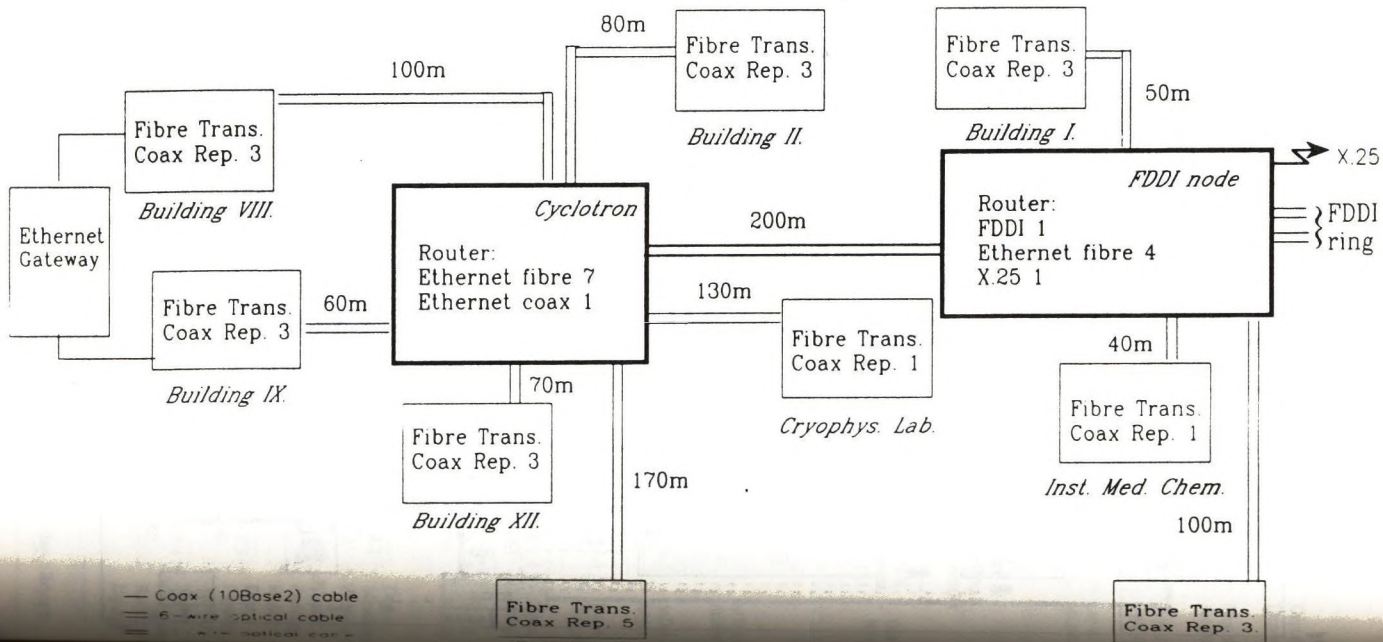
Key of the Signs

— : Optical Cable - - - : Coaxial Cable (Number) : Building Ident. No. Rep. : Repeater



Network's logic arrangement of DATE

Optical backbone of the Institute of Nuclear Research *(proposal)*



A nyilvános csomagkapcsolt adatátviteli szolgáltatás

Bérci Márton
PLEASE Adatátviteli Szolgáltató Kft.

1. Adathálózati technológiák

Az egymástól földrajzilag távol lévő, elosztott számítástechnikai rendszerek, alkalmazások számára szükséges adatátviteli technológiák között fontos szerepet kapnak az adathálózati megoldások.

Az adathálózatok közös jellemzői, hogy azokhoz a felhasználói berendezések digitális, szabványos interfészeken csatlakoznak, továbbá hogy az adathálózatokban mindig található felügyeleti rendszer (általában centralizált), melynek segítségével a hálózat üzemeltetője folyamatosan nyomon követheti a rendszer állapotát, valamint gyorsan méríthatja el, határolhatja be az esetleges hibákat, azok helyét.

Ez a közös jellemző továbbá, hogy az adathálózatokat kifejezetten adatátviteli alkalmazások számára tervezték, így az egyéb hálózatok (pl. a távbeszélő hálózat) másodlagos felhasználásához képest jobb minőségű, testreszabott megoldásokat kínálnak.

Először a vonalkapcsolás elvén alapuló úgynevezett vonalkapcsolt adathálózatok (CSPDN, Circuit Switched Public Data Network) jöttek létre, a 70-es években.

A szabványos, egységes, digitális előfizetői csatlakozási felületen (interfészen) szabványosak a hívásfelépítési eljárások.

A hívásfelépítés során a hívó végberendezés szándéka szerinti partnerrel a kapcsolat gyorsan felépül, és a hívó illetve hívott közötti átviteli út (lehet fizikai érpár, hangfrekvenciás csatorna, PCM időrés, stb. illetve ezek kombinációja) az összeköttetés teljes időtartamára rendelkezésre áll, függetlenül attól, hogy az átviteli uton valójában történik-e adatátvitel, vagy nem.

Az elvből következik, hogy az átviteli út teljes kapacitása rendelkezésre áll a kapcsolat ideje alatt, valamint hogy a késleltetési időt a jel terjedési sebessége határozza meg.

Igy a késleltetési idő kicsi és időben állandó. Ugyanakkor a vonalkapcsolás technológiájából következik az is, hogy egy fizikai összeköttetésen egy időben csak

egy kapcsolatot, kommunikáció jöhet létre, valamint az is, hogy a hívó illetve hívott végberendezéseknek azonos sebességűeknek kell lenniük.

A vonalkapcsolás elvén működő adatátviteli hálózatok nagyfokú elterjedését a csomagkapcsolás elvén működő adatátviteli hálózatok megjelenése korlátozta. A technológia az amerikai hadiipari fejlesztések eredményeképpen jött létre és a nyolcvanas években széleskörű polgári alkalmazást nyert. Ezekben az években a csomagkapcsolás szinte valamennyi fejlett országban elterjedt, napjainkra a csomagkapcsolt hálózatok a távbeszélőhálózathoz hasonlóan világméretű hálózatot alkotnak.

A nyilvános csomagkapcsolt adatátviteli hálózatokban (Packet Switched Public Data Network, PSPDN) a végberendezések a hálózathoz szabványos interfészeket csatlakozhatnak, valamint szabványosak a hívásfelépítési eljárások is.

A hívás során a hívó és a hívott végberendezés között a kapcsolat gyorsan felépül és a partnerek között egy logikai összeköttetés épül ki. Az adatátviteli fázis során az információ a hálózatban egységes méretű részekre bontva, úgynevezett csomagokban kerül továbbításra. A technológiából következően a logikai kapcsolat (virtuális áramkör) nem zárja ki azt, hogy ugyanazon a fizikai összeköttetésen egyidőben további virtuális kapcsolatok épüljenek fel. Egy-egy virtuális áramkör a fizikai összeköttetést csak akkor terheli, amikor adattovábbítás történik. A csendes időszakokban a fizikai összeköttetésen a további virtuális áramkörök által szállított adatok (csomagok) továbbítására kerülhet sor.

Az a tény, hogy egy-egy fizikai összeköttetést egyidőben több logikai kapcsolat terhelhet megteremti az átviteli utak optimális kihasználhatóságát. Ugyanakkor az eljárás maga után vonja a késleltetési idő növekedését a vonalkapcsolt technológiához képest, illetve azt, hogy a késleltetés időben nem állandó.

A csomagokra bontott információtovábbítás lehetővé teszi (a forgalomvezérlési eljárások segítségével) azt is, hogy a hálózaton egymástól eltérő sebességű végberendezések is kommunikálhassanak egymással.

További eltérést jelent a vonalkapcsolt rendszerektől az is, hogy a csomagokra bontott információ továbbítás lehetővé teszi hibavédelmi eljárás alkalmazását a hálózatban. Ennek következményeképpen a csomagkapcsolt hálózatban a hibásan vett adatok valószínűsége olyan kicsiny, hogy az a legtöbb alkalmazás számára elhanyagolható.

A csomagkapcsolás fenti tulajdonságai okozták végső soron azt, hogy a csomagkapcsolt hálózatok a rendelkezésre álló tulnyomóan analóg átviteli rendszerek világában széleskörűen elterjedtek.

Az alaphálózat minőségi változása következtében -elsősorban a fejlett országokban egyre nagyobb tért nyernek a nagy megbízhatóságú, igen alacsony bit-hibaarányal rendelkező nagysebességű és egyre alacsonyabb áru digitális összeköttetések- már napjainkban is egyre nagyobb szerepük van a bérelt áramköri adatátviteli hálózatoknak.

Ezekben a hálózatokban a szolgáltató szintén szabványos digitális interfészekben nyújt adatátviteli szolgáltatást rögzített végpontok között. A végpontok közötti kapcsolat időben változhat, a kapcsolást azonban nem hívásfelépítési eljárások, hanem a hálózat üzemeltetőjének illetve felügyeleti rendszerének beavatkozásai hozzák létre (természetesen a felhasználók kívánságai szerint).

2. A csomagkapcsolás főbb jellemzői

Felhasználói szemmel vizsgálva a csomagkapcsolás főbb jellemzői az alábbiak:

- több egyidejű összeköttetés egy fizikai áramkörön,
- hibavédelem,
- sebességillesztés,
- a vonalkapcsoláshoz illetve bérelt összeköttetésekhez képest nagyobb, időben nem állandó késleltetési idő.

Fenti jellemzők magából a csomagkapcsolási technológiából adódnak.

A csomaghálózati végberendezések és a csomagkapcsolt hálózat

közötti felületet (interfészt) a CCITT X.25 jelű ajánlása írja le. Az ajánlás a nyílt rendszerek összekapcsolására vonatkozó OSI referencia modellnek megfelelő rétegelésben (abból is az alsó három rétegre vonatkozóan) adja meg a szükséges együttműködési követelményeket.

A referencia modell rétegei hierarchikusan egymásra épülnek. A rétegek az adatokat a felettük álló rétegtől kapják, az alattuk álló rétegnek adják át és viszont. Minden réteg a vele azonos szintű réteggel kommunikál.

Az X.25 jelű ajánlásban az alsó három réteg funkciói kerültek meghatározásra.

A fizikai réteg (1. réteg)

A fizikai réteg a felette lévő adatkapcsolati rétegtől kapott információt továbbítja a vele kapcsolatban álló fizikai rétegnek az átviteli uton, közegen, illetve fordítva.

A csomagkapcsolt hálózat esetében ez tulajdonképpen a

végberendezés és a hálózat közötti interfész jellemzőinek meghatározását jelenti, azaz a csatlakozási felület mechanikai, elektromos, funkcionális illetve eljárásbelli meghatározását. Ezen paraméterek meghatározása teszi lehetővé azt, hogy a második rétegtől kapott bitsorozat az átviteli közegen keresztül a társ fizikai réteghez eljusson.

A csomagkapcsolt adathálózatban a leggyakoribb interfésztípus a V.24 (RS232C), de ezenkívül megtalálhatóak az X.21, V.35, V.36, G.703 stb. típusok is.

Az adatkapcsolati réteg (2. réteg)

Az adatkapcsolati réteg feladata a felette lévő 3. rétegtől kapott információ, adat hibamentes átvitelének biztosítása. A 2. réteg funkciója természetesen az 1. réteg szolgáltatásának igénybevételén keresztül valósul meg.

Az adatok hibamentes átvitele a második rétegek közötti procedura alapján megy végbe. Az eljárás a HDLC protokoll speciális megvalósítása, a neve LAPB. Lényege az, hogy a társ második rétegek között az adatcsere keretekben történik. A keretek a harmadik rétegtől kapott átviendő információn túl a protokoll működéséhez szükséges egyéb adatokat is tartalmazzák. Ezek a kiegészítő adatok teszik lehetővé azt, hogy az átviteli uton történt esetleges bit hibákat a protokoll kiszűrje illetve ez esetben a keret ismétlését kérje a társ második rétegről.

A protokoll lényeges paraméterei a nyugtázatlanul elküldhető keretek száma, a válasz nélküli keret ismétlési időzítése, a megengedett ismétlések száma, valamint az információs keretekben lévő adatok hossza.

A hálózati réteg (3. réteg)

A hálózati réteg végzi el a végberendezések közötti összeköttetések felépítését, bontását, az adatcsomagok átvitelének szervezését (irányítást, forgalomvezérlést) a hálózaton.

Az összeköttetések felépítése a logikai csatornák megfelelő összerendelését jelenti, természetesen mindig a társ harmadik réteggel. Egy-egy virtuális kapcsolat felépítésében a hálózatban páronként több harmadik társ szint is közreműködik, a logikai csatornák összerendelését mindig a társ harmadik szintek között kell végrehajtani.

A forgalomvezérlés teszi lehetővé a különböző sebességű végberendezések együttműködését azáltal, hogy az adó oldal csak meghatározott számú csomagot továbbíthat nyugtázás nélkül. Ugyancsak a harmadik réteg feladata az, hogy a virtuális összeköttetésen az információt hordozó adatcsomagok sorrendhelyesen érkezzenek meg rendeltetésű helyükre.

A harmadik réteg protokolljában a második rétegéhez hasonló jellegű paraméterek (ablakméret, időzítések, stb) biztosítják a zökkenőmentes együttműködést.

Csomagkapcsolás hazánkban

nyilvános csomagkapcsolt adatátviteli szolgáltatás 1989-ben indult útjára hazai fejlesztésű eszközökkel.

Az induláskor zárt felhasználói kör igényeit kielégítő, később nyilvános hálózattá fejlődött rendszer nagyarányú bővítésére a fejlett technikára vonatkozó korlátozások feloldása után kerülhetett sor.

Az 1990. október 18-án megkötött MATÁV - Siemens szerződés alapján 1991-ben megkezdődött a csomagkapcsolt adathálózat országos szintű, mennyiségi és minőségi bővítése.

A rendszer teljes megvalósítására három ütemben kerül sor.

Az első ütemben, mintegy 3.250.000 DEM értékben Budapesten, Pécsen és Debrecenben létesültek EWSP típusú központok, továbbá kiépült a hálózat központi felügyeleti rendszere is a Táviró és Adatátviteli Igazgatóság épületében, Budapesten.

Az első berendezések átvételére 1991. végén sor került

A második ütem során további 17 városban -a megyeszékhelyeken és Sopronban- kisebb központok telepítésére kerül sor. Azoknak a központoknak az átvételi eljárása napjainkban folyik.

A szolgáltatás beindítása a teljes országos hálózaton terveink szerint 1992. szeptember 1-el megkezdődik.

A második ütem -melyben a központokon kívül a budapesti távbeszélő központokban átviteltechnikai berendezéseket is telepítettünk- összértéke több, mint 5.640.000 DEM.

Az első két ütem együttesen mintegy 1200 port kapacitást (közvetlen csatlakozási lehetőséget) biztosít.

A szerződés harmadik ütemében -értéke mintegy 12.370.000 DEM- a már üzemelő központok bővítésére, valamint további három központ telepítésére kerül sor, 1993. végéig.

A bővítés három lépése együttesen 3000 port kapacitást biztosít, mely az igényeknek megfelelő ütemben kerül üzembe-elyezésre.

A teljes szerződés szerinti 21.257.000 DEM összeg 15 %-ára a Magyar Távközlési Vállalat belföldi bankhitelt vett fel. A további 85 %-ra a Siemens céghitelt nyújtott a vállalatnak.

A hitel és a kamat visszafizetése 1990-től 1999-ig tart.

A beruházás teljes tervezett bekerülési költsége -mely a szerződéskötés óta történt többszöri forint leértékelés miatt feltehetően növekedni fog - mintegy 1,4 milliárd Ft.

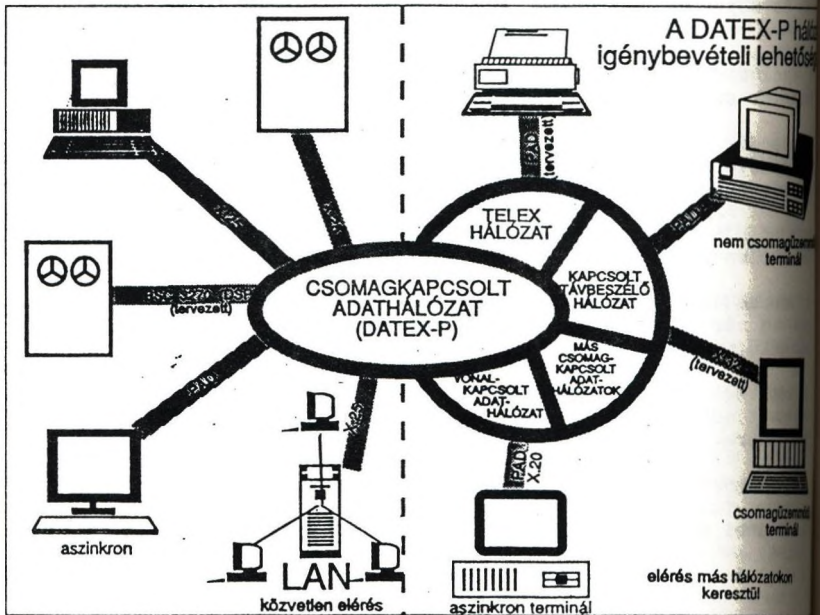
4.A szolgáltatás igénybevételi lehetőségei.

A csomagkapcsolt szolgáltatást alapvetően kétféle módon lehet igénybevenni. A 4.1. ábra szerint megkülönböztetünk két közvetlen elérést, illetve elérést más nyilvános csomagolt hálózatokon keresztül.

A két megoldás közötti fő eltérést az adja, hogy a közvetlen elérésnél a végberendezés (illetve a hozzá csatlakoztatott, a hálózat részeként értelmezett és a szolgáltató tulajdonában lévő átviteltechnikai eszköz) és a hálózat központja (Node) között dedikált, bérelt jellegű átviteli út van, míg a kapcsolt hálózatokon keresztül történő elérésnél a végberendezés (illetve a hozzá csatlakoztatott előfizetői illetve más szolgáltató tulajdonában lévő átviteltechnikai eszköz) és a csomaghálózat között a kapcsolat egy távbeszélő hálózati, vonalkapcsolt adathálózati stb. hívás felépítése után jön létre.

Az első esetben tehát egylépcsős választási eljárásról, a második esetben pedig kétlépcsős választási eljárásról beszélhetünk.

A további részletek a 4.1. ábrán találhatóak.



4.1. ábra

5.A szolgáltatás díjazási rendszere

A nyilvános csomagkapcsolt szolgáltatás díjazási rendszere valamint a csomagkapcsolt technológia között szoros kapcsolatot találhatók.

Azért, hogy azt a korábbiakban láttuk, egy-egy logikai (virtuális) összeköttetés csak akkor telheli a hálózat átviteli útjait, központjait, amikor a kapcsolaton adatátvitel történik. Az ugynevezett csendes, adatmentes időszakok a hálózat számára csak kicsiny terhelést okoznak.

Fentiekből következik, hogy a hálózat használat díja erőteljesen összefügg az átvitt információ mennyiségével.

A csomagkapcsolt hálózatban az átvitt információ mennyiségét a hálózat méri és az információ átviteli díj ennek megfelelően kerül számításra.

A hálózat az átvitt információt az ugynevezett szegmentekben méri. Egy szegment 64 oktettet (nyolc bites csoport, byte-ot) jelent. Az átvitt információ mennyisége megkezdett szegmentek szerint számlázódik.

A hálózat használati díjak között találhatók a sikeres kapcsolatfelépítések, illetve a kapcsolatok időtartam díjai is. Ezek a díjak a megfelelő alkalmazás kiválasztása esetén az átvitt információ díjához képest általában igen alacsonyak.

A hálózat használati díjakat általában a hívó fél fizeti.

A díjszabás másik eleme a havi előfizetési díj.

A közvetlenül csatlakozó felhasználók a sebességnek megfelelő havi alapdíjat fizetnek, míg a más kapcsolt hálózatokból hívók egységesen egy igen szerény díj ellenében érhetik el a hálózatot. (A ráhordó hálózat díjait természetesen fizetni kell.)

A csomagkapcsolt hálózat használati díjait valamennyi felhasználó egyformán fizeti.

A díjszabás harmadik eleme a csatlakozási díj. Ez a díj csak a közvetlenül csatlakozó felhasználókat érinti, két sebességkategóriában országosan egységes.

A csatlakozási díj tartalmazza, lefedti a bekapcsolással összefüggő valamennyi költséget.

A nyilvános csomaghálózati díjszabás fontos jellemzője, hogy a nagyforgalmu felhasználók az átvitt információ díját tekintve bizonyos mennyiségek túllépése esetén kedvezményben részesülnek, továbbá hogy a belföldi forgalomban éjszaka jelentős, 50 %-os kedvezmény lép életbe.

A nyilvános csomagkapcsolt szolgáltatás díjai mint láttuk több összetevőből állnak. Az alkalmazás jó kialakításával általában biztosítani lehet az elfogadható díjakat.



Bérci Márton

Mérnök-közgazdász, 1978-ban végzett a BME Villamosmérnöki Karának híradástechnikai tagozatán. Jelenleg a PLEASE Kft. marketing területét vezeti. Nyilvános csomagkapcsolt szolgáltatás, bérelt áramköri adatátviteli szolgáltatás, minitex szolgáltatás és üzemetkezelő szolgáltatások a főbb szakterületei.

A Kossuth Lajos Tudományegyetem informatikai rendszere

Gál Zoltán

KLTE Informatikai és Számító Központ

Általános helyzetkép

A Debreceni Kossuth Lajos Tudományegyetem egyike a legnagyobb keletmagyarországi egyetemeknek nemcsak a hallgatók, oktatók és kutatók létszáma, hanem az itt felhalmozott szellemi értékek mennyisége tekintetében is.

Mintegy közel 1400 egyetemi hallgató, 550 oktató és kutató végzi tanulmányait különböző szakokon és tudományterületeken. Informatikával a Természettudományi kar minden hallgatója közvetlen kapcsolatba kerül tanulmányai során, de a Bölcsészettudományi karról is sokan használják felhasználói szinten a létező egyetemi informatikai rendszer által biztosított alkalmazásokat. Nappali szinten évente 50 programozó és programozó matematikus szerez el a számítástechnikai igényeknek megfelelő oklevelet. Jelenlegi állapotban tekintélyes mennyiségű PC számítógépet (közel 250 darab) használnak az egyetemi dolgozók és hallgatók, amelyek a különbözőbb termekben és szobákban helyezkednek el, mint például a Főépület alagsorában található Anyagraktár vagy a harmadik emeleti Német Tanszék irodája.

Komolyabb hardver erőforrások az egyetem Informatikai és Számító Központjában (ISZK), valamint a Kémia épületben helyezkedő Elméleti Fizika Tanszéken vannak. Itt a DEC és IBM gépei találhatók: Mikro VAX II, két darab VAX 3500, VAX 3500-510 és IBM RISC 6000. Több kis kiterjedésű, viszonylag kevés felhasználós lokális hálózat is működik a különböző épületekben. Ezek főként Novell Netware típusú LAN-ok amelyek vékony Ethernet

kábellel kapcsolják össze a PC-s munkaállomásokat. Ilyen hálózatok vannak a főépületben elhelyezkedő Gazdasági Hivatalban és Matematika épület harmadik emeletén, ahol oktatási célokból három terem PC/AT típusú munkaállomásairól érhető el a saját Novell szerverek. Jelen állapotban egy másik, szerver funkciójú számítógépek csoportját azok a működő rendszerek képezik amelyek a mostani egyetemi adatátviteli hálózat segítségével egymás között is kommunikálhatnak. Ide sorolhatóak az ISZK és Nagykönyvtár Novell és UNIX szerverei. Az ISZK és főépület között egy 20 méteres vastag Ethernet szegmens van lefektetve, amelyre az egyetemi könyvtár, Pszichológia Tanszék, Szociológia Tanszék és ISZK alhálózatai kapcsolódnak.

Az országos X.25 csomagkapcsolt hálózat a PAD funkciót megvalósító WampBox-on keresztül érhető el. Ennek egyik szinkron soros portjára kapcsolódik az ISZK-ban található VAX 3500-as számítógép, szinkron csatolókártya segítségével. Az X.25 hálózat és az egyetemi lokális - nagyrészt DECnet - hálózat közötti router funkciót az említett gépen futó PSI szoftver valósítja meg. Ezt a lehetőséget leginkább elektronikus levelezésre használják fel az egyetem dolgozói. UNIX szerverek is működnek a hálózaton, így jelenleg bármely, egyetemi hálózatra kötött gépről elérhetőek a UNIX, VMS, Novell operációs rendszerek alatti szolgáltatások.

2. Célkitűzések és megvalósításuk lehetőségei

A KLTE oktatói és kutatói számára az utóbbi időben lényeges szemponttá vált egy korszerű lokális adatátviteli hálózat kiépítése, amelynek segítségével a közeljövőben megvalósul a nemzetközi információs rendszerekkel való nagysebességű kapcsolattartás. A hardver eszközök rohamos fejlődéséből adódó dinamikus változások arra a határozatra késztették az egyetem informatikai részének tervezőit, hogy a legmodernebb adatkommunikációs berendezések beszerzésére összpontosítsák figyelmüket.

A jelenlegi egyetemi hálózat kezdetleges kiépítettségi fokát figyelembe véve kihasználjuk a lényeges struktúra módosítási lehetőséget és kikerüljük azt az egy-két év múlva szükségszerűen bekövetkező kellemetlen lépésfodulatot, amit a vastag Ethernet

vezetés és az utóbbi néhány évben működtetésre került programozatlan hálózati berendezések fognak okozni, növekvő szolgáltatási igények esetén. Egy korszerű LAN kiépítésének igénye összekapcsolódik azzal a lehetőséggel, amit az 1991 utolsó évtizedében a debreceni Universitas által elnyert Világbanki támogatás nyújt. Ez nemcsak a KLTE belső hálózatának kiépítését segíti elő, hanem a város többi felsőfokú intézményei és legjelentősebb kutató intézménye (ATOMKI) számára egy nemzetközi szinten is csak most elterjedő nagy sebességű FDDI számítógépes hálózatra történő kapcsolódást is biztosít.

Utóbbiakban a KLTE belső adatátviteli hálózatára vonatkozó koncepcióról egy néhány szót.

A tervezett LAN struktúrája

A pillanatnyilag létező, épületeken belüli vékony Ethernet csatlakozások és szegmensek módosítása nélkül és az egyetlen megmaradt vastag Ethernet kábel, valamint a még használat alatt álló két multiport repeater felszámolása után a következő struktúra épül: első megközelítésre a KLTE lokális számítógép hálózata csillag/fa topológiájú lesz, amely lehetővé fogja tenni az egyes épületek nagykapacitású számítógépei (VAX-ok, könyvtári szerver, későbbi workstationok) számára a városi FDDI gyűrűre történő kapcsolódás lehetőségét is. Az egyes épületek közötti lokális kábeleket fektetünk le, amelyek minősége nemcsak Ethernet-Ethernet, hanem Ethernet-FDDI, FDDI-FDDI típusú hálózatos berendezések közötti kommunikációt is meg tudják valósítani. Ezek 50/125 típusú multimodusu üvegszálalás kábelek lesznek.

A lokális csillag/fa topológia gyökér része az ISZK-ban van. Az egyetemi hálózat az ISZK épületében elhelyezendő routingszolgáltatásával fog a városi FDDI gyűrűre kapcsolódni. A városi adatátviteli hálózat a Kémia épület déli oldalán elhelyezkedő telefon központban (TK - lásd ábra) keresztül fog belépni az egyetem területére és haladni fog tovább az ISZK-ig. A városi adatátviteli hálózat a számítógépes FDDI gyűrűn kívül egy telefon hálózatot is magába foglal. Így az egyetem telefon központjában a logikai rendező doboz fogja elosztani az egyetemi telefon és

számítógépes hálózatok számára szükséges optikai szálakat. Fizikai tagoltsága, elhelyezkedése, jelenlegi illetve várható adatforgalmát tekintve az épülő egyetemi hálózatot öt alhálózatokra volt célszerű osztani.

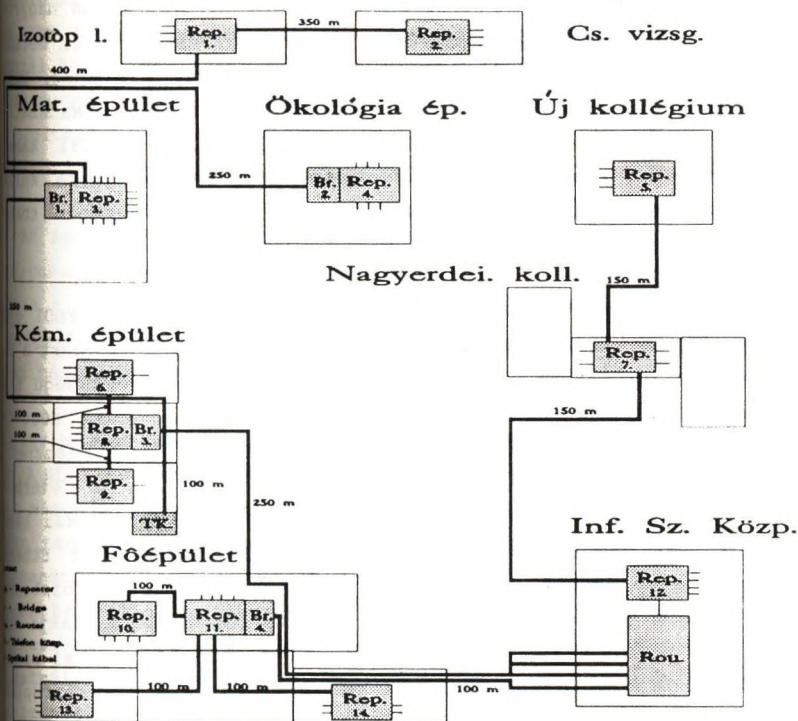
Ezek a következők (lásd ábra):

- I. Izotóp labor, Csillagvizsgáló, Matematika épület
Ökológia épület
- II. Kémia épület
- III. Nagyerdei kollégium, Új kollégium
- IV. Főépület
- V. Informatikai és Számító Központ, Zeneműtár

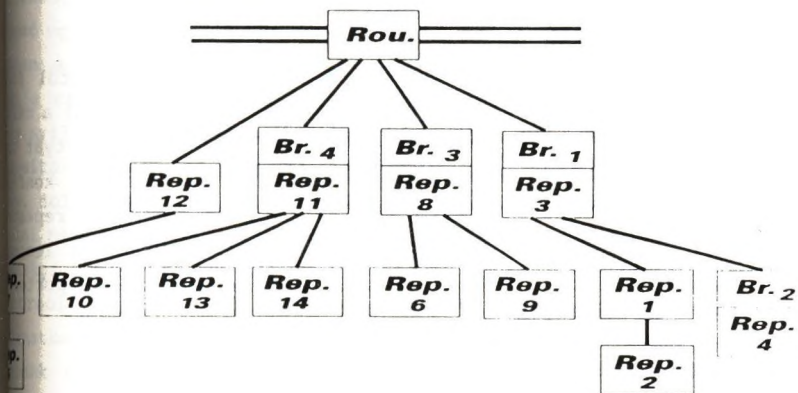
Az öt alhálózat közül négy adatforgalmát az egyetemi router fog fizikailag szeparálni. A két kollégium az ISZK alhálózatához egyik szegmensére fog kapcsolódni. Az egyes épületekben moduláris készülékek lesznek, amelyekbe úgy multiport repeater kártyák mint bridge kártyák is elhelyezhetők lesznek. Lényeges szempontnak tartottuk a készülékek programozhatóságát. Erre azért tartottuk igényt, mert a közel négyszáz állomást összekapcsoló egyetemi hálózat egyrészt a hálózatos hibadetektálás lehetőségeit, másrészt az adatforgalom szabályozását is lehetővé kell tennie. Ugyanakkor a biztonságos biztosítania kell a mesterségesen előállított szűk adatátviteli keresztmetszetek által jelentkező problémák tanulmányozhatóságát. Ilyen menedzselhető készülékek (lásd ábra) a Br1-Rep3, Br2-Rep4, Br3-Rep8, Br4-Rep11 illetve Rep12 moduláris berendezések lesznek.

A nagy épületekbe, mint amilyen a Kémia vagy a Főépület, a CSMA/CD fizikai és adatkapcsolat szintű protokollból származó Ethernet szegmenshossz korlátok miatt több multiport repeater fogunk elhelyezni. Ezek ugyancsak üvegszálalás Ethernet vonalak segítségével fognak a Br3-Rep8 illetve Br4-Rep11 készülékekhez kapcsolódni.

Egy másik fontos problémát a router típusának kiválasztása jelentette. Leglényegesebb kritérium az 'encapsulation' jelenség kizárása volt, mivel a nyílt rendszerek által biztosított tulajdonságok jóval rugalmasabbak és biztonságosabbak. Ez a várható hálózat későbbi fejlesztése céljából létfontosságú követelmény. Éppen ezért IEEE 802.1d Spanning Tree algoritmus alapján történt



Kossuth Lajos Tudományegyetem lokális hálózatának topológiai elrendezése



Kossuth Lajos Tudományegyetem lokális hálózatának logikai rajza

a lokális és városi hálózat menedzselése. Mivel a LAN nagyrészt oktatási célokat fog szolgálni, a hálózati eszközök és berendezések multiprotokollosak lesznek, biztosítva a DECnet, TCP/IP szabványok alkalmazását is. Problémát jelentett a Matematika, Kémia és Főépületet az ISZK-val összekötő optikai kábel szálszámának nyereséges kiválasztása. Itt figyelembe kellett vennünk az optikai kábel árának szálak függvényében történő logaritmikus változását.

Ezen kívül felmerült az egyetemen belül egy FDDI gyűrű kiépítésének lehetősége is. Erre azért van szükség, mert a nagy számítási kapacitással rendelkező gépeket nagysebességű hálózattal tervezzük összekötni. Ilyen számítógépek a VAX 6000-510, IBM RISC 6000 valamint további RISC workstationok lesznek. A belső FDDI gyűrű megvalósítása viszonylag egyszerűen elérhető, mivel az említett épületek között nyolc szálas optikai kábel fogunk használni. Ezeknek csak két-két szálán lesz Ethernet forgalom, a többi belső FDDI (4) valamint (2) biztonsági célokat szolgál. Ennek a gyűrűnek fizikailag csillag alakja lesz, központi részét az ISZK-ban elhelyezésre kerülő logikai rendező doboz fogja képezni. A belső és külső gyűrűk közötti kapcsolatot a router egy újabb FDDI dual attachment kártyával fogja biztosítani. A belső FDDI és az Ethernet egyidejű működése során tanulmányozni lehet a két hálózattípus hatékonyságát és ilyen célú összehasonlításokat lehet végezni.

Az egyetemi LAN a router soros portján keresztül fogja elérni az országos X.25 csomagkapcsolt hálózatot. Ebből a célból egy 64 Kbaud és egy 2 Mbaud átviteli soros illesztőkártyát tervezünk a routerbe. Nagyobb rugalmasság és fizikai konfigurálhatóság céljából az egyes hálózati berendezések (repeater, bridge, router) mellett logikai rendező dobozok is lesznek, amelyek szükség esetén az optikai szálak passzív átcsoportosíthatóságát biztosítják.

4. Az egyetemi LAN felhasználási területei

Az egyetem lokális hálózatának legjelentősebb felhasználási területét az oktatás képezi most és a jövőben is. Ilyen célokat

...gálnak a Matematika épület és az ISZK számítógépes termei, ... több tíz PC és karakteres illetve nagyfelbontású grafikus ... minál lesz. Egy külön teremben távoktatáshoz szükséges multi- ... iás munkaállomások lesznek, amelyekről ugyancsak az egyetemi ... segítségével lesz elérhető a fővárosi távoktatási centrum ... atbázisa.

Az épületek közötti és az épületeken belüli számítógépes ... kapcsolat biztosításával bármelyik lokális erőforrás elérhe- ... lesz a hallgatók számára. Így például esténként és hétvégén ... hallgatók a kollégiumból használhatják majd az egyetemi és ... rosi hálózatot.

Egy másik lényeges terület a könyvtári rendszer használata ... sz. A KLTE Nagykönyvtára a könyvek számát tekintve ország szín- ... a második helyen áll mintegy 3 millió kötettel. Ezen kívül ... den tanszéknek külön kis könyvtára van, amelyek adatainak ... zehangolása és korszerű közös használata csakis a belső számí- ... gépes hálózattal valósítható meg. Ez maga után vonja azt a ... etkezményt, hogy nemcsak a belső, hanem a városi, sőt országos ... hálózatról is lesznek könyvekkel kapcsolatos információ lekérde- ... si tranzakciók.

A következő felhasználási területet az elektronikus levele- ... s jelenti. Kétszázötvenet meghaladja az ELLA postafiókkal ren- ... kező egyetemi dolgozók száma. Fontos szerepet tölt be egyetemi ... nten az információs rendszerek és adatbázisok használata. ... ól a célból egy információs blokk kidolgozása van tervben, ... mely nyilvános, könnyen kezelhető felhasználói környezetet fog ... osítani olyan alkalmazások számára, mint az egyetemi telefon- ... gyv, dolgozók, hallgatók nyilvántartása a dékániák és rektori ... tatal részére, belső faliújság és levelező rendszer, az egyetem ... adasági hivatala számára szükséges adminisztrációs rendszer. ... k Oracle adatbáziskezelő környezetben fognak elkészülni. Jelen ... llanatban az egyetemi hálózatról fél gigabájt nagyságrendű ... tbank érhető el Pasworks segítségével, amely installálható ... amint futtatható PC-s programokat tartalmaz.

A működő egyetemi hálózat különböző operációs rendszerek

közötti információátvitelt biztosít. Így VMS és DOS között DECnet, Pasworks, DECwindows segítségével valósítjuk meg a kapcsolatot. VMS-t UNIX-al egyik oldalon CMU-TEK, másik oldalon Telnet által biztosított TCP/IP protokoll alapján kötünk össze. Hamarosan ULTRIX rendszer is lesz a VAX gépeken. Netware - DOS kapcsolat Novell 3.11-es segítségével működik az egyetem több épületében is. UNIX operációs rendszer oktatása hálózatra kötött PC-k segítségével történik, amelyeken Telnet biztosítja a terminál emulációt és fájltranszfert. MS-WINDOWS és X-WINDOWS környezeteket X11/AT segítségével kapcsolunk össze.

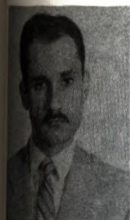
Lépések történtek az egyetem EARN csomóponttá válása érdekében, amihez VAX 6000-510-es gépünk és a SZTAKI 4341-es IBM gép közötti kapcsolatot megvalósító Jnet rendszer megrendelés alapján történő leszállítását várjuk. INTERNET domén címet igényeltünk, amellyel a nemzetközi színvonalas számítógép hálózatok információs rendszereihez fogunk tudni hozzáférni.

Összefoglalás

A Kossuth Lajos Tudományegyetem informatikai rendszerének hatékony kihasználása és további fejlesztésének szükségszerű feltétele az egyetemi adatkommunikációs hálózat lehető legkorszerűbb kiépítése, valamint az ehhez tartozó szoftverkészlet kifejlesztése. A pillanatnyi szerény eszközpark ellenére számottevő erőfeszítések történnek az egyetem országos és európai fejlett felsőfokú oktatási intézményei szintjére való felzárkozása céljából. Ennek egyik és biztos megvalósítási módja egy korszerű információs rendszer kidolgozása. Egyetemünk adottságaira és jellemzőire számos más magyarországi egyetem és főiskola saját struktúrája hasonlít, így jelen dolgozat egy gyakorlatban megvalósuló lehetséges példa ilyen problémákkal foglalkozó felsőfokú intézmények számára.

Gál Zoltán

Hardver szakos villamosmérnök, 1990-ben végezte a műszaki egyetemet. Debrecenben, a Kossuth Lajos Tudományegyetem Informatikai és Számítóközpontjában dolgozik tudományos segédmunkatársként. Adatátviteli hálózatokkal, operációs rendszerekkel és rendszerprogramozással foglalkozik.



EGYÉB ALKALMAZÁSOK ("SZ VII.")

Elnök: Havass Miklós



1940-ben született Szegeden. Egyetemi diplomát 1963-ban szerzett a Szegedi Tudományegyetem Alkalmazott matematika szakán. Végzés után a Pénzügyminisztériumban, majd a Nehézipari Minisztériumban dolgozott operációkutatási, ill. szoftverfejlesztési témában.

1972-ben csatlakozott a SZÁMALK elődintézményéhez az INFELOR-hoz, ahol a Rendszerprogramozási főosztály vezetőjeként résztvett a Videoton-gépek alapszoftverének létrehozásában. 1986-ban kinevezték a SZÁMALK vezérigazgatójának.

A NJSZT alapító tagja, jelenleg elnöke. A MTESZ alelnöke. Tagja az IEEE, ill. a Bolyai János Matematikai Társulatnak. Tudományos publikációinak száma 60 felett van. Kítüntették Neumann János, illetőleg Fényes Elek Emlékéremmel.

BLAST, AZ IGAZI KOMMUNIKÁCIÓS SZOFTVER

Unyi Gábor
COMFORT Kft.

A számítógépes adatátvitel megvalósításával sokan és régóta foglalkoznak Magyarországon is. Általában problémát okoz a különböző típusú számítógépek elektronikus összekötése, a különböző típusú modemek inkompatibilitása, a telefonvonalak hiánya, a meglévő vonalak zajossága.

Számos vállalat éppen ezért ma is postai úton továbbítja a mágnesszalagokon, vagy floppylemezeken meglévő adatait.

A szállítással kapcsolatos gondok (adathordozó megsérülése, elveszése, kései kézbesítése) a feldolgozásokat hátráltatják, esetleg lehetetlenné teszik.

Ahol nincs sem telefon sem pedig más adatátviteli közeg, ott sajnos nem lehet mit tenni, a hagyományos módon kell az adat továbbítását megoldani.

Ahol van telefonvonal, vagy más lehetőség, ott a vonalak zajosságával kell megküzdeni.

Természetesen a vonali zajokból eredő hibák elhárítása érdekében különböző hibajavítási módokat használnak az olyan protokollok, mint például az Xmodem, a KERMIT, és a BLAST. Ezek az adatfolyamatokat adatrészekre bontják, amelyek egyaránt tartalmaznak adatokat és vezérlési információt. A vezérlési információ a CRC-hez hasonló hibajavító kódokat foglal magába.

A hibajavító protokollok hiba esetén egyszerűen újra adják az adatokat. A legtöbb protokoll várakozik az átküldött adatok helyességét nyugtázó jelre. Ha hiba volt, akkor az adatokat ismét átküldi. Az Xmodem 128 bájtos fix blokkhosszt használ. A bővítések 1024 bájtos blokkokat tartalmaznak, a javítás CRC-vel történik, de a nyugtázásra várni kell. A várakozás zajos körülmények között olyan sokáig eltarthat, hogy az adatátvitel szinte megszűnik. Rosszabb esetben a vonali kapcsolat is megszakad időközben.

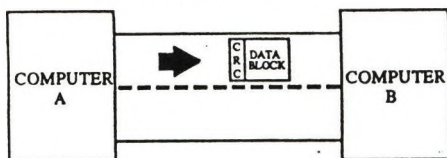
A KERMIT hasonló az Xmodemhez, azonban ennél a protokollnál a nyugtázás is védettséget kap.

A BLAST protokoll, mely minden BLAST kommunikációs szoftvertermék része, egyedülálló " csúszóablakos" protokollt hasznosít, amely teljes duplex működést tesz lehetővé, és még szélsőségesen zajos feltételek közepette is megengedi a folyamatos átvitelt.

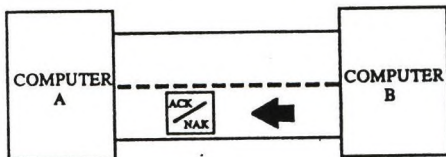
(Lásd 1. sz. ábra)

KERMIT és Xmodem fél duplex ACK/NAK protokoll

1. lépés: küldés



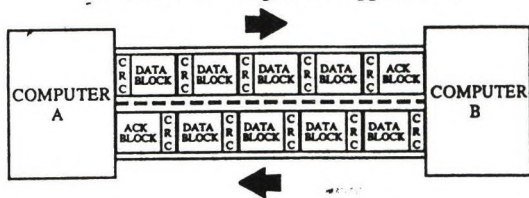
2. lépés: fogadás



ACK/NAK kommunikáció (Start/Stop)

BLAST teljes duplex protokoll

küldés és fogadás egyidőben

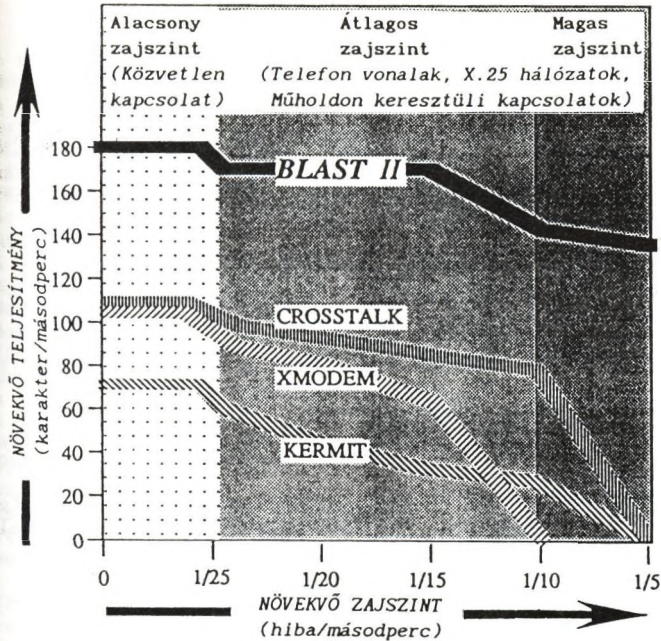


BLAST kommunikáció (Sliding Windows)

1. számú ábra

A blokkok - kiadásukkal párhuzamosan - megőrződnek egy " ablakban " (pufferben) mindaddig, amíg nem kapnak nyugtázást.

A nyugtázás folyamatosan, az adatok folyamatos adásának megfelelően (teljes duplex, kétirányú átvitel) érkezik. Hiba észlelésekor a 16 bites CRC



2. számú ábra

rendszer segítségével az adó behatárolja a blokk helyes másolatát, s meg is őrzi ezt a nyugtázás nélküli csomagok "ablakában", s elvégzi a hibás csomag újra-adását. Időközben a vevő folytatja az új adatok vételét, de addig nem ad ki adatokat, amíg a javított blokk meg nem érkezett és át nem esett az ellenőrzésen. Ez az eljárás lehetővé teszi, hogy a

BLAST nagy sebességgel adjon szélsőségesen zajos viszonyok közepette - esetleg műholdas összeköttetések át, csomagkapcsolt hálózatokon - 12 másodperces terjedési késleltetés esetén is. (Lásd 2. számú ábra)

Abban az esetben, ha az adatátvitel nehézségeit a használatban lévő modem okozza, akkor ne habozzunk, cseréljük le minőségi termékre (pl. US Robotics által gyártott termékre).

Ha különböző számítógépeket akarunk helyileg, vagy modemen keresztül összekötni gyors és megbízható adatátvitel érdekében, akkor olyan kommunikációs szoftverre van szükség, mint a BLAST, amelynek nagyon sok számítógéptípusra, illetve operációs-rendszerre meg van az adott verziója.

A BLAST KOMMUNIKÁCIÓS SZOFTVER JELLEMZŐI

- * A US Robotics 1980 óta fejleszt és készíti a különböző BLAST verziókat. Más termékek általában csak egy feladatot támogatnak, a BLAST legújabb verziói azonban integrálni képesek a terminálemulációkat, a fájlátvitelt, a távoli vezérlést és a helyi hálózati támogatást egy szoftverben.
- * A világon több, mint 70.000 referenciahelyen üzemel a BLAST valamelyik verziója. Sok világhírű cég bíz a BLAST használatában, amelyek a BLAST felhasználók ezreinek szolgáltatnak adatot (például a Nabisco, General Motors, Exxon, AT&T, Boeing, Data General,..). Mindegyik vállalat azért választotta a BLAST-ot, mert nincs másik olyan szoftver a világon, amelyik jobban tudna adatot átvinni a zajos telefonvonalakon, X.25 hálózatokon, műholdakon és nemzetközi kapcsolásokon keresztül.
- * Minden BLAST szoftver egy szabadalmazott teljes-duplex protokollt használ a 100%-osan hibamentes adatátvitel érdekében. A BLAST a legkedvezőtlenebb viszonyok között is magas teljesítményt ad. A BLAST csak az átvitel közben megsérült adatsomogokat ismétli meg. Ha a vonali kapcsolat megszakad, akkor a BLAST képes folytatni az adatátvitelt onnan, ahol a vonali szakadás volt. A BLAST protokoll az adattömörítéssel használható mikro-, mini- és nagyszámítógépes környezetben végpontok, illetve helyi hálózatok között, de megengedi az elavultabb Kermit, Xmodem, Ymodem, Ymodem-G, vagy a Zmodem protokollok alkalmazását is. Az adattömörítéssel a vonali összeköttetés névleges átviteli sebességének többszörösét tudja szolgáltatni.
- * A legújabb verziójú BLAST szoftverek magukba integrálják a PC ANSI színes, a DEC VT320/220/100/52, az IBM 3101, a WYSE 60/50, az Ampex D80, a Data General D461/411/410/200, az ADM3A és a TTY terminálemulációkat.
- * A távoli vezérlés lehetőségét is szolgáltató BLAST verziók lehetővé teszik a távoli számítógép teljes billentyűzetének és képernyőjének a használatát, amely egyszerűen alkalmazható a probléma feltárásban, szervizelésben, oktatásban. Ez a szolgáltatás gondoskodik a jelszó szerinti, illetve a visszahívást megkövetelő védelemről, a menü vezérelt paraméterezési lehetőségről és a távoli egér kezelésének lehetőségéről.
- * A legújabb BLAST verziók azonnal megjelennek, amikor a környezeti feltételek megváltoznak. Így például a QEMM, DESQview 386, 386MAX, MS DOS 5, DR DOS 5 támogatása már elkészült.

A terminál emuláció szolgáltatás segítségével a PC-ről

különböző számítógépekbe való aktív bejelentkezéseket és ottfolyó munkavégzést lehet elérni. 17 különböző terminál emulációval rendelkeznek a meglévő BLAST PC-s verziók. (Például a VT-320 terminál emulációval VAX számítógépekkel a PC-nél ülő munkatárs úgy tud dolgozni, mintha egy VT-320-as terminálja lenne.)

A fájl átvitel szolgáltatás segítségével szöveges, vagy

bináris adatokat lehet átvinni az egyik számítógépből a másikba.

A BLAST szoftverek különlegessége a BLAST protokoll használata, amely a teljes duplex módú vonalhasználatot, tömörített átvitelt, 100%-os hibamentességet, nagy átviteli sebességet, automatikus hibakorrekciót nyújt a felhasználónak.

A távoli PC vezérlésének segítségével az összeköttetésben

szereplő számítógépek egyike felkészíthető arra, hogy több, másik számítógép átvehesse a vezérlését. A bejelentkezések felhasználóhoz és jelszóhoz rendelhetőek hozzá, így az illetéktelen bejelentkezések kiszűrhetőek, letiltathatóak. A külső számítógépek bejelentkezéseit egy, a tárba ültetett szoftver figyelő és automatikusan lekezeli, emberi beavatkozást nem igényel. Ez esetben a figyelő PC-n más munka is folyhat. Ha egy bejelentkezés sikeres, akkor a paraméterezhető beállítás függvényében - a klaviatúra, a képernyő, az egér vezérlési joga megszerzhető, így oly módon tudunk az idegen (figyelő) PC-n dolgozni, mintha a sajátunk lenne.

A figyelő szoftver arra is fel van készítve, hogy adatokat is tudjunk átvinni a két számítógép között a figyelő PC munkájának felfüggesztése nélkül.

A helyi hálózat támogatás segítségével több, különböző

PC-s hálózatba tudunk aktív résztvevői lenni. Például a NOVELL hálózat egy gépe rendelkezik a BLAST szoftverrel is így az a BLAST és a NOVELL szolgáltatásait egyaránt élvezni tudja.

Abban az esetben, ha BLAST PROFESSIONAL SERVER változatát a NOVELL hálózat server gépén installáljuk, akkor a BLAST szolgáltatás a hálózat minden egyes PC-jére kiterjedhet.

A számítógépek közötti kapcsolat alapvetően két módon történhet:

- helyi összeköttetésben,
- távoli összeköttetésben.

Helyi összeköttetésben az alábbi hardver eszközök szükség-

gések:

IBM-PC kompatibilis számítógép (512KB RAM, 1.3MB winchester kapacitás, 1.2MB-os, vagy 1.44MB-os floppy, 101-es klaviatúra, monitor), soros port, és a két gép összekötéséhez RS-232 interface kábel

Távoli összeköttetésben az alábbi hardver eszközök szükség-

gések:

IBM-PC kompatibilis számítógép (512KB RAM, 1.3MB winchester kapacitás, 1.2MB-os, vagy 1.44MB-os floppy, 101-es klaviatúra, monitor), soros port, külső modem, a modem és a PC összekötéséhez RS-232 interface kábel, telefon kábel a telefon fővonalba való becsatlakozáshoz.

(A soros port, külső modem, a modem és a PC össze kötéséhez az RS-232 interface kábel nem szükséges, ha mindezek helyett modem kártyát használunk. Utóbbi esetben viszont a modem kártya esetleges tévműködése esetén az egész PC-t kell resetelni, ami nem szerencsés megoldás.)

Unyi Gábor

*A COMFORT Kft. ügyvezető igazgatója, üzememlék.
Hardver és rendszerszervező szakértő.*



ORACLE Card : Grafikus eszköz a relációs adatbázis-kezelők világában

Jakó János, Kiss Csaba Zsolt, Molnár Tamás
egyetemi hallgatók, az IQSoft Rt. munkatársai
Kossuth Lajos Tudományegyetem

Bevezetés

Az ORACLE Corp. 1991-ben jelentette meg az első, ORACLE Card nevű interaktív, programozható, grafikus fejlesztői/felhasználói környezetet biztosító programcsomagját. Előadásunk témája relációs adatbázishoz kapcsolható grafikus alkalmazások (grafikus alapú felhasználói interfészek) vagy új, a relációtól eltérő felfogású információs rendszerek építésének bemutatása a Card használatával.

1.1. Új alapszoftver megjelenése a piacon

Az ORACLE Corp. piaci sikerében valószínűleg jelentős része van azoknak az erőfeszítéseknek, amelyekben a cég minden eléggé elterjedt platformra (így hardver, mint operációs rendszer) megpróbálja termékeit adaptálni, valamint a különböző platformok közötti hordozhatóságot biztosítani. Mivel a Microsoft a PC alapszoftverek piacán a Windows révén tekintélyes részesedést szerzett, a fejlesztőknek a további sikerek érdekében ezt a piaci szegmenst sem lehetett lefedetlenül hagyniuk.

2. Új igények az adatbázis-kezelésben

A 80-as évek adatbázis-igényeit a relációs technika lényegében megoldotta, az évtized végén és különösen a 90-es években egyre nagyobb igény mutatkozott az ún. struktúrátlan információk (pl. dokumentumok, szövegek, grafikák) megfelelő kezelésére. Az ORACLE termékek között a Card volt az első, amely a relációs technika támogatása mellett a struktúrátlan információ kezelésére is figyelmet fordít.

3. A felhasználók újabb interfészigényei

A hardver-erőforrások teljesítményének fejlődésével a felhasználók interfészigényei is egyre növekedtek, és az egyre kifinomultabb igények is növekvő hatással voltak a hardver fejlődésére. Hamarosan elterjedtek a grafikus interfészek, amelyek a korábbi karakter alapúaknál az információ hatékonyabb megjelenítése mellett emberközelibbek is lettek. Az ún. Point & Click technika átalakította a személyi számítógépek, munkaállomások és terminálok világát. A Card kifejlesztésével az ORACLE Corp. egyszerre mind egy hatékony grafikus interfészt jelentetett meg mind a PC-k, mind a nagyobb adatbázisok, Client-Multiserver rendszerek mellett szolgáló terminálok piacán.

Kapcsolat a Windows-zal

1. A Windows mint környezet

Habár a Microsoft Windows különböző verziói 1985 óta forgalomban vannak, igazán csak a 3.0-ás változat 1990 májusi kibocsátása után terjedt el széles körben ez a felhasználói környezet. A korábbi verziókhöz képest az áttörést rugalmas memóriakezelés jelentette, amely segítségével a felhasználók átléphetik a DOS korlátait. Így sok alkalmazás férhet hozzá a legfeljebb 16 MB RAM-hoz és profitálhat az egyszerű task-switching-ből.

A Windows 3.X kétségkívül meglevő előnyei mellett a környezetnek megvannak a maga hátrányai is. Ahogy ezt gyakran hangsúlyozzák, a Windows legfőbb korlátja a DOS-tól való függősége. A DOS legnyilvánvalóbb hátránya pedig a 16 bites architektúra. A 16 bites címezésmóddal a DOS csak 640 K tényleges memória használatát engedélyezi. A memória-probléma megszüntetése érdekében a Microsoft ellátta a Windows 3.X-et a saját memóriakezelőivel, amelyek mind az XMS, mind a DPMS specifikációt képesek használni. Sajnos sok népszerű szoftver memóriakezelője nem fér össze a Windows 3.X memóriaelrendezésével, és így nő az egész rendszer instabilitása. Az eltérő memóriagazdálkodás vagy egyszerűen a memória felülírása okozza leggyakrabban az igen kellemetlen helyrehozhatatlan alkalmazási hibát (Unrecoverable Application Error, UAE), ami miatt gyakran nem csak az adott programot, hanem a gépet is újra kell indítani. Mivel a DOS egytaskos (egy processt futtató) operációs rendszer, ha egy program "elrontja" a memóriát, valószínűleg az egész gép leáll. Ezzel szemben a fejlettebb rendszerek mint pl. az OS/2 vagy a UNIX képesek elszigetelni a különböző processeket. Ha egy program összeomlik, az nem érinti a többi működését és a gép folytathatja a munkáját.

A Windows 3.X említett hibái és hiányosságai miatt az ezt használó cégek vezetői általában úgy döntenek, hogy csak olyan alkalmazási területeken használják a Windows-t, ahol egy esetleges hiba nem okoz komoly károkat. Nagyon kevés az olyan cég (ha van egyáltalán), ahol a komoly adatbázis-alkalmazások Windows alatt futnak. Azonban jól beváltak az olyan Windows alapú adatbázis-alkalmazások, mint pl. a döntéstámogatás, ahol egy esetleges UAE még elviselhető.

2.2. ORACLE-alkalmazások Windows-ra

Ezen tények ismeretében az ORACLE komplex fejlesztői rendszerek helyett inkább olyan eszközök készítésére koncentrál, amelyekkel a döntéstámogatás területét és a végfelhasználói piacot célozza meg. Az ORACLE for Windows és az ORACLE Card igyekszik kielégíteni mind a fejlesztők, mind a felhasználók igényeit.

ezek a szoftverek nyilvánvalóan nem sorolhatók egy csoportba az ORACLE alkalmazási szoftvereivel vagy CASE-eszközeivel. Az ORACLE for Windows alkalmaz DDE Managert, így tetszőleges, DDE-t használó Windows-alkalmazás számára elérhetővé teszi az ORACLE adatbázist. Az ORACLE Card egy komplett fejlesztői rendszert nyújt, amely tartalmaz grafikus képernyő-szerkesztőt, ábrászerkesztőt, interpretert a hozzá tartozó nyelvhez, stb. Ahogy a Windows környezet egyre inkább stabilizálódik (már amennyire ez lehetséges), az ORACLE várhatóan fogja kínálatát ezen a területen.

Az ORACLE Card

1. Az ORACLE Card részei

Az ORACLE Card több komponensből áll, főbb részei a Query Builder, Table Builder és a Stack Builder. A fejlesztőnek és a felhasználónak grafikus és szöveges szolgáltatások állnak rendelkezésére az alapvető file műveletek mellett. A funkciók a menüből illetve a toolboxból érhetők el interaktívan. Egy alkalmazásban a grafika többféle módon elkészíthető illetve módosítható. Importálható kép PCX és BMP típusú file-ból, a Clipboardról, emellett nagyszámú eszközből válogathatunk. A grafikus objektumok, például a scannelt képek az adatbázisban tárolhatók.

A Query Builder egy grafikus lekérdezés-generátor. Lehetővé teszi, hogy a felhasználók listákból való válogatással állítsák össze a lekérdezést. Tudja kezelni a mezőket, order by-okat és a logikai operátorokat is. A lekérdezés és eredménye menthető file-ba. Így a billentyűzet használata nélkül riportok készíthetők és más Windows-alkalmazásokba a Clipboardon keresztül bemásolhatók.

A Table Builder segítségével a felhasználók az SQL használata nélkül, Point and Click technikával hozhatnak létre és módosíthatnak táblákat, adhatnak hozzáférési jogokat.

A Stack Builder egy hatékony, eseményvezérelt alkalmazásgenerátor, amely lehetővé teszi a fejlesztők vagy felhasználók számára, hogy hordozható, grafikus

kliens-szerver adatbázis-alkalmazásokat írjanak. A Stack Builder segítségével csupán az egér használatával készíthető egy default képernyő a kiválasztott táblához, amelyen a tábla kívánt oszlopai megjeleníthetők és a szükséges adatbázis-műveletek elvégezhetők. Az így elkészített alkalmazások tudják kezelni a keresési feltételeket és mezőnként meg tudják jeleníteni az adatbázisban szereplő értékek listáját.

3.2. Alapfogalmak

Az ORACLE Card-dal egyedülálló vagy adatbázishoz kapcsolódó alkalmazások készíthetők. Az elkészült alkalmazások hordozhatók, változtatás nélkül átvihetők PC-ről Macintosh-ra és megfordítva. Az alkalmazások eseményvezéreltek, grafikus és képernyőn meg nem jelenő objektumokból állnak. Objektumon itt a hagyományos, OOP terminológiában megszokotthoz hasonló, attribútumokkal és eseménykezelőkkel (módszerekkel) rendelkező entitás értendő. Az adott objektumhoz tartozó eseménykezelők és függvények összességét az objektum scriptjének nevezzük. A programozáshoz (a scriptek megírásához) az ORACLE Cardban az ORACLE Talk nevű nyelv áll rendelkezésre. A nyelvhez interpreter tartozik. Az ORACLE Talk az Apple HyperTalkjából származik, támogatja a grafikus objektumokat és az SQL-t.

3.3. Kapcsolat az adatbázissal

A Card alkalmazások kapcsolódhatnak lokális, illetve az SQL*Net segítségével hálózaton lévő relációs adatbázishoz. A kapcsolatot az ORACLE Access nevű eszköz teremti meg, ami az ORACLE standard interfésze az interpreteres nyelvekhez az adatbázis elérésére. Az ORACLE Access a Hyper*SQL interfészből származik, amit eredetileg az ORACLE Macintosh-ra készült termékeihez fejlesztettek ki. Az Access segítségével a scriptekbe közvetlenül beírhatók tetszőleges SQL utasítások illetve PL/SQL blokkok.

4. A Card Objektumai

A legnagyobb egység a stack. A stack mint operációs rendszer file is megjelenik. Egy stack mindig tartalmaz legalább egy backgroundot és minden backgroundhoz tartozik legalább egy card. Ez a három fő objektum. A stacknak és a képernyőn megjelenő képe, egyetlen lényeges attribútuma a hozzá tartozó objektum. A background és a card a képernyőn is megjelenő objektumok. Ezekből mindig pontosan egy látszik, egy background és egy hozzá tartozó card. A backgroundon és a cardon helyezhető el az összes még nem tárgyalt objektum. A background a rajta lévő objektumokkal együtt közös hátteret ad a hozzá tartozó objektumoknak. A többi objektum számára vonatkozóan nincs megkötés. Hat ilyen objektumtípus van. Az első három szöveget tartalmazó objektum, ezek a field, textfield, wffield. Röviden, ezek alkalmasak a szöveges információ megjelenítésére. A többi kérésére, megadható formátummal, színnel, betűtípussal, stb. Ezek a mezők összekapcsolhatók össze az adatbázis hagyományos (karakteres, dátum, numerikus) típusú oszlopaival. A paintobject bittérképes ábrák megjelenítésére és — az importálás lehetőség alkalmazásával — bevitelére szolgál. Ebben jeleníthetők meg az adatbázis LONG RAW típusú oszlopaiban tárolt grafikák, fényképek. A drawobject egyszerű geometriai alakzatok (kör, téglalap, szabályos sokszög) megjelenítésére szolgál gazdag paraméterezési lehetőséggel. Segítségükkel diagramok, szemléltetések viszonylag könnyen készíthetők. A button megfelel a Windows környezetből is ismert nyomógombnak, de többféle stílusból választható. A gombok csak a billentyűvel kezelhetők, billentyűt nem lehet hozzájuk kötni.

5. Programozás ORACLE Cardban

Egy Card-alkalmazás elkészítése a képernyők tervezéséből, az objektumok elhelyezéséből, valamint a scriptek megírásából áll. Az eseményvezéreltségből kiindulva a fejlesztőnek nincs más programozási feladata, mint az eseménykezelő programrészletek megírása és elhelyezése a megfelelő objektumok segítségével. Az események indításáról és a kommunikáció végrehajtásáról az ORACLE Card környezet gondoskodik.

Az ORACLE Talk szintaxisa nagyon hasonló az angol nyelvre.

Például :

answer "How are you ?" with "Fine" or "Terrible"

get the last word of card field "job"

if it is "sailor" then ask "What's your name ?"

A nyelv számos beépített eseménnyel és függvénnyel támogatja a programozó munkáját, emellett elérhető az összes menüpont és a toolbox legtöbb funkciója. Ezek a beépített események mindig valamely objektumtípushoz kapcsolódnak, például a gombhoz többek között a mouseUp nevű, amelyet az objektum akkor kap meg, ha a felületén klikkentés történt. A következő példán láthatjuk, hogy a programozó ehhez az eseményhez (a gomb felengedéséhez) hogyan rendelhet tevékenységet :

Az adott button scriptjében:

```
on mouseUp
```

```
    beep
```

```
    go to card "Preferences" -- másik cardra megy
```

```
end mouseUp
```

Az események továbbadására a következő hierarchia szolgál :

egyéb objektumok -> card -> background -> stack -> library -> home

A library és a home ugyanolyan felépítésű stack, mint a többi. A library scriptjébe lehet olyan kezelőket írni, amelyek több stackből is elérhetőek. A library és a home stack objektumai nem jelennek meg a képernyőn, egyetlen hatásuk az eseménykezelő láncba való bekapcsolódás. Library-nek bármely stack kinevezhető, a home a programcsomagban szerepel, nem lehet megváltoztatni.

Ha egy esemény érkezik egy olyan objektumhoz, amely nem kezeli, akkor az a láncban a következő objektumhoz adódik tovább. Azok az események,

melyekhez a home-ban sincs kezelő, ha standard (beépített) események, egyszerűen elvesznek, ha a felhasználó által definiáltak, akkor hibaeseményt hoznak. Erre az eseményre szintén lehet kezelőt írni.

Rögzetül :

Az ORACLE Card segítségével minimális programozási munkával színvonalas adatbázis-alkalmazások készíthetők, amelyek az eddigi karakter alapúaktól eltérően tartalmazhatnak grafikus adatokat is és az információ megjelenítése mellett az adatokat ábrákkal, diagramokkal illusztrálhatják. A programcsomag eszközeivel akár kezdő felhasználók is könnyedén elkészíthetnek egyszerűbb, adatfelvitelre és lekérdezésre szolgáló alkalmazásokat.

Köszönetnyilvánítás

Hasznos tanácsaikért és segítségükért ezúton szeretnénk köszönetet mondani Dr. Juhász Istvánnak és Dobóné Balogh Juditnak, a KLTE oktatóinak és Pap Miklósnak, az IQSoft Rt. munkatársának, valamint a KLTE ISZK munkatársainak.

Felhasznált irodalom :

ORACLE Card User's Guide Version 1.0. Belmont, California, USA

ORACLE Corporation, 1991.

ORACLE Card Programmer's Guide Version 1.0. Belmont, California, USA

ORACLE Corporation, 1991.

Matthew R. Miller : ORACLE Solutions for the Windows 3 Environment.

ORACLE Corporation, 1992.

A szerzők a Kossuth Lajos Tudományegyetem harmadéves programtervező-matematikus hallgatói. 1991. decembere óta az IQSoft Rt. külső munkatársai. Az ORACLE Carddal 1992. januárja óta foglalkoznak, első Card alkalmazásuk egy SQL alapú grafikus információs rendszer, amely a Parlament számára készült. A rendszert az IQSoft az IFABO'92-n kiállította.



Kiss Csaba Zsolt



Molnár Tamás

**Mentőállomások szimulációja SIMULA 67 nyelven,
Clipper adatfordítóval**

Hargitai Róbert
ELTE TTK

Bevezetés

A Norvég Számítóközpontban 1980 és 82 között kifejlesztettek egy portábilis technológiát SIMULA-fordítók gyártására. Az első, ezzel a technológiával előállított fordítót készítő magyar team vezetője, Gáspár A., továbbfejlesztette a módszert, és 1986-ban — a világon elsőként — IBM PC/XT/AT gépeken működő SIMULA-fordítót írt. Egy évvel később e fordító felhasználásával Pap Z. dr.—ral és Moro Á.—val megvizsgálta a SIMULA 67 bázisú szimuláció alkalmazási lehetőségeit az Országos Mentőszolgálatnál. Elkészítettek egy SIMULA-programot, amely egy mentőállomás működését modellezi. A szimuláció célja az volt, hogy a tömegkiszolgáló rendszerek vizsgálatánál szokásos eljárások (kapacitásbecslés, megbízhatóság- analízis, szűk keresztmetszetek kimutatása, a működési paraméterek optimálisra hangolása stb.) segítségével feltárják és elemezzék a Mentőszolgálat fejlesztésének lehetőségeit működési szabályok kidolgozása, tömeges balesetek során a mentés megszervezése, valamint erőforrások (pl. mentőautók) elosztása terén.

Az ELTE TTK Általános Technika Tanszékén készülõ szakdolgozatomban azért foglalkozom ezzel a témával, hogy kialakítsam egy technika tanterv-modul körvonalait, amelynek a mentésügy áll a középpontjában. Írtam egy Clipper nyelvű adatfordító programot, amely a fent említett SIMULA—programnak egyfajta környezetet teremt: lehetővé teszi a különféle mentőállomások adatainak dBase adatbázisokba való kényelmes bevitelét, karbantartását, és a szimulációs eredmények dBase adatbázisának eltérő szempontok szerinti megjelenítését.

1. A mentőállomás működése

Az Országos Mentőszolgálat alapvető szervezeti és működési egysége a mentőállomás. Minden mentőállomáson a nap minden órájában ügyeletet lát el egy-egy szolgálatvezető (diszpécser). A sürgősségi betegellátásra és betegszállításra vonatkozó igények általában telefonon érkeznek a mentőállomásra. A szolgálatvezető a mentőegységeket rádióon irányítja. A befutó igényeket a bejelentés pillanatában menetleltre rögzíti, és meghatározza prioritási osztályukat. A szolgálatvezető íróasztalán a végrehajtásra váró feladatok menetleveleiből tehát várakozási sorok képződnek.

A legmagasabb prioritású kiszolgálást igénylő kategória az azonnali, ide tartozik az akut életveszély állapotában lévő betegek ellátási igénye. A következő kategória a három órán belül ellátandó igényeké, melybe az első szakellátásban már részesült, nem az akut életveszély állapotában lévő, de sürgős ellátást igénylő esetek tartoznak. Amennyiben egy ilyen igény kielégítését 3 óra alatt az adott mentőállomás nem tudja megkezdeni, a feladat automatikusan átsorolódik a legmagasabb prioritású kategóriába. Minden további eset a harmadik kategóriába tartozik. Ilyen például az arra rászoruló beteg hazaszállítása is.

Az igények érkezési folyamatát egy-egy mentőállomás körzetében számos tényező befolyásolja:

- az egészségügyi és kommunális infrastruktúra fejlettségi foka,
- a terület jellemző gazdasági szerkezete (mezőgazdasági vagy ipari),
- népsűrűség,
- korösszetétel,
- az idegenforgalom szezonális hatása stb.

A mentési feladatok végrehajtását jelentős mértékben befolyásolja az egészségügyi infrastruktúra kiépítettsége (pl. kórházak távolsága stb.).

A mentőellátásban előfordul olyan eset, amikor egynél több mentőegység vesz részt a mentési feladat végrehajtásában (pl. mentőautó - repülőgép - mentőautó). Jellemzőes működési sajátosság a többszintű mentés. Ez azt jelenti,

hogy amennyiben nincs lehetőség megfelelő szintű ellátásra (pl. nagyobb balesetnél nincs elég szabad rohamkocsi a körzetben), alacsonyabb szintű egységeket mozgósítanak (pl. esetkocsit), hogy addig is konzerválják az állapotot, amíg az alkalmas segítség meg nem érkezik. Megtörténhet, hogy valamely tevékenységet meg kell szakítani, pl. a szállító autó köteles megállni, ha véletlenül egy baleset színhelyére ér.

A felelősségteljes munka alapos hely- és emberismeretet követel a diszpécserőtől, akit egy rádiós irányító és egy segéd állandóan támogat tevékenységében. Mindig tudnia kell, hol tartózkodnak a mentőautói, és milyen feladatot bízhat bennük űlő embereire.

A mentőállomás működése a tipikus napi feladatok megoldása során első közelítésben egy klasszikus tömegkiszolgáló rendszernek megfelelően modellezhető.

2. A szimulátor

A program indulásakor egy standard (ASCII formátumú) szövegfájlból olvassa be a szimulálni kívánt mentőállomás adatait: a mentőautók számát, típusát (roham, eset, one = orvos nélküli, szállító) és napi munkabeosztását (az egyszerűség kedvéért feltételezzük, hogy ez minden nap egyforma); a szimuláció időtartamát (a szimulációban az időt mindig valós számként ábrázoljuk, óra.századóra formában); valamint a három betegtípus (azonnali, háromórás, alkalmi) mindegyikénél a két bejelentés között átlagosan eltelt időt, a minimális és maximális szállítási időket, továbbá a bejelentéseket és a szállítási időket "gyártó" véletlenszám—generátorok kezdőértékeit.

A kapott adatok alapján egy 'inputgenerátor' nevű objektum állítja elő véletlenszerűen a bejelentéseket, és — ezzel párhuzamosan — a 'beteg' objektumokat. A mentőállomás működését egy 'diszpécser' nevű objektum irányítja: a bejelentések alapján — meghatározott szabályok szerint — útnak indítja a megfelelő típusú 'mentő' objektumot, amely felveszi, majd — ugyancsak véletlenszerűen meghatározott szállítási idő elteltével — lerakja a beteget.

Az objektumok üzeneteket küldenek pillanatnyi állapotukról, amint abban

változás áll be (a SIMULA 67—ben a szimuláció alapvetően eseményvezérelt). Ezek az üzenetek a képernyőn jelennek meg, illetve szükség szerint a DOS input/output—átírányító funkciójával egy standard szövegfájlba küldhetők.

A szimulátoron csak minimális változtatásokat hajtottam végre, melyek közül a legfontosabbak: a szimuláció többnapos időtartammal is lejátszható; a mentők a szállítás kezdetén azt is "jelentik", hogy melyik beteget (vagy betegeket) vették föl. Munkám nagyobb részét annak az adatfordítónak a megírása jelentette, amely a különböző adatokkal elindított szimulációk futtatását és a szimulációs eredmények megjelenítését teszi kényelmesebbé, s így mintegy "adminisztrálja" a szimulátor működését.

3. Clipper adatfordító

Az adatfordító program — melyet akár a szimulátor "titkárnőjének" is nevezhetnénk — menüvezérelt rendszerű, könnyen kezelhető. Az adatokon belül négy szintet különböztet meg: a mentőállomás, a gépkocsiállomány (autópark), az ütemezés és a szimuláció szintjét. A mentőállomások szintjének adatait egy, a másik háromét pedig két—két adatfájl tartalmazza. Külön fájlban tároljuk az állomások nevét, így az adatbázis összesen nyolc fájlból áll. Az utolsó két szint összetartozik, közös azonosító kóddal hivatkozhatunk rájuk.

- a) A mentőállomás szintje tartalmazza a három betegtípus véletlenszerű érkezését és szállítási idejét befolyásoló paramétereket, valamint az állomás nevét. Az azonosító kód egy maximum négybetűs "névkód"—ból és egy év, hó, nap alakú dátumból áll. Itt tehát azt feltételeztem, hogy az egyes állomások adatai — a nyilvánvaló helyspecifikus tulajdonságok mellett — az időbeli jellegzetességek alapján különböztethetők meg legjobban egymástól. Ugyanannak a mentőállomásnak a paraméterei másként alakulnak például a pápalátogatáskor, egy tömeges baleset alkalmával, egy nyári vagy egy téli napon stb.

- b) A következő szint az autóparké. Ez mindössze a mentőautók típusának kódját tartalmazza (R=roham, E=eset, O=one, S=szállító). A gépkocsikra a rendszámukkal hivatkozunk, eltérően a szimulátortól, amely sorszámokkal látja el őket. Az autópark azonosítója egyedül a névkód, vagyis feltételezésem szerint az állomások gépkocsiállománya időben nem változik számottevő mértékben (bár a régi Nysa típusú autók lecserélése Mercedesekre már folyamatban van, és talán nem alaptalan az állomány bővülése iránti reményünk sem; ezt a pontatlanságot azonban a harmadik szint segítségével kiküszöbölhetjük).
- c) Az ütemezések szintjén található a mentőautók munkarendje (szolgálatuk kezdete és időtartama óra, perc formában). Nem szükséges, hogy egy—egy ütemezés tartalmazza az állomás összes autóját; ott a maximumot adhatjuk meg, amely általában a valóságban is több, mint ami nap mint nap elindul. Ezen a szinten tehát — melyet a névkódon és a dátumon kívül az ütemezés sorszámával azonosítunk — azzal kísérletezhetünk, hogy mi történik, ha a betegellátásban résztvevő mentőautók számát és szolgálati rendjét változtatjuk.
- d) A negyedik szinten található a szimuláció időtartama, továbbá a szimulációs eredmények, melyeket az ütemezés (áttételesen az első két szintet is magában foglalva) az időtartammal együtt már teljes mértékben meghatároz. Ezért az eredmények azonosítója megegyezik a megfelelő ütemezés kódjával. Maguk az eredmények nem mások, mint a szimuláció szereplőinek — a mentőknek és a betegeknek — az üzenetei, melyek "jelentést" küldenek az állapotukban bekövetkezett változásokról. Minden jelentés az alábbiak szerint épül föl:
- szimulációs időpont (nap, óra, perc, másodperc)
 - autó rendszáma
 - típus kódja
 - beteg sorszáma
 - üzenet

Az autó és a beteg azonosítója közül az egyik el is maradhat (pl. a betegbejelentéskor, vagy ha az autó "szabad"—ot jelent). A típuskód mindig arra vonatkozik, akié az üzenet; hogy ez autó—e vagy beteg, az az üzeneten kívül magából a kódból is kiderül (betegek esetén a = azonnali, h = háromórás, l = alkalmi). Az üzenetek betegnél: új beteg, szállításfelvétel, átadási helyen; míg autónál: szolgálat kezdete, szabad, végrehajtás kezdete, végrehajtás vége, szolgálat vége.

A 2., 3. és 4. szint esetében külön fájlban helyezkednek el az egyes csoportokat (autópark, ütemezés, eredmény) azonosító kódok, és a hozzájuk tartozó elemek (autók, üzenetek). Egy csoporton belül az összetartozó elemek rekordjai (pl. a "Bp04" kódú autópark autóinak típusai) a megfelelő elemfájlban fizikailag egymás mellett található. Az együvé tartozó elemhalmazok — fizikai — sorrendjének a halmazok azonosítóinak — szintén fizikai — sorrendje felel meg a hozzá kapcsolódó csoportfájlban. (Ha pl. a Bp04 kód az autóparkok fájljában a 3. rekord, akkor a hozzá tartozó autók halmaza az autók fájljában fizikailag a harmadik halmaz). Az egyes elemhalmazokat a fájlban belül speciális tartalmú jelzőrekordok választják el egymástól. Erre a módszerre azért volt szükség, mert az egyes halmazokhoz tartozó elemek száma jelentősen eltérhet, így nagyon gazdaságtalan lenne mindig állandó nagyságú — emiatt szükségképpen az előzetesen megbecsült maximumnak megfelelő — helyet lefoglalni.

Jelenleg az adatbevitel során lehetőség van az egyes szintek adatcsoportjainak kollektív másolására, törlésére és áttekintésére, valamint az adatok módosítására.

A szimulátort a kiválasztott ütemezéssel maga a program indítja el, miután legenerálta a futáshoz szükséges (a megfelelő helyről kikeresett adatokat tartalmazó) szövegfájl. Az eredményt azonban egyelőre csak az üzenetek sorozataként tudjuk megjeleníteni. Még dolgozom az autók és a betegek adott időpontbeli állapotának összefoglaló áttekintésén.

4. Néhány futási eredmény

A szimulátort kísérletképpen lefuttattam néhány baranyai mentőállomás adataival. A bejelentések gyakoriságát és a szállítási időket meghatározó paramétereket azonban csak becslések alapján állítottam be, így az eredményekből nem szabad messzemenő konkrét következtetéseket levonni.

Ennek ellenére az eredmény ígéretesnek mondható; a szimulátoron — legalábbis elméletileg — tanulmányozhatók olyan jelenségek, mint a mentőautók számában, típusában, munkarendjében, vagy a bejelentések gyakoriságában és fajtánkénti eloszlásában bekövetkezett változások hatása.

5. A továbbfejlesztés lehetőségei

A szimulációs eredményeket rendkívül sokféle szempontból lehetne megjeleníteni, figyelembe véve a mentőellátással szemben támasztott speciális követelményeket és igényeket.

A gyakorlatban való alkalmazást elősegítendő, meg lehetne írni a szimulátornak egy újabb változatát, amely nem valószínűségi változókkal, hanem regisztrált adatokkal dolgozik, így vizsgálható lenne például egy—egy súlyosabb baleset, vagy egyéb szempontból nehéz, bonyolult helyzet ellátásának minél jobb megoldása.

Többfajta szimulátorverzió kidolgozásával lehetővé válna az eltérő működési szabályok hatásának vizsgálata. Megszámozott programokkal azt is el lehetne érni, hogy az adatfordítóból választhassuk ki az éppen elindítani kívánt szimulátort.

Tömeges balesetnél — helyenként más esetben is — a körzet mentőállomásainak együttműködésére van szükség. Ez teszi indokoltá, hogy olyan modellt is kialakítsunk, melyben egy helyett több állomás szerepel (pl. egy megye állomásai).

A szimulációs modell a valósághoz képest jelentős egyszerűsítéseket tartalmaz. Például a végrehajtásban megszakítás következhet be — nemcsak

olyankor, ha a szállító autó menet közben balesethez ér, hanem akár a diszpécser is megváltoztathatja korábbi döntését (súlyosabb eset jött közbe). A háromórás és az alkalmi betegek szállításánál (amikor egyszerre több beteget visz az autó) gyakran előfordul, hogy a betegek nem egy helyen szállnak föl és le; és még hosszan sorolhatnánk az egyedi részleteket.

Mindezekre tekintettel lenni nem kis feladat, azonban a modell használhatósága nagymértékben függ a modellezési technika kifinomultságától.



Hargitai Róbert

1967. október 30-án született. Itt végezte az általános iskolát, majd a Piarista Gimnáziumot. Jelenleg az ELTE TTK matematika-technika tanári szakának ötödéves hallgatója. Emellett haramdik éve tanul az ELTE angol szakfordító kurzusán. Idén kezdte tanulni a számítástechnikát. Szeptembertől a Neumann János Közgazdasági Szakközépiskolában fog programozást tanítani.

PELLER RÓBERT

SZÁMALK

Számítástechnika Alkalmazási Vállalat

AZ INFORMÁCIÓ TECHNOLÓGIA FEJLŐDÉSÉNEK NÉHÁNY SZEMPONTJA
SZUBJEKTIV KÖZELÍTÉSBN

Összefoglalás: Az előadás az információ technológia lehetőségeinek egy nem szokásos megítélését kísérli meg. Feltételezi, hogy a társadalmi környezet a korábbiaknál nagyobb figyelmet érdemel az információs rendszerek létrehozásában, biztonságos fenntartásában.

A következőkben információ technológia /IT/ alatt az információ kezelés módját, lehetőségeit értjük.

Ebben a közelítésben különösebb magyarázatot nem kíván, hogy az jelentős mértékben függ a mindenkori

- műszaki,
- gazdasági,
- társadalmi /emberi/

környezettől.

Az IT tehát függ a társadalmi, emberi megnyilvánulás minőségétől, vagy is a kulturától, ezért nagy jelentőségű, hogy az összefüggésekre a korábbiaknál nagyobb figyelmet fordítsunk. A körülöttünk történő események számos jelensége arra utal, hogy az információ kezelés módja, lehetőségei nem kélőkép kerülnek figyelembe vételre, akár csak az utóbbi időszak történéseinek irányításában, feldolgozásában. Anélkül, hogy részletesen kitérnénk most a jogszabály alkotás, vagy

a gazdaság irányítás számos megnyilvánulásával összefüggő, az információ kezelés módjának zavarára utaló jelenségre, azt hiszem, a napi sajtó alapján erre mindannyian számos példát olvashatunk. Mi e jelenségek alapján is elfogadjuk az IT kultúra függőségét, ez esetben a történelmi előzmények, a társadalmi háttér figyelembevétele alapvető összefüggésekre mutathat rá.

Itt Magyarország, de szélesebb értelemben az egész Kárpát-medence sajátos helyzete nem megkerülhető.

E szerint az itt élő népeknek

- történelmük során mérlegelniük kellett a latin-keresztény és a bizánci; későbbiekben a német és ottomán; majd a modern korban a nyugat-európai és a szovjet befolyás között,
- a fentiek eredményeként sajátos kulturális sokszínűség alakult ki, amely az itt élőkben sajátos képességet alakított ki a környezet elfogadására, illetve kompromisszumok megkötésére,
- ugyanakkor sokszüségének az a következménye, hogy sajátos bizonytalanság volt a történelem folyamán a területi, nemzeti identitást illetően. Ez viszont bizonyos értelemben a környezet elfogadásának, illetve a kompromisszumok keresésében nehézségeket, surlódásokat jelent.

Magyarországon, de a térség, az egész kelet- közép-európai régió jelenleg olyan átalakuláson megy keresztül, amelyben ez a sajátos kettősség érvényesülni látszik.

Példaként ismét a napi, politikai események szolgálhatnak, amelyek egyes esetekben kompromisszumok elfogadásában, illetve az ésszerűtlenségig menő kompromisszum képtelenségben nyilvánul meg.

Az a változás a társadalmi berendezkedés minden elemére kiterjed, így:

a politikai rendszerváltásra,
a gazdasági környezet/struktúra megváltoztatására,
a társadalmi berendezkedés újra alakítására.

változások tartalmát, lehetőségeit láthatóan nagymé-
rűben befolyásolják a kulturális hagyományok, ezen keresz-
tül a társadalom által elfogadott információs strukturák és
tartalmak.

Mindezeket átgondoljuk a régió számos politikai, illetve
gazdasági jelentése új megvilágításba kerül.

Mult hagyatéka a változások végrehajtásához nem túl biz-
tosító:

- az infrastruktúra, különösen a telekommunikációs rendszerek,
a közlekedés,

- az oktatás-képzés lehetőségei,

- a közügyek intézésének kulturáltsága

igen szegényes. Ennek következménye, hogy hasonlóképpen

jeletlen a minket közelebről érintő számítástechnikai szol-
galtatások színvonala.

Maradott és alacsony értékű a számítógép állomány, kezdetle-
sen és széttöredezett az információ technológia, vagyis ebben
az állapotban is tükrözi a társadalom milyenségét; az informá-
ció kapcsolatok korábbi felépítését, illetve ezzel összefüg-
gésben az ennek megfelelő technológiai színvonalat képviseli.

A problémát súlyosbitja, hogy az emberi erőforrások is a kör-
nyezetnek megfelelően szegényesek, céljaikban rövidlátóak.

Azt jelenti, hogy ezek együttese nem sok jót kínál az in-
formáció technológiák közeljövőbeli fejlődését, illetve be-
fogadását illetően, már csak azért sem, mert a fentiek alap-
vetően befolyásolják, hogy

mi módon gyűjthető,

- mi módon dolgozható fel, tárolható, vagy archiválható,
- miként osztható el, illetve miként használható az információ,
- továbbá milyen minőségű (egyáltalán milyen minőségű lehet) az ilyen körülmények között kezelt információ.

Az információ technológia szélesebb összefüggéséből az is következik, hogy a számítógépesítés, illetve a korszerű telekommunikáció lehetőségeinek alkalmazása nélkül is léteznek. Az eddig elmondottak alapján az a következtetés is körvonalazható, hogy a környezet jelentősen befolyásolja, így a politikai, gazdasági szerkezet, valamint az az érték is, amelyet az információnak az érintett személyek és/vagy közösségek adott környezetben tulajdonítanak.

Az eddigiek alapján egy fontos megállapítás tehető:

Az információ technológiának a hatékonysága kiemelkedő módon függ az emberi erőforrás és a közösség integrációjától. Ebből a szakmánk szempontjából további fontos következmény is adódik, nevezetesen, hogy a fejlett számítógépes erőforrások alkalmazása nem elégséges, sőt, nem feltétlenül szükséges feltétele jól működő információs rendszernek.

Következmény az is, hogy az információ technológia, amely egy adott gazdasági, társadalmi környezetbe ágyazott, nem lehet lényegesen jobb, magasabb színvonalú átlagában annál. Ennek következtében a fejlődés meglehetősen lassú, illetve lassú kell legyen, mert az emberi lehetőségek változása korlátozott ütemű. Különösen figyelemre méltó ez, hiszen a számítástechnika, illetve a kommunikációs technika különösen gyorsan fejlődő, illetve fejlesztett terület az utóbbi évtizedekben. Tehát reális a veszélye annak, hogy a technikai fejlődés elszakad a társadalom indokolt lehető-

égeitől, és ebből a szakmát érintő számos konfliktus hely-
et adódik. Különösen akkor, ha figyelembe vesszük azokat
korábbi ígéreteket, amelyeket e szakma sem teljesíthetett,
eljesített a korai időszak eufóriája alapján a hatvanas-het-
venes években elképzeltékből.

rt hiszem, hogy az eddig elhangzottak alapján néhány figyel-
ztetés és ajánlás megtétele segítheti a közel jövőben mun-
kat.

tek:

hatékony IT kiépítése nagy mértékben függ az egyének és kö-
zösségek nyitottságától,

az IT fejlesztéséhez nem elegendő a számítástechnikai, illet-
ve telekommunikációs erőforrások fejlesztése, bár kétség ki-
vül ez ami egyszerűbbnek tűnik,

a jogi, gazdasági, társadalmi vezetésnek figyelembe kell ven-
ni az információ technológiákat, nemcsak fordítva.

ennek elhanyagolása hatalmas költségeket okozhat a társadalom-
nak, de különösen érzékenyen reagálhat rá a gazdaság,

ez azt is jelenti tehát, hogy e területen sem hagyható figyel-
men kívül a költség/haszon /előny/ vizsgálása,

a nagy rendszerek megértése, kiépítése, felügyelete nem magá-
tól érthető, ezzel összemérhető nagyságu a működtetéshez
szükséges információ rendszerek használatba tétele,

a társadalmi életnek minden területén az információs kapcso-
latok állandó vizsgálata szükséges és nem helyettesíthető
akció-szerű nekilendülésekkel,

nehéz a döntés, hogy mit kell és/vagy lehet, illetve érdemes
számítógépesíteni, amennyiben az információ értéke nem kellő-
képpen tudatosul. Az információ technológiában is alapvető
fontosságú a minőség orientáltság.

CHINOIN

P45KONG.DOC

Előadás tervezet az 1992. évi Neumann kongresszusra

A NEMZETKÖZI ÉLVONALHOZ VALÓ FELZÁRKÓZÁS ESÉLYEI (Irta: Homonnay Gábor)

A PRIVATIZÁLÁS

A privatizálás során a Chinoiban 40%-os részesedést vásárolt a SANOFI. Egyben opciót kapott a többségi részesedésre 1994-től. Néhány mondat a SANOFI-ról.

A CHINOIN

Néhány szó a Chinoinról és a számítástechnikai eredményekről. A 70-es évek kötegelte, integrált rendszerei. A mikroszámítógépek térhódítása a 80-as években.

FEJLESZTÉSEK KEMÉNY KORLÁTOK KÖZÖTT

A 70-es évek második felétől a továbblépés komoly akadályba ütközött. Még 1989-ben is csak egy csökkentett teljesítményű nagyszámítógépet vásárolhatott a Chinoin. A jelentős eszközhiányok hatásait vázolja fel az előadás e része.

A CÉLKITŰZÉSEK

A Chinoin korábbi célja és a SANOFI elképzelései egybe vágnak: a Chinoinból világszínvonalú vállalatot kell csinálni.

AZ ÁLTALÁNOS FEJLESZTÉS KÉT VONALA

A felzárkózásnak láthatóan két vonulata van:

- a technológiai (logisztikai, gyártási, gyártásellenőrzési) fejlesztés és egyszerűsítés tennivalói,
- az irányítási, szabályozási, informatikai fejlesztések teendői.

Az előadás e része az informatikai fejlesztések körével, szerepével foglalkozik.

A NEMZETKÖZISÉG ELŐNYEI

A közvéleményben él egyfajta kép a nemzetközi cégek előnyeiről. Hogyan érzékelhető ez belülről? Ezt kívánja megvilágítani e rész.

AZ ALKALMAZKODÁS TERÜLETEI

Az új munkastilushoz alkalmazkodni kell. Ez az alkalmazkodás sokszor erőfeszítésbe kerül. Milyen tennivalók vannak e területen?

A CHINOIN ESÉLYEI

Végezetül a Chinoin (és lehetségesen a változni képes magyar nagyvállalatok) esélyeit foglalja össze az előadó.

Budapest, 1992. február 8.

Homonnay Gábor

Fazekas Béla:

A BasisPlus és alkalmazásai

BasisPlus

A BasisPlus egy gyors, szöveges és relációs adatbáziskezelő rendszer melyhez komplett könyvtári alkalmazás (TechlibPlus) is van.

A BasisPlus/TechlibPlus rendszert az IDI (Information Dimensions Inc. – a nagyméretű szöveges információs rendszerek piacának vezető vállalata.) fejlesztette. Az IDI a Batelle leányvállalata. A Batelle Memorial Institute a világ egyik legnagyobb független kutatóintézete, 8000 alkalmazottal és 600 millió dolláros évi forgalommal.

A Basis nagyméretű szöveges adatbázisok kezelésére fejlesztették, a legnagyobb Basis adatbázis 120 millió rekordot tartalmaz.

Az adatbáziskezelő kliens/szerver architektúrájú, a kliens processzek egy kernelen keresztül férnek hozzá az adatbázishoz. A kernel többszörös hozzáférésű szerver, ami elosztott hálózati környezetben való működést is biztosítja.

A BasisPlus rendszer előnyei az alábbiakban foglalhatók össze:

- szöveges és relációs adatbáziskezelő
- nagy teljesítmény
- komplett könyvtári rendszer
- referenciák külföldön és Magyarországon
- IDI és a Batelle a háttérben

Alkalmazások

A BasisPlus az államigazgatásban, kutatásban, gyártásban, sajtónál, könyvtárakban, múzeumokban széleskörűen használják, több mint 2000 alkalmazásban.

A BasisPlus formattált szövegek, sőt komplex dokumentumok tárlására is alkalmas, megőrizve a szöveg stílusjegyeit, a grafikákat, fotókat: SGML és DDIF/CDA formátumok használatával. Mind az adatbázison belüli, mind az azon kívüli, de az adatbázisból hivatkozott kép kezelését biztosítja.

Magyarország egyik legnagyobb adatbázisa az MTI Sajtóadatbankja – mely több mint félmillió dokumentumot tartalmaz, naponta frissített, 24 órán át elérhető – szintén a Basis adatbáziskezelőt használja.

ZÁRÓ PLENÁRIS ÜLÉS:

**Alkalmazásfejlesztés a 90-es években.
Nagy világcégek válasza.**

Elnök: Dr. Bakonyi Péter

SAA AD/Cycle Status and Directions

Application Standardization - SAA

Mojmir Baumgartner, IBM EE

In March 1987 IBM announced a unified standard architecture for applications designed to run on four of the IBM's strategic operating systems (OS/2 EE, OS/400, VM, MVS). This operating systems are designed to enable exploitation of a wide range of computers including the PS on one side and the largest corporate mainframe on the other side of the spectrum, they are often called 'SAA platforms'. The announced architecture is today widely known as SAA (Systems Application Architecture).

The unified application architecture, if implemented consequently, allows for applications developed on one of the stated platforms to run on any other platform without modifications. The applications are portable ! In the source form they are identical on all systems. They look the same for the programmer but for the enduser as well. The consequences are obviously very advantageous: reduced cost for training of application programmers and users, portability of solutions and transferability of skills etc., increase productivity.

A very important part of every platform are the application 'enablers', making the whole idea work. A typical example is the unified type of database on all platforms - relational database. Access to data in relational database is programmed with standardized language called SQL (SAA, ANSI, ISO standard). Language SQL is part of standardized CPI (Common Programming Interface) which is used on all platforms to communicate with application enablers.

One of the longterm goals of SAA is a standardized solution for distributed processing, in particular for distributed relational database. The standard which is part of SAA and is called DRDA (Distributed Relational Database Architecture) is already implemented in corresponding database management systems on all SAA platforms. It is obvious that a standardized application architecture should be beneficial in such a complex environment.

Standardization of application development process - S.4.4 AD/Cycle

Problems we have to deal with in the process of application development and maintenance are well known. They shall become even more difficult to manage in the future, because of increased complexity and capacity of the application development and execution environment and the increased sophistication of user requirements. The number of qualified professionals for application development shall always be limited, therefore it is really important to make better use of their skills, and help them to master the increased complexity and meet the expectations regarding the productivity and the quality of their products.

IBM's strategy in this domain is to provide **the environment for application development and maintenance** which corresponds to customer business interests and allows for increased productivity and quality of application development and maintenance and better control of all processes in the application life cycle. The foundation of this strategy is SAA.

Solution announced Sept.19,1989 is called **AD/Cycle (Application Development Cycle)**. AD/Cycle was originally defined as the 'framework' for application development which defined the standards, methods, tools used in **all phases of the application life cycle** (requirement definition, analysis, design, produce, build and test, production, maintenance, redevelopment etc.) and their integration. The real purpose of the framework is to allow efficient usage of many combinations of IBM and non-IBM products, which are the most appropriate for several tasks. The selection of the tools depends of course on the formal method chosen for a particular task beforehand. A tool designed for one phase of application development has to communicate with another tool that addresses another phase even if the two tools come from different vendors.

Today we think of AD/Cycle in terms of a new paradigm called 'composite applications'. Other examples of this paradigm are System View, Information Warehouse, Office Vision. The main reason for the wide acceptance of this new paradigm is really inability of any (even the biggest) single company to provide complete solutions. In a composite application some principles have to be followed. One important requirement is the ability to plug/unplug new components (tools in AD/Cycle). Another requirement is of course the extendability of the composite application with new elements (e.g. inclusion of new tools as the application development technology advances). It is obvious that some formalized framework is an absolute necessity, as well as an established management process to control the evolution of a composite application. The level of integration of every single component has to be precisely defined (standards), every product has to conform to the standard requirements in order to become part of such a composite application. In composite application AD/Cycle many components are available today partly from IBM, partly from other vendors. *Current State of AD/Cycle Framework*

In AD/Cycle framework can be conceptually presented as two tightly interconnected groups of products which enable a high level of automatization of application development process.

1. Application Development Tools

A comprehensive family of CASE tools, languages and utilities, which use as the frontend a PWS (Programable Work Station). On the PWS the OS/2 operating system has to be used and it has to be connected to an SAA host computer, which is in fact acting as a server providing some shareable development resources and also integration of tools. The tools are used in **cooperative mode** where part of process is executed on the PWS and part on host. The tools belong to three categories:

- Tools for specific life cycle tasks:
 - tools for modeling, analysis, design, forward engineering, reengineering, reverse engineering etc.

- compilers (C, COBOL, FORTRAN, RPG, PL/I, REXX), generators (CSP, ...),
- language sensitive editors
- test tools
- Cross life cycle tools which support activities like:
 - process management
 - project management
 - impact analysis
 - documentation
 - reuse
- Utilities to populate the repository using as input existing sources (dictionaries, database descriptions etc.)

All tools are exploiting the power of the PWS with OS/2 by executing as much as possible of their functions locally and using GUI (Graphical User Interface) and multitasking in particular. Their user interface has to be strictly SAA CUA (Common User Access) compliant. They are all installed and invoked in a standard fashion. The majority of the CASE tools for the initial phases of the application life cycle (including design) are not produced by IBM but by specialized vendors which are members of IBM International Alliance. The list of members includes the well known names like Bachman Information Systems, Easel Corporation, INTERSOLV Inc., Knowledgeware Inc., Micro Focus, Synon. The tools for later phases (compilers, generators, test tools etc.) are mainly provided by IBM. A special strategic position is assigned to the application generator CSP (Cross System Product). One of the present standard requirements for all tools used for the application design phase is their ability to express the design in a language called ESF (External Source Format). ESF can be understood by CSP as input for application code generation.

2. Application Development Platform

In the long term solution for the composite application AD/Cycle the interoperability of all tools should be enabled using common Application Development Platform installed on the host computer. Application Development Platform is taking care of the integration.

Platform performs a group of functions using corresponding products, which are all provided by IBM. Platform is supporting cooperative operation of all tools. The most important parts of platform are:

- **Workstation Services** - enabling installation and invocation of tools.
- **Repository Services** - supporting common descriptive database for all participants in the application development and maintenance process. Repository should contain all the descriptive information relevant in the process. It is based on standard INFORMATION MODEL, developed jointly by IBM and members of International Alliance. The information model is not complete yet and surely never will be. If it is to support future technologies it shall be continuously maintained and refined. One of the really difficult problems is the 'generality level' of the information model and as the consequence the complexity of the corresponding repository. It in fact has to support many different approaches and methods introduced by members of IA. The exchange of information between the various tools shall occur via repository. With current version of information model the parts which are well defined are the ones relating to modeling, relational database in general, DB2 in particular, IMS hierarchical database, 3GL (in particular COBOL), end user interface.

Repository Services are supported by the IBM product called

Repository Manager, which is currently available only on MVS platforms and is DB2 based.

- **Library Services** supported by the IBM product SCLM (part of ISPF/PDF) are taking care of program libraries which are containing the actual code in different stages of development, testing or production.

Levels of integration

Today two possible levels of integration are actually defined by IBM and International Alliance:

1. Level 1 is a transient solution. On this level each tool has to comply to only two requirements: It has to use CUA for its user interface and it has to produce ESF as the end result of the application design phase.
2. Level 2 is the complete solution. Each tool has to comply with demands of level 1, but also with additional requirements. One requirement is the standardized installation of the tool to run in cooperative mode using Workstation Services of the Application Development Platform. The other additional requirement is data integration based on Repository Manager and Library Manager functions of the Application Development Platform. International Alliance announced their products will comply to level 2 by December 1992.

AD/Cycle is a consequence of the information systems evolution and incorporates many of the existing IBM products. At the same time it is introducing many new ones. Many of them are available and can be used today. AD/Cycle is open-ended (extendible) and has clearly defined interfaces (CPI) for all the building blocks (tools). High level technical support is provided to 'tool builders'. *Future directions*

One of the permanent directions is to further develop the Information Model and as a consequence the Repository Manager. One of the areas to be included in order to completely support the current level of application development technology and as the technology shall advance, the information model shall have to follow the progress.

There is a statement of direction to implement complete AD/Cycle on all SAA host platforms. Currently Repository Manager is available only on MVS systems, but many other functions are available on other SAA platforms as well. The users on non-MVS platforms can in fact use tools which comply with Level 1 of integration, but also some elements of Level 2 integration (Workstation Services, Library Services, Cooperative Development Environment for 3GL).

The long term direction of AD/Cycle is to move from 'integrated application development' to 'model driven application development'. There are speculations that in far future modeling and design tools may even gradually replace SAA's original CPI's as used by programmers today. This certainly is relatively far off. Enormous investments are still required to advance the integrated application development solution supported by the composite application AD/Cycle to a level of completeness which shall satisfy even the most demanding users. But many of the underlying technologies have been very successful and are widely used today, for example DB2, CSP, OS/2 and many CASE tools provided by members of International Alliance. IBM and IA are firmly committed to complete AD/Cycle and make it the real solution for the application development and maintenance problem.

Mojmir BAUMGARTNER comes from Slovenia and he works for IBM Eastern Europe as a database specialist. His responsibilities include support for AD/Cycle and related IBM products. He has been deeply involved with database and related application development for more than fifteen years. In his professional life has been dealing with various types of database ranging from traditional hierarchical and network database to the latest relational database, including of course a multitude of products, in the last five years in particular DB2. His experience includes data modelling, application and database design and implementation, consulting, teaching in IBM education centers in many countries and on local university, work in IBM SQL/DS lab in Toronto. He is author of some database related articles and a textbook (Database Management Systems) published by University of Maribor in Slovenia.

Information technology strategies of international corporations in the 90s.

Heinz PROKOP - ALLIANZ AG.

1 The significance of information processing for financial service companies

Companies offering financial services, primarily banks and insurance companies, have for a long time striven to make intensive use of modern information and communications technology in order to rationalise operational procedures, improve retail services and provide effective support for their sales forces where these exist. Extensive and complex data bases with a tight network of terminals bear witness to the level of this development. Information and communications technology is therefore of strategically important to insurance companies because - more than in the manufacturing industry - information systems represent the production environment and infrastructure of the intangible product "insurance" and thereby guarantee the insurance company's ability to function and to act as well as strengthening its ability to compete. This development will continue in the future, although it will not be characterised by continuously growing numbers of installed hardware and software, but rather by increasingly intensive utilisation coupled with a trend towards integration and networking (integration = functional combination of various components, such as data, text, speech and image; networking = centralised and decentralised combination of various hardware and software components). The principal points of departure for future information technology strategies are the market and competition. This is because

- national markets are becoming more international

- products are expected to be more specialised and customers are becoming more demanding
- sales and marketing strategies are becoming more varied (for example banking and insurance markets are converging)
- higher standards of customer counselling and staff training are expected.

2 The Allianz as a worldwide company

The Allianz identified these requirements of the market and competition at a very early stage and, drawing the consequences, developed in the past few years from an almost exclusively national insurer into one of the biggest international insurance groups. Today the Allianz is represented in more than 50 countries and on all continents by more than 50 insurance companies and representative offices. It is thus the world's most international insurer and can offer its industrial customers, in particular, worldwide services. In 1989 worldwide premium income was approximately DM 31.8 billion, of which DM 19.4 billion (61 %) was accounted for by non-life and personal accident insurance and DM 12.4 billion (39%) by life and health insurance. The Allianz holds a leading position among the world's top ten insurance companies. In Germany and Europe it is the undisputed number one. Its head office is in Munich and it is from here that the group's worldwide activities are directed.

3 Information technology at the Allianz

Information processing also plays a major role at the Allianz. Worldwide the company currently has 41 computer centres with more than 37 mainframes in operation. In Germany alone the Allianz has 9 computer centres where the biggest central units available on the market are installed. On every working day they carry out approximately 8 million online transactions. Currently there is a total of approximately 2000 MIPS and approximately 4000 gigabytes of disk capacity installed. A total of 31,000 full-time and more than 80,000 part-time agents advise our customers and sell our products. More than 50,000 terminals and PCs are installed in complex national and international networks allowing the office-based staff and the sales force access to all relevant data. The Allianz operates one of the biggest and most complex networks worldwide.

4 Market requirements in the 90s

By the end of the century the times and the market conditions will have undergone drastic changes. Key factors will be

- deregulation and harmonisation in the European Community
- demographic development in western countries
- changing income structures in West and East as well as the economic integration of the eastern countries
- the customer (individual needs, more counselling)

- the competition (new competitors, new products, new distribution channels)
- the development of information technology.

These factors will make it increasingly important for a company to respond to the market more quickly and more flexibly than its competitors. Time-to-market is the catchword for this management theory. The Allianz has just had to implement it in the information technology sector at its new acquisitions in Germany's eastern federal states in order to stay ahead of the competition in that part of the country. New on-line application systems for the administration of 20 million policies, 6,000 terminals and one of the biggest satellite networks in Germany had to be installed within a few months.

5 The IT strategy of a global company

It is becoming more and more important not just to install information technology at the various companies in each country, but - in view of the strategic importance - to develop a global IT strategy. Its goal must be to optimise

- organisation and methods
- security (backup)
- utilisation

and achieve necessary synergies in the technical and operational fields. Only in this way will it be possible to offer international customers constant and rapid service in the future.

The core elements of such an international IT strategy are:

1. Clear organisational and structural rules on coordination and advice (e.g. advisory board encompassing all important international IT functions)
2. Development of an international communications system (components of which are electronic mail, planning and control systems, etc.)
3. Coordinated use and exchange of IS experts with a view to international know-how transfer
4. Coordination and optimisation of hardware and software procurement (discount opportunities through group contracts)
5. Coordination and exchange of methods, standards, applications that can be used universally
6. Future common applications must be subject to an application strategy with the following chief characteristics:
 - 6.1 Future flexible applications must be based on an applications and functions architecture which makes it possible to
 - set up information systems with re-usable modules throughout all business branches
 - use software from various suppliers via standard interfaces and

- initiate cooperation among all Allianz companies.

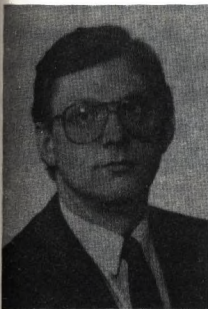
An applications architecture of this kind is currently being implemented.

- 6.2 Furthermore, future applications systems must to a far greater extent support not only "how" a task is performed, but also "what" it entails. Key aspects here are process and knowledge-based systems, developing intelligent documents, e.g. letters, invoices, forms, etc., and providing them with functions which simplify working procedures, not just at one's own workplace but also for the addressee.
- 6.3 Information and communications technology must place instruments and tools at the disposal of the users which only provide the technical framework, e.g. generators, interfaces, drivers, analysis and evaluation systems as well as the technical infrastructure (networks, hardware and software).
- 6.4 An IS development environment is needed which, using standardised and automated processes and with the aid of modern software technology, such as process models, active data dictionaries, prototyping and test procedures, makes business-oriented applications possible.
- 6.5 Future systems must also support complex counselling, discussion and decision-making processes, i.e. also cooperation in organisations.

- 6.6 It will be necessary to develop distributed systems with distributed data bases and information pools which link up centralised and decentralised solutions in a way that is compatible with organisational requirements. This includes forms of electronic self-service, i.e. passing administrative functions on to the customer, for example account statements, profit notifications via teletext.
- 6.7 Of growing importance are man-machine interfaces and the design quality of software ergonomics (user interface software, dialogue managers, adaptive systems). Particularly high demands will have to be made on these in future and thus also on the adaptation of information and communications systems to the needs and habits of the users.

Heinz Prokop

Studied mathematics and physics at the Universities of Marburg and Giessen. Member of the Board of Management of Allianz Versicherungs-AG since 1987 with overall responsibility for organisation and accounting and thus also for the entire information technology sector.



Harold K. WILSON:

STRENGTH AND DIVERSITY : THE KEYS TO GROWTH IN THE 1990s

Business Objectives

The question, "When is our information system going to start paying off?" is one we still hear among corporate executives. They fear that the massive investment they made in IT in the 1980s as a support function for their primary business will fail to translate into improved business performance as their systems become obsolete.

The answer, of course, is that IT never was an end itself. The use of technology in the 80s was mainly to mechanise old ways of doing business, IT provided increased efficiency and control for many already existing job designs, work flows and organisational structures that came of age in a different competitive environment and before the advent of the computer. To enable a corporation to be successful in the 90s, IT must be viewed as a strategic tool for improved competitive performance.

In a time of rapidly changing technologies and ever-shorter product life cycles, outsourcing has become a viable alternative for many European corporations. Outsourcing for consulting, systems development, integration and management offers financial and technological benefits to remaining competitive in the era of change. Our business is about closing the gap between the latest technology and the application of the technology to competitive ends by our users. This involves two perspectives; the technology itself and an understanding of the business strategy of the companies who want to use it.

The business objectives of sophisticated user organisations increasingly recognise the role of IT is securing and enhancing their market position. That represents a dual challenge - a challenge for the user, to recognise the central and strategic role of IT in meeting the corporation's business objectives, and a challenge for the supplier, to deliver technology solutions to fulfil that strategic role. In the 1990s, our understanding of our customers' business objectives is our starting point.

The strategic importance of information technology is increasing a business recognise the need for timely, accurate and usable information to complete in an enlarged and more complex European marketplace. Everyone of our customers has completely reviewed their business strategy and objectives to take account of the opportunities and challenge presented by the Single Market - so that Europe is now the largest IT market in the world, accounting for 36.6 per cent of the \$307 billion spent globally on IT products and services in 1990 (source: IDC).

Indeed their market, like those of most major corporations, is now becoming truly global. Advances in telecommunications technology, and steady progress towards telecommunications liberalisation, have forced companies to reorganise their multinational operations; trade barriers continue to fall, while information technology becomes more powerful every day: the number of components per chip is doubling every 18 months.

As a result, corporate survival itself now depends upon supplying customers not only across national borders, but across one, global, market. Intensifying competition has been joined recently by the effects of economic downturn, while the need to address wider markets, including the new markets to the east, is stretching resources to the limit.

It is no surprise, in this scenario, the outsourcing has so rapidly entered the corporate mainstream. As modern technology becomes more powerful, the use of that power becomes more strategically important. For example, in the past 20 years, cost per MIP has declined by factor of 100,000, while telecommunications technologies have reduced in cost by 10-15% annually. The mere stockpiling of technology which is freely available to one's competitors offers no competitive advantage whatever.

For today's global corporation, the use of consistent, strategic information throughout the corporation to gain competitive advantage is the key. To function successfully in the global environment, an information infrastructure must link worldwide business units, leveraging resources in a manner that transcends national boundaries and geographic markets.

This infrastructure, with its ability to link all segments of the business together, becomes the backbone of the successful global business. By providing a pipeline for common financial, marketing and operating information, it adds the flexibility necessary to adapt constantly to the ever-changing competitive marketplace.

Let me illustrate my point with an example.

Caterpillar is a world leader in designing and manufacturing earth moving equipment and diesel engines. In Belgium, EDS is implementing an ambitious plant automation system for Caterpillar's Gosselies plant, by far the company's largest manufacturing site outside the US. The plant employs 5,000 people to manufacture machinery which is exported to other European countries as well as the Middle East and North Africa.

The system designed by EDS is a prototype which can be adapted to other Caterpillar plants located around the world. The system simultaneously coordinates operations at the plant's different buildings and automatically supplies parts and materials so that production can continue without interruption.

Caterpillar uses the system to coordinate operations between its welding, machinery and final assembly building at Gooselies, and EDS is now developing a similar automation project in the company's Grenoble plant. This is a natural outgrowth of Caterpillar's objective of an automation design which can be adapted to different locations with only minor modifications.

The example of Caterpillar is a good illustration of a customer operating in a global market, using a global outsourcing supplier in partnership to develop solutions which can then be replicated and modified to suit its operations in worldwide locations and different markets. The value to the customer comes not only on the use of technology, but in the understanding the IT service provider brings to the organisation's operations and global needs. Also, the depth of knowledge, expertise and infrastructure resources that the provider has at its disposal.

The Impact of Consolidation

The past year has seen our business - outsourcing - take centre stage in another way. Europe's business pages have been scattered with the story of consolidation in the IT service sector. Cap Gemini Sogeti signed a major agreement with Daimler-Benz, while EDS has been strengthened by the acquisition of SD-Scicon.

What we have seen in the emergence of a number of global IT service providers I think it is interesting to consider how the concept of globalisation applies to IT service providers. How will global service providers compete, how many will survive, and - most importantly - why is that a good for the customer?

Clearly, the global service providers of tomorrow will have the resources to withstand intense competition for business from global customers. Demands and expectations will continue to increase following the completion of the single market, as long as markets in eastern Europe continue the process of modernisation, and a strategically-significant players emerge from Japan and the Pacific Rim.

I mentioned earlier that we provide value for our customers thanks to intellectual capital and technological infrastructure. But in order to compete at global level, meet our customers' global needs, one needs to have a global perspective. Any company with a global aspirations, to extend its business worldwide, must operate with fewer barriers and provide its people with greater access to information.

For the global service provider, the perspective is a prerequisite to establishing the technical infrastructure, and marshalling its human resources, so that it may effectively meet the needs of its customers across the world, and over time. The Caterpillar example I mentioned provides an illustration of that: merely designing and implementing that control system as a solution to circumstances in Belgium at that time would have been inadequate. It is the mark of the global supplier that solutions are designed with a view to replication in other geographical locations, and to revision over time.

But what of the future? How will EDS, and the other emergent global IT service providers, sufficiently enhance their offering to customers over the next few years?

It is firstly important to recognise that developments in the European theatre are not occurring in isolation. Competition is not diminishing - quite the reverse. Quite apart from the factors I have already discussed, the IT services industry is being further stimulated and galvanised by the emergence of a new global players from the Far East.

The Japanese software and services market is forecast to grow to Y.16 trillion by the end of the decade, a process chiefly driven by new entrants to the market. The next few years will see the emergence of a number of new strategic players, formed as a result of privatisation, or as the result of breaking off of IT divisions of larger businesses.

Within the Japanese market itself, the competition posed by new entrants - many affiliated to major financial institutions such as Sumitomo - is forcing leaders in the software industry to rethink their systems integration service capabilities. Similarly, software and service suppliers affiliated to hardware vendors have reexamined their relationship with their parent organisations, and this is resulting in greater differentiation in service offerings.

As the position of Japanese businesses in traditional areas of strength - electronics and automobiles, to cite two obvious examples - continues to be threatened by market entrants from Pacific Rim countries, we predict the explosion in the Japanese IT services sector will have an increasing impact in other markets worldwide.

So how will they, and we, compete in future? In my view, the successful in IT services will need to meet five major criteria.

1. INTERNAL GLOBAL INFRASTRUCTURE
2. TELECOMMUNICATIONS AND PROCESSING POWER
3. INVESTMENT IN NEW TECHNOLOGIES
4. FINANCIAL STRENGTH
5. TECHNICAL CAPABILITIES

EDS' global infrastructure includes 21 Information Processing Centres (IPCs) around the world. These centres store customer accounts, handle processing needs, and are equipped with the latest backup and security systems. An average IPC has a processing capacity of 200 MIPS.

Those centres are linked by robust and resilient private telecommunications network. 280 000 terminals worldwide are linked to EDS*NET, which processes about one billion transactions per moth.

This infrastructure is a basic requirement for EDS to manage and implement voice and data networks on behalf of customers across the globe. The telecommunications solution may involve development of custom software and the provision of processing resources. Increasingly, we are also involved in providing teleconferencing, electronic mail and business satellite services.

We are not alone in recognising the value of adding to our recourse base, either through acquisition or partnership. The synergies between our business and that of SD-Scicon have surpassed our expectations, and the software development skills of the SD-Scicon group will enable us to broaden the services we provide to customers in key European and global sectors.

Investment if that sort is a necessity for survival, and will increase as a differentiator over the next few years, further singling out the small group of service providers able to fulfil their commitment to the new global of the late 1990s.

In sum, the success of global IT suppliers depends on how they anticipate, manage and profit from this era of unprecedented change, and the increased expectations of customers.

Harold K. WILSON - Director

Electronic Data Systems (Europe) SA
Geneva, Switzerland

Personal Profile:

Responsible for all EDS activities in East Europe and for pan-European communications.
Former: President, ITT CR Systems;
Senior Associate, Booz Allen & Hamilton;
Vice-President, Litton Mellonics.

**Software for people who can't predict
the future**

**Manfred Joseph
ORACLE**

The Headlines:

- * "IBM allies with Apple!"
- * "DEC pushes Unix!"
- * "Oracle sets Supercomputer record!"
- * "Borland buys Ashton Tate!"

The Gamble

- * Graphical User Interfaces
 - Windows or Presentation Manager 7
 - Motif or Open Look?
 - DecWindows or New Wave or Macintosh
- * Operating Systems
 - DOS or OS/2
 - Proprietary or Unix
 - VM or MVS
- * Networks
 - TCP or SNA or DECNET or NETWARE or LAN MAN
or LAN SERVER or VINES?
- * Hardware architecture
 - VAX or SMP's
 - Mainframes or Massively Parallel Computers?
 - Workstations or 486 PC's?
 - SPARC or MIPS?

We support them all

The Result

- * Dominant Market Share
- * A Billion Dollar company in less than 15 years

Meeting The Challenge

- * Complete Set of Products and Services
- * different operating system
- * different hardware
- * different databases
- * Tools and Applications on top of the database
- * Technology transfer services around the world

Islands of Data

- * Decentralization in the 80's
- * Connectivity in the 90's to transform a complex network into a single computer with transparent data sharing

Portable and Open

- * pre-requisite for transparent data sharing
- * Ability to operate with other repositories (DR/7, RMS etc.)
- * preserves investment

Complete Set of Tools

- * CASE
- * Application Development
- * Office Automation

CASE

- * A Methodology
- * Full life-cycle from Strategy to Maintenance

Development Tools

- * experience built in the products
- * integrated with CASE and Office Automation

Office Automation

- * Extension of DBMS services
- * back-end and front-end type

Architecture

- * time-sharing mode with terminals
- * Client/Server

Graphical User Interfaces (GUI's)

- * various
- * native "look and feel"

Interoperability

- * Tools work with various back-ends
- * various front-ends to the DBMS

Applications

- * Financial Applications
- * Hungarian-ization underway

System Integration

- * One-stop solutions
- * partnership

A NYILT RENDSZEREK VILÁGA

IAN J. LEACH

ICL HUNGARY

Az elmúlt években egyre nagyobb népszerűséget vívot ki magának az a gondolat, hogy az információ technika lehetőleg minden elemét függetleníteni kell egy adott gyártótól, lehetővé téve ezzel különböző szállítók rendszereinek teljesen nyílt összekapcsolását.

Az ICL ennek a gondolatnak -a nyílt rendszerek alkalmazásának- az első pillanattól kezdve aktív támogatója, és egyik továbbvivő motorja, hiszen alapító tagja az X/OPEN, majd később a UNIX International szervezeteknek. Tevékenységét ma is az ezen szervezetek által kidolgozott ajánlások, és szabványok alapján végzi.

Ennek bizonyítéka volt az a nagyszabású rendezvénysorozat, amelyet ez év februárjában tartottak az ICL bracknelli központjában, "World of Open Systems" címmel, ahol a cég bemutatta a nyílt rendszerek fejlesztése terén elért eredményeit. A nyílt rendszerek megvalósítása az ún. "OPENframework" keretében történik, amely magába foglalja a nyílt rendszerek felépítését, az alkalmazott szabványokat, valamint az egyes elemek közötti kapcsolatokat is.

Az "OPENframework" struktúra hardver platformját a már jól ismert DRS család alkotja, amely az ideai bejelentéseknek megfelelően rendkívüli változáson ment át, hiszen ma már a teljesítményskála a néhány felhasználós géptől az ezer felhasználós rendszerig terjed.

Megjelent az első Level 200-as sorozatú 40 Mhz-es Sparc RISC mikroprocesszoron alapuló asztali szerver gép amelyet az ICL anyavállalatánál a FUJITSU-nál fejlesztettek ki.

A család nagyobb eleme a Level 400-as gépcsalád, amelynek jelenleg négy tagja van, a 25 Mhz-es egy processzoros változattól a két processzoros 40 Mhz-es változatig.

A DRS 6000-es család legnagyobb tagjai a Level 600-as sorozatú gépek, amelyek az egy processzoros 33 Mhz-es változattól a négy processzoros 40 Mhz-es változatig öt tagot tartalmaznak.

Ezen utóbbi két gépcsalád belső felépítése is teljesen megegyezik, így a kártya szintű kompatibilitás is biztosított közöttük.

Az ICL kifejlesztett egy teljesen új, a legszigorúbb ergonómiai és biztonsági igényeket is kielégítő, 13 tagból álló személyi számítógépcsaládot, amelynek legkisebb tagja egy 25 Mhz-es I80386SX processzoron alapul, míg legnagyobb tagja 50 Mhz-es I80486-os duálprocesszoros rendszer.

A DRS 6000-es család tagjain az XPG3 Plus és az IEEE POSIX 1003.1 igényeit messzemenően kielégítő UNIX SVR4.0 operációs rendszer fut. Mivel a család tagjai azonos típusú SPARC Risc processzoron alapulnak, ezért az alkalmazások gond nélkül átvihetők.

A személyi számítógép családon minden ma elterjedt operációs rendszer futtatható, sőt igazi kuriózum, hogy a UNIX SVR4.0 is megjelent ezen a gépkategórián.

Az ICL a jövőben sem kíván lemondani a technológiai

fejlődés által nyújtott lehetőségekről, ezért további fejlesztéseket tervez a nyílt rendszerek keretében úgy hardver mind szoftver vonalon.

Várható a megjelenése az egy kártyán két processzort tartalmazó rendszereknek, valamint az új 64 bites SuperScalar Sparc processzor alkalmazásának is.

Szoftver oldalon a közeljövőben az SVR4.0 továbbfejlesztése és a B2 biztonsági szintet nyújtó változatának a megjelenése várható, lehetőséget nyújtva az alkalmazások további bővítésére.

DOMINO - a Siemens-Nixdorf CASE-stratégiája

Alfons Demmler

A szerző 15 éve dolgozik a Siemens-Nixdorf-nál, illetve a Siemens-nél. Eredeti végzettsége matematikus.

Kezdetben fejlesztési feladatokkal foglalkozott, majd az utóbbi két évben a CASE-rendszertervezési részlegnél tevékenykedik.

CASE and CARE
comprehend

Methods

Tools

and

Consultation

CASE and CARE from SNI
are distributed
under the trademark

DOMINO

Phase-oriented life-cycle

Signed-off "milestones" at the end of each phase which serve as baselines for subsequent phase

Three DOMINO engineering disciplines:
organization, systems, and software engineering

One uniform graphical language through all life-cycle phases (GRAPES)

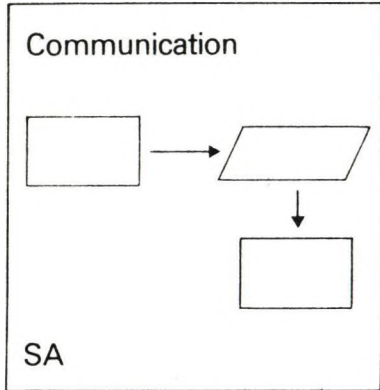
Object-oriented modelling technique which allows simulation and planning for re-use of components

Tool integration by a common graphical user interface and a common database (ERMS)

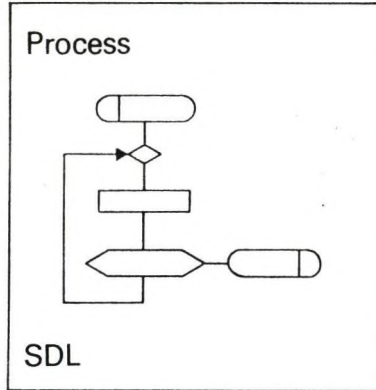
GRAPES Language:

The heart of the DOMINO Methodology is the graphical language GRAPES, which is supported by dictionary-based tools.

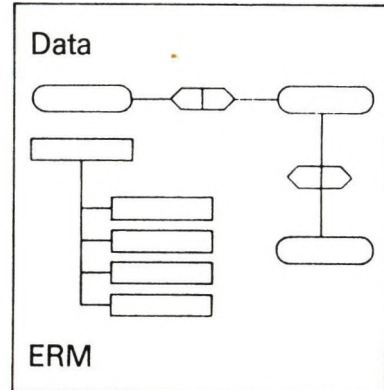
The GRAPES Language combines classical techniques.



(Structured Analysis
Data Flow Diagram)



(Specification and Description
Language, Process Diagram)



(Entity Relationship Modelling
Entity Relationship Diagram)

Three Coherent Views on Systems

Emphasis of SNI-CASE

Goals

- **Development of commercial software**
characterized by efficient cooperation of programs and OLTP-products:
database, transaction monitor and user interfaces
- **Development of systemoriented and technical software**
characterized by efficient cooperation of application and hardware

Types

- **Host applications**
- **Client/Server applications**

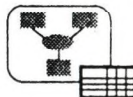


Target System, Development System**Target System**

- Host applications
 - SIA HW and SW platform
BS2000, SINIX
(UTM, SQL-DB, FHS, XMS, Dialogbuilder)
 - MVS
(DB2, CICS, CUA)
- Client/Server applications
 - LAN platforms
 - Server: BS2000, SINIX / UNIX, (OS/2)
 - Client: SINIX / UNIX, MS-WINDOWS 3.x

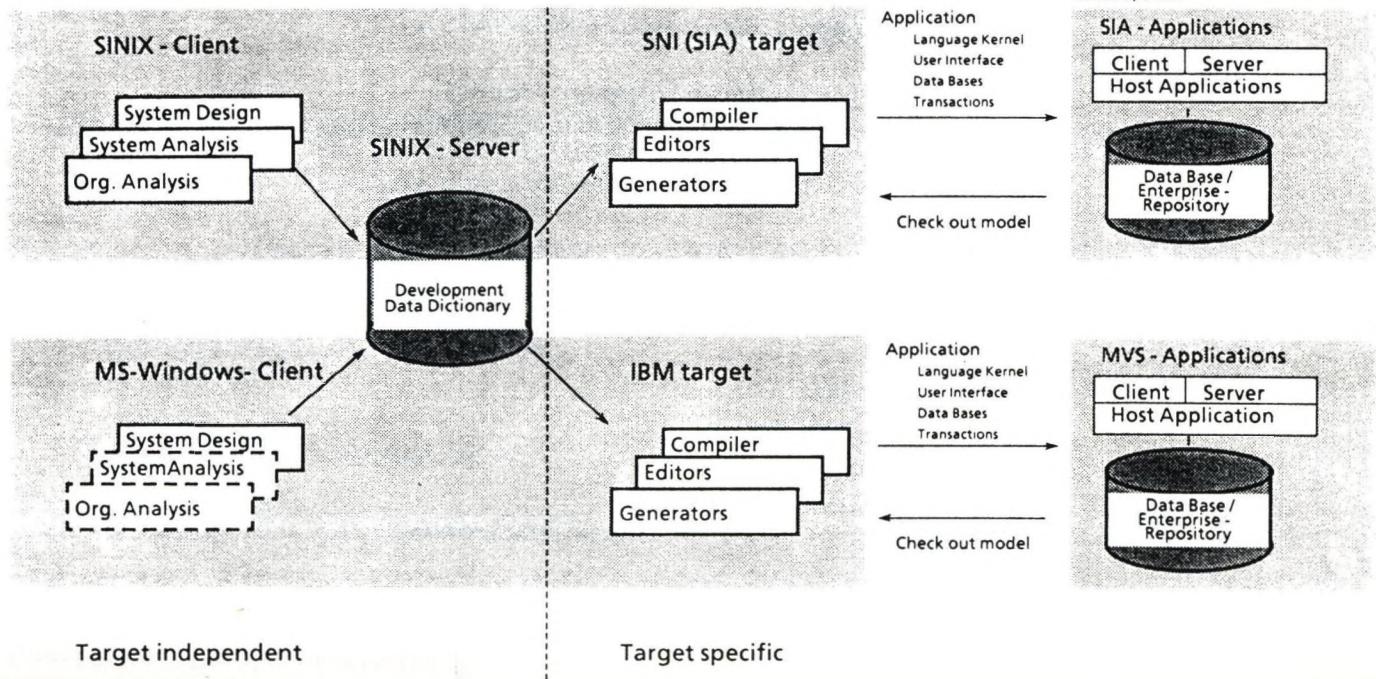
Development System

- Client/Server architecture with data dictionary on SINIX / UNIX server
Clients: SINIX/UNIX, MS-WINDOWS 3.x
- Connection to host systems BS2000, MVS
(Integration, test, installation of the application)



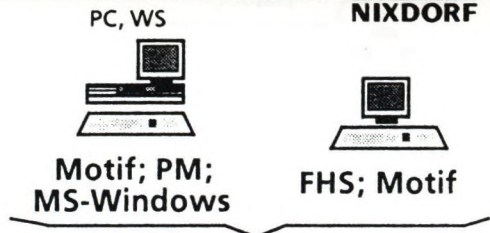
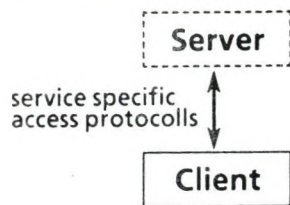
Development System: Client / Server

Target System



207

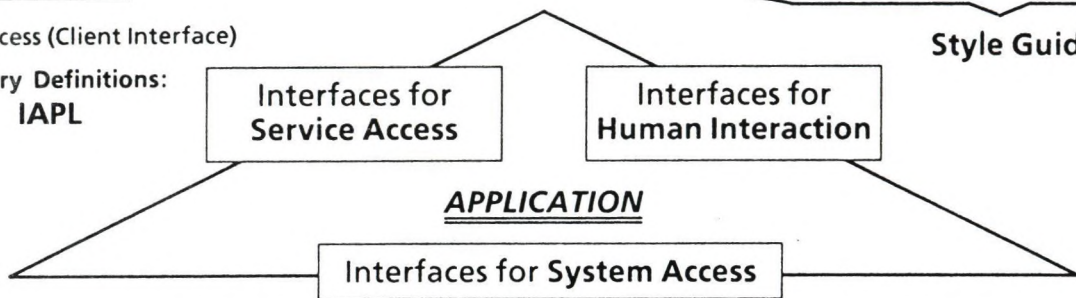




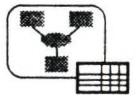
service access (Client Interface)

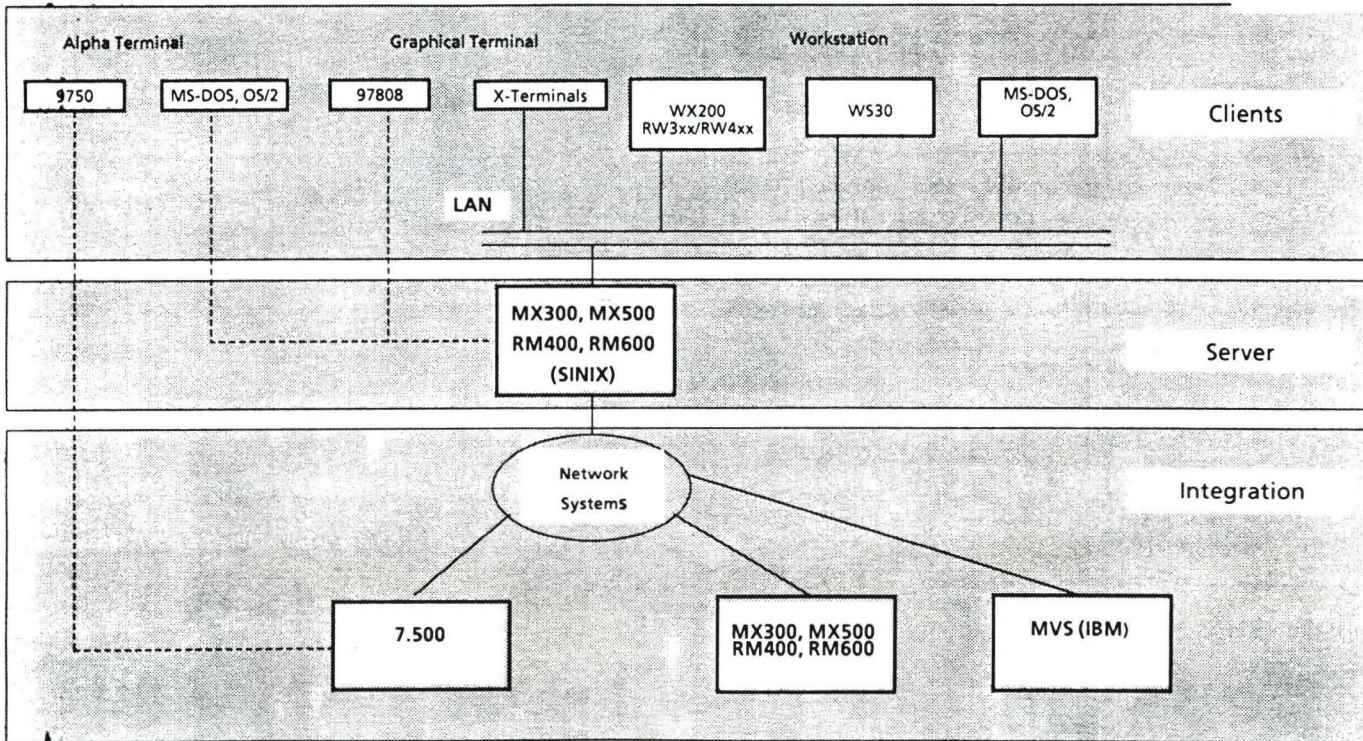
Style Guide

Preliminary Definitions:
IAPL

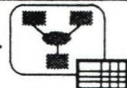


3rd Generation Languages	4 GL	Transaction processing local / distributed	Progr.-Progr. communication	DB - access local / remote / distributed	DD - access
C, COBOL, FORTRAN	DRIVE	KDCS / KDCS - VTV	CPI-C	SQL	IDDS





209



PRINT-PACK