NJSZT MŰSZAKI ÉS TERMÉSZETTUDOMÁNYI EGYESÜLETEK SZÖVETSÉGE
NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG
Budapest. V., Báthory u. 16. Levélcím: Budapest 5. Postafiók 240 1368

II. NEUMANN KONGRESSZUS

3. SZEKCIÓ ELŐADÁSAI

Székesfehérvár,
1983. november 14–17.

IZ/336/3

Dr. Halassy Béla:

ADAM&EVA - Adat- és eljárásmodellezési segédeszköz

1. Általános háttér

Az utóbbi időben felgyorsultak azok a kisérletek, amelyek az információs rendszerek fejlesztésének számitógépes támogatására irányulnak. Ezeknek a próbálkozásoknak az egyre inkább kikristályosodó strukturált tervezési elvek, a mind szélesebb körben terjedő adatszótár-rendszerek /pl. az IDMS IDD/ és a rendszerek leirására és elemzésére szolgáló nyelvek /mint amilyen az MTA SZTAKI-ban kifejlesztett SDLA, vagy a nálunk is alkalmazott PSL/PSA/ teremtettek alapot.

A rendszerfejlesztés számitógépes támogatásának gyakorlati lehetőségei tovább nőttek azáltal, hogy ujabb elvi eredmények születtek a fejlesztés módszerét illetően. Két, meglehetősen ujszerü tervezési elv került előtérbe:
- az adatok és eljárások párhuzamos fejlesztésének lehetősége és követelménye;
- a fogalmi-logikai-fizikai tervezési szintek szétválasztásának igénye.

Mindkét ujszerü elv az adatbázisok tervezését szolgáló adatmodellezésben alakult ki és bizonyitotta érvényességét.

Az elvi lehetőségek ellenére mégsem születtek széleskörüen elterjedt rendszerfejlesztési segédeszközök. Ugyanis az ilyen jellegü programrendszerek tervezésének és alkalmazásának több feltétele van. Ki kellene alakitani egy olyan fogalmi rendszert, amelyet következetesen alkalmaznának a segédeszköz fejlesztésénél és használatánál. Egyértelmüen meg kellene alapozni egy olyan szabályrendszert, amely rögzitené, hogy az adat- és eljárásmodell felépitését milyen feltételek mellett tekintik optimálisnak. Össze kellene állitani egy eljárásrendszert, amely meghatározná azt, hogy milyen tevékenységsorozatban, milyen dokumentációval, milyen fejlesztési ellenőrzőpontokkal kell kialakitani az adat- és eljárásmodellt.

A fogalmi-, a szabály- és az eljárásrendszer egyenként is igen összetett. Az ujszerü elvek nemcsak a felhasználó, hanem még a viszonylag széleskörüen tájékoztatott szakember számára is alapos átgondolást igényelnek a megértéshez. A tájékozott szakemberek között pedig éles a vita az emlitett elveket, fogalmakat illetően. Nem csoda, hogy a fejlesztési segédeszközök mindennapos alkalmazása még várat magára.

2. Az ADAM&EVA fogalmi rendszere

Mellékelt ábránk a modellező segédeszköz adattárát, szótárának felépitését mutatja. A tervezett adatmodellben a valós rendszer jelenségeit egyedekkel /2/, azok sajátosságait tulajdonságokkal /1/, összefüggéseit kapcsolatokkal /5/ tükrözzük a fogalmi elemzés szintjén. Minden egyed tetszőleges tulajdonsággal irható le. A tulajdonságok közül egyesek hordozzák a kapcsolatokat /16/. Mód van az egyed-altipusok modellezésére is /12/. Többszörös, önmagába visszamutató, családfa, halmaz /szinguláris/ és hierarchikus kapcsolatszerkezetek épithetők fel /15/.

A meglévő rendszer adatforrásainak /bizonylatainak, nyilvántartásainak stb./ elemzéséhez szolgál az eredet fogalma/3/. Minden eredet adott számu tulajdonság származási helye /17/. Modellezhetjük az összetett adatforrások belső szerkezetét is /13/. Az eredet fogalmat használjuk az információfeldolgozási müveletek eseményeinek jelölésére is, hiszen minden esemény tulajdonságokkal irható le.

Az eljárásmodell az információs tevékenységeket megindító és befejező eseményekből /3/ illetve az események között végrehajtott eljárásokból /4/ áll. Az események-eljárások összefüggései /18/ adják a tevékenység-események hálós szerkezetét. Az eljárások hierarchikus lebontására is van mód /14/. Az információfeldolgozási müveleteket két csoportba soroltuk:

- Felhasználói müveleteknek hivjuk az információátalakitó tevékenységeket. A tulajdonságokon /adatokon/ végzett feltételes vagy feltétel nélküli algoritmusok külön is leirhatók /20/. A tulajdonságok származtatási összefüggéseit, az ugynevezett előzményhálót /11/ lehet részletezni a számitás leirásával.
- Adatkezelési müveleteknek nevezzük azokat a tevékenységeket, amelyek az adatoknak az adatszerkezetekből való előkeresésére szolgálnak. Az egyedekben lévő tulajdonságok előkereséséhez egyedeket és kapcsolatokat kell kezelnünk adott logikai sorrendben. Ezt a müveletsorozatot rögziti a navigáció /19/.

A szótárban megadhatjuk a modell-elemek szinonimáit. Mindezek az elemek másodlagos névkulcson keresztül is elérhetők. A tulajdonságoknál külön specifikálható az általános értékhalmaz /"domén"/ és annak sajátos alhalmazai /"attributumok"/. A tervező meghatározhatja a szótárban azokat a feltételeket /korlátokat/ is, amelyek esetén az elemek és szerkezetek érvényesülnek.

3. Az ADAM&EVA modellezési szabályai

Az adat- és eljárásmodellt akkor tekintjük optimálisnak, ha
- valósághü,
- teljes,
- minimális /nem-redundáns/
- és egyértelmü.

A tulajdonságok elemzésénél kizárjuk a homonimákat. Ezzel biztositjuk a tulajdonságok egyértelmüségét és fokozzuk azok teljességét illetve javitjuk a valósághüséget. A tulajdonság-egyed viszonyok elemzésével szürjük ki a szinonimákat, ezzel csökkentve a tulajdonságok számát.

Az egyedek elemzésénél a rendszer egyértelmü neveket generál.

Rendszerünk az ismert ötödik normál formán tulmutató egyedszerkezetek képzésére alkalmas, hogy a lehető legtökéletesebb képet adhassuk a valóságról minimális tulajdonság-egyed viszony, vagyis a legkevesebb adat segitségével. Rendszerünk felhivja a figyelmet a normál formák ellentmondásaira /a szerkezeti egyértelmüség hiányosságaira/.

A kapcsolatok elemzésénél kiszürjük a tranzitiv és ciklikus szerkezeteket /minimalitás/. A neveket itt is a rendszer generálja /egyértelmüség/. A kapcsolhatóság vizsgálata utal bizonyos hiányzó kapcsolatokra /teljesség/. A kapcsolatokat a rendszer a tulajdonság-egyed viszonyok alapján generálja. Vagyis az egyedek valósághüsége a biztositéka a kapcsolatok valósághüségének. Rendszerünk a legbonyolultabb kapcsolati viszonyok elemzésére is alkalmas, hogy ezzel fokozzuk a valóság modellezésének tökéletességét.

A tulajdonságok előzményhálójának elemzése /megszakitás- és ciklusszürés/ a felhasználói müveletek egyértelmüségét és teljességét garantálja. Hasonló előzményháló elemzést végezhetünk az események-eljárások összefüggéseire is. Az eredetek tulajdonságainak átfedésével csökkenthetjük a feldolgozási müveletek eseményeinek /input-output/ számát, fokozva azok egyértelmüségét. Az adott kimenetek előállításához szükséges navigációs utvonal alternativáit, az adott eljárás által érintett adatmodell-részeket /az adatalmodellt/ a rendszer automatikusan generálja. Mivel minden egyedre, tulajdonságra és kapcsolatra megfelelő létrehozási, törlési, karbantartási és feldolgozási müveletnek kell vonatkoznia, rendszerünk képes kimutatni, hogy a tervező által megadott eljárásrendszer teljes-e ebben az értelemben, vagyis azt, hogy valamilyen adatmodell-elemre nem hiányzik-e valamilyen eljárás.

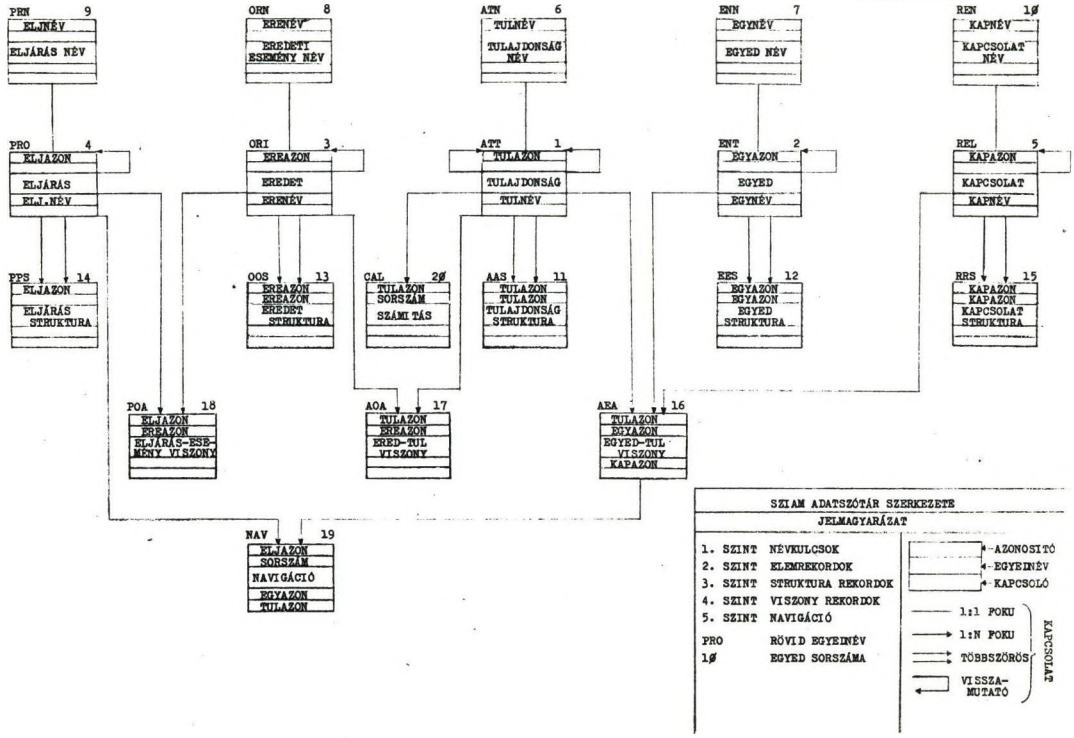
Az adatmodell-korlátok karbantartásához és keresztellenőrzéséhez illetve a már megtervezett adat- és eljárásmodell ujrastrukturálásához megfelelő karbantartó rendszer; a létrehozott modell elemeinek, szerkezeteinek és korlátainak lekérdezéséhez szelektiv lekérdezőrendszer áll rendelkezésre. A tervező, karbantartó és lekérdező programok sorrendjét eljárásrendszerünk határozza meg. A rendszer szabványos bemeneti és kimeneti formátumai garantálják, hogy a rendszertervezőnek mindig rendelkezésére álljon a fogalmi szintü rendszerdokumentáció.

4. Technikai megvalósitás

Már az előzőekből is sejthető, hogy az ADAM&EVA segédeszköz kialakitásánál nem terveztünk SDLA, PSL/PSA, DDL vagy egyéb tipusu modellező nyelvet. Inkább a strukturális kezelés mellett döntöttünk. Tapasztalataink szerint az ilyen nyelvek alkalmazása esetén rendkivüli mértékben megnövekszik az adatrögzitési igény.

Maga a modellező rendszer ESZR és kompatibilis gépekre készült. Mivel a rendszerszótárt a VSAM-ra biztuk, csakis a virtuális kezelést biztositó gépek jöhetnek szóba. A rendszer - szemben elődjével, a SZIAM-mal - OS és DOS operációs rendszerek alatt egyaránt müködik. Programjai kivétel nélkül PLIOPT forrásnyelvben irottak.
A rendszer müködtetéséhez kártyaolvasóra /vagy megfelelő más input berendezésre/, nyomtatóra, egy szalagra és két lemezegységre van szükség. A központi valós memóriaigény 256 Kbyte. A tényleges lemezigény a modellezendő probléma nagyságától függ. Elvileg a rendszer 100 ezer adat és ugyanennyi eljárás modellezésére alkalmas.

A segédeszköz bárki által bérelhető vagy megvásárolható. Használatát Modellezési Kézikönyv és Programozási Kézikönyv irja le. A modellezéssel kapcsolatos ismereteket a terméktől független tanfolyamokon is el lehet sajátitani.


PRN 9
ELJNÉV
ELJÁRÁS NÉV
ORN 8
ERENÉV
EREDETI ESEMÉNY NÉV
ATN 6
TULNÉV
TULAJDONSÁG NÉV
ENN 7
EGYNÉV
EGYED NÉV
REN 1Ø
KAPNÉV
KAPCSOLAT NÉV
PRO 4
ELJAZON
ELJÁRÁS
ELJ.NÉV
ORI 3
ERÉAZON
EREDET
ERENÉV
ATT 1
TULAZON
TULAJDONSÁG
TULNÉV
ENT 2
EGYAZON
EGYED
EGYNÉV
REL 5
KAPAZON
KAPCSOLAT
KAPNÉV
PPS 14
ELJAZON
ELJÁRÁS STRUKTURA
OOS 13
ERÉAZON
ERÉAZON
EREDET STRUKTURA
CAL 2Ø
TULAZON
SORSZÁM
SZÁMÍTÁS
AAS 11
TULAZON
TULAZON
TULAJDONSÁG STRUKTURA
EES 12
EGYAZON
EGYAZON
EGYED STRUKTURA
RRS 15
KAPAZON
KAPAZON
KAPCSOLAT STRUKTURA
POA 18
ELJAZON
ERÉAZON
ELJÁRÁS-ESEMÉNY VISZONY
AOA 17
TULAZON
ERÉAZON
ERED-TUL VISZONY
AEA 16
TULAZON
EGYAZON
EGYED-TUL VISZONY
KAPAZON
NAV 19
ELJAZON
SORSZÁM
NAVIGÁCIÓ
EGYAZON
TULAZON
SZIAM ADATSZÓTÁR SZERKEZETE
JELMAGYARÁZAT
1. SZINT NÉVKULCSOK
2. SZINT ELEMREKORDOK
3. SZINT STRUKTURA REKORDOK
4. SZINT VISZONY REKORDOK
5. SZINT NAVIGÁCIÓ
PRO RÖVID EGYEDNÉV
1Ø EGYED SORSZÁMA
AZONOSITÓ
EGYEDNÉV
KAPCSOLÓ
1:1 FOKU
1:N FOKU
TÖBBSZÖRÖS
VISSZA-MUTATÓ
KAPCSOLAT

dr. Budavári Elemér
Marx Károly Közgazdaságtudományi Egyetem Matematikai
és Számítástudományi Intézet

META-INFORMÁCIÓRENDSZERREL TÁMOGATOTT RENDSZERFEJLESZTÉS ÉS ADATBÁZISTERVEZÉS

1. A rendszerfejlesztést és az adatbázistervezést támogató software eszközök helyzete

Az adatbázistechnika magyarországi alkalmazásának fő problémája, hogy a tervezési, fejlesztési módszerek, segédeszközök szintje elmaradt az alkalmazható software eszközök szinvonalától. Sem elméletileg sem gyakorlatilag nincs pontosan tisztázva, hogy az adatbázis-alapú információrendszerek adat - és eljárásmodellezési folyamata miként illeszkedik a rendszerfejlesztés klasszikus munkafázisaihoz. Ezért a tervezők alapvetően csak az adatmodellezésre, pontosabban a séma kialakitására koncentrálnak és elhanyagolják az eljárásmodellezést, a felhasználói rendszer működési jellemzőinek a vizsgálatát.

A két modellezési munka párhuzamos végzésének követelménye a szervezés tárgyáról teljes képet igényel mind az adatszerkezet, mind annak használata vonatkozásában. A teljes kép viszont csak folyamatosan, a fejlesztési folyamat előrehaladtával áll össze, egyre mélyebb és részletesebb formában.

A kialakitandó információrendszerre vonatkozó ismeretek, dokumentációk alapvetően két fő csoportra tagolhatók aszerint, hogy a logikai vagy a fizikai tervezés körébe tartoznak. Az adatbázis tervezés módszereiből következően a logikai tervezés a teljes rendszerre vonatkozó információigénnyel lép fel. Ezzel szemben a fizikai tervezés, illetve megvalósitás rendszerint alrendszerenként, több éves időszak alatt történik.

Igy egy nagyméretű számitástechnikai fejlesztési program sikeressége elsősorban a logikai tervezés stabilitásán, dokumentáltságán és koordinálhatóságán múlik.

A rendszerfejlesztés egyes munkafázisainak elősegitésére és dokumentálására már hoztak létre számitógépes segédeszközöket, azonban ezek között nincs kapcsolat, nem ölelik fel a fejlesztés és üzemeltetés teljes folyamatát.

Ezért egy olyan számitógépes támogatású rendszerfejlesztési eszköz kialakitásához fogtunk hozzá, amely egyaránt alkalmas a kialakitandó információrendszer tartalmának, működési rendjének, számitástechnikai alkotóelemeinek egységes elvek szerinti meghatározására, kezelésére és dokumentálására. A módszer alapjául egy kétszintű információrendszer szolgál, amelyben:

* az alsó szint: a kialakitandó tárgy - információrendszer,
* a felső szint: a meta - információrendszer, mely adatokat tartalmaz a tárgy - információrendszernek a fejlesztés és működtetés szempontjából fontos jellemzőiről, igy:

- a tárgy - információrendszer szervezeti felépitéséről, tevékenységi rendszeréről;
- az egyes rendszerekben szereplő adatok köréről, azok jelentéséről, jellemzőiről;
- az adatok szerkezetéről, az adatokhoz való hozzáférés jellegéről, módjáról;
- az adatok származási és rendeltetési helyéről;
- a tárgy - információrendszer fejlesztési fázisairól, gépesitettségi állapotáról, stb.

2. A meta - információrendszer koncepciója

A meta - adat fogalmát, szűkebb értelemben, az adatbázistechnika alkalmazása során vezették be az adatbázisban tárolandó adattipusok azonositó és leiró adatainak megnevezésére. E szemlélet általánositásával jutottak el a rekordtipusok, file-ok, programok, majd pedig az információrendszer egyéb, nem számitástechnikai alkotóelemeit leiró meta - adatokhoz. A nagytömegű meta - adat tárolásának és kezelésének bonyolultsága ezen a területen is az adatbázistechnika alkalmazásához illetve az adatútmutató / adatszótár rendszerek kialakitásához vezetett.

Az adatszótárak készitői egyrészt az adatbázis - kezelő software igényeit, másrészt a legáltalánosabb számitástechnikai és információi rendszerbeli kategóriákat, jellemzőket és kapcsolatokat vették figyelembe, s nem a rendszertervezés illetve az adatbázistervezés adatigényét. Ezért többségük meghatározott adatbázis - kezelő rendszer működésének alapja, s nem a rendszerfejlesztés teljes életciklusa alatt alkalmazható segédeszköz.

Az adatbázistervezéshez kapcsolódó elméleti kutatások feltárták azon információk körét, amelyek nélkül a tervezés nem végezhető el. A gyakorlati, vállalati munkák viszont azok összegyűjtésének és kezelésének problémáira, valamint a fejlesztési munkafázisokkal való szoros összefüggésre világitottak rá. Az elméleti és gyakorlati szempontok ütköztetése és szintézise a számitógépes meta - információrendszer koncepciójának kidolgozásához vezetett.

A meta - információrendszer

- a meta - adatbázisból és
- az azt kezelő meta - programrendszerből áll.

3. A meta - adatbázis tartalma

A meta - adatbázis tartalmának meghatározásához a számitógépes tárgy - információrendszer fejlesztésének munkafázisait vettük elemzés alá, s megvizsgáltuk a hagyományos, illetve az adatbázison alapuló fejlesztés közös és specifikus adatigényét. Megvizsgáltuk továbbá, hogy az adatbázistervezés három fő modellezési lépése

- a koncepcionális modellezés,
- a logikai számitástechnikai modellezés és
- a fizikai számitástechnikai modellezés

miként illeszthető a rendszerszervezés hagyományos munkaszakaszaihoz.

A meta - adatbázis tervezése módszertanilag azonos bármely más adatbáziséval, tehát meg kell határozni:

- a META - EGYEDEKET,
- a META - ISMÉRVEKET és
- a META - KAPCSOLATOKAT.

* A META - EGYEDEK a tárgy - információrendszerben megfigyelt objektumok, jelenségek tipusait definiálják (pl.: SZERVEZET, TEVÉKENYSÉG, PROGRAM, TÁRGYEGYED, stb.).
* A META - ISMÉRVEK egy adott META - EGYED tipus előfordulásainak azonositására, leirására és jellemzésére, valamint a META - KAPCSOLATOK létrehozására szolgálnak;
* A META - KAPCSOLATOK a META - EGYEDEK közti természetes adatkapcsolatok, adatkezelési funkciók realizálására szolgálnak.

A tárgy - információrendszer fejlesztése folyamán egyes META - EGYEDEK csak meghatározott fázisban, mások viszont

több vagy valamennyi modellezési fázisban szerepelnek, s igy META - ISMÉRVEIK köre fokozatosan bővül.

A META - ISMÉRVEK specifikálása különösen fontos feladat volt, minthogy az adatszótár rendszerek nem léptek túl az azonositó és leiró (definiáló) META - ISMÉRVEK körén, tehát a felhasználóra, elsősorban az adatbázis adminisztrátorra hárították a feladatot.

A META - ISMÉRVEK nagyobb része kötelező minthogy azonositó, kapcsolatteremtő és osztályozó funkciót töltenek be. Kisebb részük választható, elhagyásuk nem érinti a meta - adatbázisra épülő programrendszer működését. Arra is van lehetőség, hogy a felhasználó a rendszer generálásakor, általa fontosnak itélt új META - ISMÉRVEKET vegyen fel.

A META - ISMÉRVEK egy része kimondottan az adatbázis tervező információ igényeit hivatott kielégiteni, illetve a tervezés eddigi eredményét kivánja rögziteni. Igy specifikálásuk bizonyos mértékig kényszerpályára tereli a tervezőt.

4. A meta - programrendszer funkciói

A meta - programrendszer alapvetően négy részből áll:

* az előkészitő, definiáló, ellenőrző programrendszerből, amely:
  - létrehozza a meta - adatbázis definiciót,
  - elvégzi a meta - adatbázisba kerülő adatok szintaktikai és szemantikai ellenőrzését (off - line adatbevitel esetén);

* a betöltő, aktualizáló programrendszerből, amely:
  - elvégzi az elsődleges betöltést, illetve
  - a folyamatos bővitést, módositást;

* a visszanyerő, tájékoztató, elemző - tervező programrendszerből, amely szolgáltatja:

  - a tárgy - információ rendszerre vonatkozó aktuális fejlesztési információkat,
  - az adatbázis tervezési segédleteket és
  - az egyéb software vagy tervezési eszközökhöz szükséges alapadatokat;

* a karbantartó és helyreállító programrendszerből, amely felhasználói vagy rendszerszintű hibák elháritására szolgál.

5. A meta - információ rendszer működése

A rendszerfejlesztési munkát a szakaszosság és az iterativitás jellemzi. Ebből következően a meta - adatbázis csak fokozatosan töltődik, tartalma állandó változás alatt áll. E változások jelenthetik;

- új META - EGYED előfordulásainak betöltését,
- új meta - adatbázis rekord bevitelét,
- meglévő meta - adatbázis rekord adott mezőinek elsődleges megadását, vagy módositását,
- új kapcsolatok létesitését, vagy meglévő kapcsolatok módositását.

A meta - adatok elsődleges rögzitésére megfelelő bizonylat tipusok szolgálnak, amelyekről off - line bevitel esetén az adatrögzités, on - line bevitel esetén pedig a terminál kezelés történhet. Mindkét megoldási módot a mezőnkénti bevitel jellemzi, minthogy a meta - adatok keletkezésének és folyamatos bővülésének, változásának leginkább ez felel meg, s a végfelhasználó számára is ez biztositja a legnagyobb kényelmet.

A visszanyerő, tájékoztató meta - programrendszer elsődleges betöltés, változások esetén standard reportokkal látja el a tervezőket, melyek alapdokumentációként funkcionálnak, de ad hoc igények kielégitésére is lehetőséget nyújt.

6. A rendszerfejlesztés és az adatbázis tervezés folyamata

A meta - információrendszer alkalmazása esetén a tervezők az adatbázis tervezés korábban emlitett három modellezési fázisát követik. Ezen belül azonban több elemi lépés található, mely mindegyikéhez önálló meta - adatnyerési eljárás tartozik. A főbb elemi lépések a következők:

1./ A tárgy - információrendszer a statikus tevékenység modelljének megállapitása és dokumentálása,

2./ A tárgy - információrendszer dinamikus tevékenységi modelljének feltárása és tárolása a meta - adatbázisban,

3./ Az elemi tevékenységekhez tartozó egyed tipusok meghatározása és definiálása,

4./ Az elemi tevékenységi kapcsolatok ismérv tartalmának egyedenkénti meghatározása, definiálása és tárolása,

5./ Az egyedtipusok ismérvlistájának összeállitása, az ismérvek elemzési, csoportképző kódjainak a meghatározása, az adatbázis várható tartalmának a körülhatárolása.

6./ A koncepcionális szintű egyedkapcsolatok definiálása és tárolása,

7./ A koncepcionális ismérvkapcsolatok elemzése az információs rekordok és csoportok képzése a funkcionális függőség alapján (egyedtipusonként).

8./ Logikai adatbázis rekordtipusok képzése a koncepcionális adatmodellből,

9./ A tevékenységek számitástechnikai egységekre, program funkciókra bontása,

10./ A tevékenységi kapcsolatok programfunkciókra vetitése, a program kapcsolatok meghatározása,

11./ A program kapcsolatok adattartalmának a definiálása, az adatigények csoportositása (adatbázis, külső file-ok),

12./ A logikai adatelérési igények meghatározása, a logikai adatbázis szerkezet megfelelő finomitása,

13./ A tárgy - információrendszer adatbázis sémájának definiálása,

14./ A programok, rekordtipusok, file-ok végleges formájának a meghatározása,

15./ A fizikai megvalósitás, programozás, tesztelés,

16./ A fizikai megvalósitás, üzemeltetés paramétereinek rögzitése a meta - adatbázisban.

7. A meta - információrendszer megvalósitása

A megvalósitási lehetőségek mérlegelésekor elsősorban az IDMS adatbázis kezelőrendszer szolgáltatásait vettük figyelembe. A meta - adatbázis tartalmának megtervezése után világossá vált, hogy az IDD 1.2. változattal ez nem valósitható meg, a magasabb verzió számú változatok pedig még nem álltak rendelkezésre, igy a saját fejlesztésű meta - adatbázis kialakitása és meta - programrendszer kidolgozása mellett döntöttünk.

Jelenleg a meta - információrendszert támogató software egy része rendszertervezési, más része programtervezési és programozási fázisban van.

Garai Péterné - Straub Elek
Központi Statisztikai Hivatal

Statisztikai adatállományok megőrzésének és forgalmának jogi szabályozása

Bevezetés

Az 1973. évi V. törvény az állami statisztika egységes rendszerének megvalósitása és müködtetése érdekében előirja a különböző statisztikai tevékenységek összehangolását, és ennek irányitását a KSH elnökének feladatává teszi.

A statisztikai tevékenységek összehangolása mellett a KSH-nak e tevékenységekről nyilvántartást is kell vezetnie, amelyek egyik célja az érdekeltek - a statisztikai tevékenységet végző szervek és a statisztikai adatok felhasználói - tájékoztatása az adatgyüjtésekről, a rendelkezésre álló adatokról és az igénybevehető adatátadási szolgáltatásokról.

A törvény 7. §-a szerint az összehangolás során el kell érni a számitástechnika hatékony alkalmazását a statisztikai tevékenységben, gondoskodni kell arról, hogy a statisztikai tevékenységet végző szervek a statisztikai adatokat megőrizzék és egymásnak kölcsönösen átadják.

Az állami statisztikai rendszer a legtöbb országban, igy Magyarországon is erősen strukturált. Olyan részrendszerekre bomlik, amelyek többé-kevésbé önállóak. Szervezetileg és a feladatok szempontjából elkülönülnek egymástól a központi állami és az igazgatási statisztikai rendszerek, tematikusan az önmagukban is összetett társadalom- és a gazdaságstatisztika. Mindehhez hozzájárul még az is, hogy egymással párhuzamosan számtalan önálló adatfeldolgozási rendszer müködik. A részrendszerek összehangolása - integrálása - az egyik legtöbb eredményt igérő fejlesztési irány, amely azonos cél felé mutató különböző tevékenységeket, módszereket foglal magába. A részrendszerek együttmüködésének egyik legfontosabb megnyilvánulási formája a statisztikai adatok többszörös, különböző felhasználó által történő hasznositása. Ennek egyaránt vannak tartalmi, szervezeti és technikai feltételei.

Tartalmi szempontból alapvető a fogalmak, osztályozások egyezményes használata.

Az adatforgalom biztositása érdekében egységes szabályozást igénylő technikai és szervezeti jellegü kérdéseket a következőképpen határozhatjuk meg: gondoskodni kell a közérdekü, gépi adathordozón tárolt elemi szintü adatállományok megőrzéséről, karbantartásáról, az adatállományokról való tájékoztatásról, az adatátadások hatásköri és szervezeti feltételeinek meghatározásáról.

A korábbi szabályozás eredményei és a felmerült fontosabb problémák

Az igy körülhatárolt problémakör első jogi szabályozására 1977-ben került sor. A lépés akkori időszerüségét mi sem bizonyitja jobban, mint hogy egybeesett a statisztikai adatok számitástechnikai eszközök utján történő forgalmának fellendülésével.

A 2/1977/VII.30./KSH sz. rendelkezés kötelezővé tette a számitástechnikai eszközökkel rögzitett és javitott statisztikai adatok megőrzését, dokumentálását és az adatállományok bejelentését nyilvántartás céljából a KSH-nak. Előirta továbbá a selejtezési eljárás módját, valamint az átadási kötelezettséget.

A jogszabály teremtette rendezett keretek nagyban hozzájárultak az együttmüködés elterjedéséhez. Általánossá vált a szemlélet, hogy az egyes minisztériumok, országos hatáskörü szervek nem csak saját célra gyüjtenek statisztikai adatokat, azok mások számára történő megőrzése, átadása közös érdek.

Mindezt jól bizonyitja a statisztikai adatok számitástechnikai adathordozón /mágnesszalagon/ történő forgalmának folyamatos bővülése. A bővülés, fejlődés több szempontból is jól megfigyelhető.
A forgalomban átadóként vagy átvevőként érdekelt szervek köre ma már magába foglalja szinte az összes, statisztikai tevékenységet folytató államigazgatási szervet. Megjelentek átvevőként olyan, e körön kivül eső szervek is, amelyek kutatásaikhoz, döntéselőkészitéseikhez ilyen módon biztositják a szükséges adatokat.
Az adatforgalom tartalom szempontjából is egyre változatosabb képet mutat. A rendelkezésre álló adatállományok egyre nagyobb hányada hasznosul igy többszörösen.
A kezdeti szakaszra egy-egy statisztikai felmérés adatait tartalmazó mágnesszalagok másolatainak átadása volt jellemző. Ma már fokozatosan terjednek a hatékonyabb megoldások, ahol az átadott adatállomány az igényeknek megfelelően szelektált. Erre elsősorban az ad lehetőséget, hogy több szerv kiépitette adatbázisait.
Jól érzékelteti az előzőeket néhány adat a KSH Számitóközpont tevékenységéről, amely külső megrendelésre 1979-ben 116, 1980-ban 132, 1981-ben 220, 1982-ben 312 adatátadást teljesitett. Az igénylők közel 70 %-a az állami statisztika egységes rendszerébe tartozó szerv, a többi azon kivüli kutatóintézet, vállalat. Bár az átadások többsége ma még mindig egyszerü másolat, 1982-ben 18 esetben történt szelekció, 34 esetben az adatok forrása a statisztikai adatbázis rendszer volt. A megőrzésre kerülő /archivált/ adatállományok száma is dinamikusan növekszik, melynek egyik oka az, hogy a KSH SZK belső szabályozása szerint éves, vagy annál nagyobb periodicitással gyüjtött adatokat nem selejteznek. Jelenleg kb. 1300 archivált alapadat-állomány van, ez a szám tartalmazza a statisztikai adatállományokon kivül az ezek értelmezéséhez szükséges segédállományokat is.
A megőrzött adatok mennyiségének és az adatforgalomnak minden szempontból hasznos növekedése azonban az elmult években fokozatosan felszintre hozott egy sor problémát is, melyek közül a leglényegesebbek a következők:
- nincs kellő áttekintés az átadott adatok további sorsáról, felhasználásáról;
- nincsenek rögzitve az adatok átadásának feltételei, elvei;
- tul bonyolult az adatállományok nyilvántartása, de egyuttal még sincs azokról megfelelő tájékoztatás;
- nem megoldott az adatok utólagos javitása, illetve annak dokumentálása;
- nem biztositott az adatállományok hosszutávu megőrzése, stb.

A problémák elemzése kimutatta, hogy ezek jelentős része megoldható a 2/1977 /VII.30./ KSH sz. rendelkezés korszerüsitése, átdolgozása utján. Javaslatot tettünk tehát egy, ennek helyébe lépő uj rendelkezésre.
Jogszabállyal azonban nem minden baj orvosolható. A koordináció változatos eszköztárának több egyéb elemére is szükség van, amelyek közül - kivül esvén témakörünkön - csupán a kölcsönös adathasznositásban érdekelt szervezetek két és többoldalu megállapodásait, szoros együttmüködését emlitjük.

A nemzetközi tapasztalatok

A munka megalapozása érdekében az előkészités során igyekeztünk megismerni a nemzetközi tapasztalatokat, azt, hogy az egyes országokban miként szabályozzák a statisztikai adatok számitástechnikai feldolgozásával kapcsolatos tevékenységeket.
Vizsgálódásaink során sok országban találkoztunk hasonló jogi előirásokkal. Általános tapasztalat, hogy a jogi keretek a kapitalista illetve a szocialista országokban lényeges eltérést mutatnak.

Mig az előbbi országcsoport általában átfogó "adatvédelmi" szabályozást dolgozott ki, addig a szocialista országokban hozzánk hasonlóan részleges előírásokkal találkozhatunk.
Az adatok rögzitésével és feldolgozásával kapcsolatban az általunk javasoltak a megismert külföldi gyakorlattal, szabályozásokkal összhangban vannak, sőt a KSH által megjelentetett, az adatgyüjtéseket ismertető katalógus tekintetében a legtöbb ország előtt járunk. Néhány esetben azonban találkoztunk szigorubb, több megkötöttséget tartalmazó előirásokkal is.
Az NDK-ban és Csehszlovákiában a számitógéppel feldolgozott beszámolójelentések rögzitése kötött, szabványos formában történik. Ennek megfelelően a kérdőivek felépitésére is egységes követelmények érvényesülnek.
A lengyel statisztikai rendszer az egycsatornás adatszolgáltatás megvalósitására törekszik. Az NDK statisztikai törvénye explicit módon tiltja, hogy a központi statisztika beszámolójelentéseit más szervek rögzitsék. Ez gyakorlatilag mindkét esetben azt jelenti, hogy más felhasználók feldolgozási igényüket csak az elrendelőn keresztül, a mágnesszalagon lévő állomány átvétele utján elégithetik ki.
A statisztikai adatok tárolását /megőrzését, archiválását/ és dokumentálását több országban szintén különböző jogi előirások szabályozzák. Ausztriában, NDK-ban, Lengyelországban hozzánk hasonlóan az irattározás, levéltározás előirásainak e területre vonatkozó konkretizálásával oldják meg a problémát.
Külön fel kell hivni a figyelmet egy, a legtöbb nyugat-európai országban érvényes előirásra. Az adatvédelmi törvények szerint a gépi adathordozón tárolt, személyekre vonatkozó adatokat tilos archiválni. A konkrét felhasználási cél teljesülése, vagy meghatározott idő letelte után az adatokat minden esetben törölni kell.
Az adatok átadása területén a nemzetközi helyzetet a következő néhány példával kivánjuk illusztrálni. Az 1978. évi osztrák adatvédelmi törvény szerint a számitógéppel feldolgozott adatok átadása csak abban az esetben lehetséges, ha /1/ ezt a törvény kifejezetten megengedi, /2/ ehhez az érintett /akire vonatkozik/ irásban hozzájárul, /3/ biztositva van, hogy az érintett nem azonositható, /4/ az adatok banki pénzforgalom céljait szolgálják, /5/ az átvevő a Statisztikai Hivatal, ahol az adatokat személytelenül, kizárólag statisztikai célból használják fel.
Az Európa Tanács konvenciója szerint az adatok forgalmában is biztositani kell, hogy a személyi adatokat kizárólag eredeti, törvényes céljuknak megfelelően lehessen felhasználni.
A lengyel statisztikai törvény garantálja a statisztikai információkhoz való széles körü hozzáférést, amit kizárólag az állami és gazdasági titok, valamint a polgárok személyes jogi szférájának védelme korlátoz.
Az NDK-ban a felhasználók a központi adatgyüjtések adataihoz szintén csak az ÁKSH-tól juthatnak hozzá. Átvevőként kizárólag az adatszolgáltatók felügyeletét gyakorló minisztérium jöhet számitásba, kivételt csak néhány funkconális szerv /terv, pénzügy, munkaügy/ képez. Az adatok átadásáról és átvételéről az állami szervek egymással szerződést kötnek, amelyben az átvevő vállalja az adatok jövőbeli kezelésével kapcsolatos részletezett kötelezettségeket.
Rendelkezés-tervezetünkben az engedély-kérésnél és az átadás engedélyezésénél kötelezően szerepeltetett témakörök meghatározásánál nagyban támaszkodtunk az NDK előirásaira.

Az előkészitő munka menete, módszerei, tapasztalatai

Az előbbiekben ismertetett indokok alapján, amelyeket a feltárt nemzetközi tapasztalatok is elátámasztottak, a KSH vezetése 1982-ben elhatározta a 2/1977/VII.30./KSH sz. rendelkezés felülvizsgálatát és átdolgozását.
Annak érdekében, hogy a különböző állami szerveknél felhalmozódott tapasztalatokat és kialakult igényeket a munkában hasznositani lehessen, a Statisztikai Koordinációs Bizottság ideiglenes adatkoordinációs munkabizottságot hozott létre. Ez az uj rendelkezés előkészitésében folyamatosan részt vett. Az igazgatási statisztikai rendszerek gyakorlatának és a képviselt 11 állami szerv véleményének megismerése a KSH számára igen sok segitséget jelentett.
Az egyetértés az uj rendelkezés kibocsájtásának időszerüségével teljeskörü volt. A munkabizottság helyeselte a KSH elképzelései szerinti egyszerüsitéseket, az adminisztráció csökkentését. Néhányan viszont - az általuk képviselt állami szerv sajátosságaira, gyakorlatára tekintettel - tul sok megkötöttséget tartalmazónak itélték meg az adatok átadására, továbbadására vonatkozó korlátozásokat, tul hosszunak tartották bizonyos adatállományok megőrzésének időtartamát.
A munkabizottság müködésének eredményeire is támaszkodva, a KSH főosztályvezetői értekezlete 1983. juniusában elfogadta az uj jogszabály azon változatát, amely az előadás irásának időpontjában hivatalos véleményezésen van az érintett szerveknél, sőt azok legnagyobb részével az egyeztetés már be is fejeződött. Számitani lehet arra, hogy az uj jogszabály az elképzeléseknek megfelelően még 1983-ban megjelenik.

Az uj szabályozás alapelvei

A hátralévő végső egyeztetések során az uj szabályozás alapvető elvei, előirásai már feltehetőleg nem fognak változni. A következőkben áttekintjük, hogy az uj rendelkezés milyen, a korábbitól eltérő lényeges mozzanatokat tartalmaz:

1. A számitástechnika statisztikai alkalmazásával összefüggő tevékenységek szabályozásának hatályát kiterjeszti minden szervre, amely ilyen tevékenységeket végez.
2. Kifejezésre juttatja a statisztikai adatokat tartalmazó iratok tartalmától és minősitésétől függő és a számitástechnikai rendszerek általános titok-, vagyon- és tüzvédelméről szóló hatályos jogszabályok, iratkezelési szabályzatok alkalmazásának jelentőségét, a szabályozott statisztikai tevékenységekkel való összefüggéseit.
3. A korábbi kettős - központi és helyi - alakilag is egységes nyilvántartást és dokumentációt megszünteti. Helyette a tárolt statisztikai adatállományokról - a segéd és másodlagos adatállományokat is ide sorolva - a tárolás helyén ir elő tartalmilag egységes dokumentációt és nyilvántartást.
4. Biztositja az érdekeltek széles körü tájékoztatását a KSH által kiadásra kerülő, az összes statisztikai adatgyüjtést tartalmazó katalógus megfelelő kiegészitésével. A jövőben ebben szerepelni fog, ha az adott adatgyüjtés adatait valamely szerv számitástechnikai eszközökkel feldolgozza és az igy átvehető.
5. Kiterjeszti és a fontosabb adatállományok esetében minimum 15 évben határozza meg a megőrzési időt. A szabályok egyuttal biztositják, hogy a történeti értéket képező adatállományok 15 év elteltével levéltári megőrzésre kerüljenek.

6. Szabályozza és egységesiti az adatátadás-átvétel engedélyezési, végrehajtási eljárását, a jogokat és kötelezettségeket, erősitve a magánszemélyek és a jogi személyek érdekeinek védelmét is. Lényeges előirás az átvett adatok továbbadásának és a magánszemélyekre vonatkozó, személyi azonositóval ellátott statisztikai adatok átadásának megtiltása, valamint a szervezetek azonositóval ellátott adataihoz való hozzáférés korlátozása.
7. Kifejezésre juttatja, hogy a számitástechnikai és a járulékos munka, illetve szolgáltatás költségeinek a megtérítésére az adatfeldolgozásra megbizást adó szerv kötelezett.

További feladatok

Az államtitkári rendelkezés megjelenésével a szabályozási tevékenység nem zárul le. A rendelkezés által szabott kereteken belül az adatátadásokkal kapcsolatos tevé nységeket és jogköröket minden ilyen tevékenységet folytató állami szervnek, igy a KSH-nak is ügyrendben kell rögzitenie. Az ügyrend készitését és annak a felhasználókkal való megismertetését azért tartjuk szükségesnek, mert csak igy látjuk biztositottnak az adatforgalom zökkenőmentes lebonyolitását. A felhasználó már igényének felmerülésekor tudja, hogy az adott állami szerven belül kihez kell fordulnia, milyen feltétellel számithat az adatok átvételére.
A jogszabály előkészitésével párhuzamosan, tulajdonképpen annak elébe menve kétoldalu kapcsolatokon keresztül megkezdtük és jelenleg is folytatjuk azoknak az információknak a begyüjtését, amelyek majd az adatgyüjtési katalógsuokban fognak megjelenni. Ezek arra vonatkoznak, hogy az egyes állami szervek milyen saját és átvett adatállományokat tárolnak mágneses adathordozón. Ahol lehet, az adatállományok konkrét lelőhelyét /pl. számitóközpont/ is közölni kivánjuk.
Lényeges feladat lesz a továbbiakban a jogszabály érvényesülésének ellenőrzése és támogatása. Az előbbit a KSH Ellenőrzési osztálya, az utóbbit a KSH adatforgalmának koordinálása céljából létrehozott Információszolgálat végzi.

Fontosabb források:

1. Az 1982. évi lengyel statisztikai törvény
2. Iszkowszki, J.: Az uj állami statisztikáról. Wiadomosci Statyczne, 1983. május
3. Az NDK statisztikai és számviteli törvénye
4. Rendelet a statisztika és számvitel szabályszerüségéről /NDK/
5. Ajánlás a központi állami szervek közötti mágnesszalagos adatcserére vonatkozó megállapodásra /NDK/
6. Az Országos Statisztikai Hivatal határozata a társadalmi-gazdasági információk irásos adathordozóinak módositásáról és gépi adathordozókra való viteléről /Csehszlovákia/
7. A japán statisztikai törvény és végrehajtási utasitásai
8. Az Egyesült Királyság hivatalos statisztikai szolgálata /tájékoztató anyag/
9. Az osztrák és NSZK adatvédelmi törvények

10. Javaslat az Egyesült Királyság adatvédelmi törvényére
11. Az Európa Tanács konvenciója automatikus eszközökkel feldolgozott személyi adatok védelmére
12. Edgar Dunn: Social Information Processing and Statistical Systems, New York, 1974.

Kovács András-Sugár Péter-Vig Ervin-Kormos Kornélia-
Apor György:

KSH Államigazgatási Számítógépes Szolgálat

Az IFP szöveges információkezelő rendszer és alkalmazásai

A szöveges rendszerek logikai alapegysége a dokumentum. A dokumentum mezőkből áll, amelyek az adatokat - az esetek nagy részében - kódolatlan formában tartalmazzák. A mezők csoportositása fix, és egy adatbázisban csak egyféle dokumentumtipus lehet. A dokumentumok egyenrangúak, közöttük semmiféle strukturális kapcsolat nem áll fenn. A feldolgozás fő célja a dokumentumok egy olyan részhalmazának visszakeresése és megjelenitése, amelyben a mezők értékei megfelelnek a kereső kérdésben specifikált összetett feltételeknek.

Mivel a dokumentumok egyenrangúak, szükségtelen rendezési kulcsokat képezni, nem kell az adatokat kódolni, hanem elegendő az ember számára a legérthetőbb /egyben a gép számára legbonyolultabb/ formában az élő beszéd szavainak képében tárolni. Különösen előnyös az könyvek, cikkek, szakirodalmi tájékoztatók vagy akár személyek esetében, ahol a hagyományos kódolás csak igen nehézkes módon volna lehetséges. Gondoljuk csak el milyen egyszerüvé vált igy egy könyv tartalmát néhány keresetlen szóvel vagy akár egy kis rövid összefoglalóval jellemezni. Megtartva az általánosság igényét példánkat folytathatjuk a könyvnél. A könyvről szóló dokumentum egyéb mezője lehet még a szerző vagy a szerzők neve, a mü cime, a megjelenés évei stb...

A visszakeresés tipikus esete: melyek azok a könyvek, amelyek tartalma ezzel és ezzel a szóval jellemezhető, szerzője ez és ez stb... További szükitő feltételként szerepelhet még a megjelenési intervallum, a hozzáférhetőség nyelve és igy tovább.

A jellemző szavakat /deszkriptorokat/ az adatbázis invertált file-ban tárolja. Egy szóhoz hozzárendeli mindazon dokumentumok azonositóit, amelyek ezt a szót tartalmazzák. Igy az összetett feltételü keresés ezeknek az azonositó halmazoknak a megfelelő összeválogatását jelenti. Az invertált file-ok alapján történő visszakeresést elsődleges visszakeresésnek nevezik. További szürés végezhető még azon mezők alapján, amelyekből nem készült invertált file. Ez a szürés már szekvenciális és ugyancsak összetett feltétel alakjában fogalmazható meg. Ezt nevezik másodlagos visszakeresésnek.

A szöveges rendszereknek két jellemző irányzata terjedt el
- a kötött és szabad szöveges rendszerek.

A kötött szöveges rendszereknél a deszkriptorokat diszkrét szavak, esetleg kifejezések formájában kell megadni. Szabad szöveges rendszereknél az élő szöveg minden szava deszkriptornak minősül. Ez utóbbinál az adatok előkészitése egyszerübb, az adatbázis szerkezete bonyolultabb és különösen a ragozott nyelveknél nehéz a kereső kérdést jól megfogalmazni.

Az IFP /Invertált File Processzor/ adatbázisának alapegysége a dokumentum. A dokumentum egy könyv, egy cikk, egy személy stb... jellemző adatait tartalmazza. A dokumentum mezőkből áll. Egy mező tartalma lehet pl. könyv esetén a könyv cime vagy tartalma /Annotáció/, szerzőjének neve, megjelenésének éve, a tartalomra jellemző szavak stb... Ez utóbbit deszkriptoroknak nevezzük. Ebből látszik, hogy az IFP, u. n. kötött szöveges adatbáziskezelő rendszer. Az adatbázis file-jai a dokumentumokat és azok mezőit a visszakeresés és a megjelenités szempontjából célszerü módon tárolják.

Bizonyos mezők tartalmából készithetünk invertált file-okat. Ezek a file-ok egy mező egy bizonyos értékéhez kapcsolva tartalmazzák azon dokumentumok azonositóit, amelyek ezt az értéket az adott mezőben tartalmazzák. Invertált file-ok alapján elsődleges visszakeresést végezhetünk, és az olyan mezőt, amelynek tartalmából invertált file készült, elsődleges mezőnek nevezzük. Ilyen minden esetben a deszkriptorokat tartalmazó mező, de ilyen lehet pl. a szerző neve is könyv esetén, vagy minden olyan mező, amit az adott alkalmazási rendszer szervezője elsődlegesnek definiál. Az elsődleges visszakeresés kérdését a Boole algebra szabályaiból ismert és, vagy, nem operátorokkal lehet felállitani.

Az adatbázis logikai szekciókra bontható. Létrehozásakor minden dokumentumhoz hozzárendelhetünk egy Ø és 4Ø95 közötti szekciószámot. Elsődleges visszakeresésnél megadható az u.n. szekció feltétel, mely szerint az elsődleges visszakeresés feltételrendszerén túlmenően csak azok a dokumentumok relevánsak, amelyek szekciószáma a megadott intervallumok valamelyikébe beleesik. Ez a lehetőség egy igen gyors szürést tesz lehetővé.

Az elsődleges visszakeresés eredménye egy eredményhalmaz, amelyben további szürés - másodlagos visszakeresés - végezhető. A másodlagos mezők értékeiből nem készült invertált file. Ezeket az értékeket a rendszer dokumentumonként egyenként végigvizsgálja. A másodlagos visszakeresés kérdését is a Boole algebra szabályaival lehet felállitani. Az elsődleges és a másodlagos visszakeresés egyetlen visszakeresési ciklusban is megadható.

A visszakeresés lépcsőzetesen is végezhető. A megelőző eredményhalmaz szükithető vagy bővithető. Ha a keresőkérdést túlspecifikáltuk, akkor a közvetlenül megelőző állapothoz vissza tudunk lépni, és uj kérdést tehetünk fel.

A másodlagos visszakeresés után kellőképpen leszükitett eredményhalmazt megjelenithetjük. Ezt nevezzük harmadlagos visszakeresésnek. Harmadlagos mezőt csak megjeleniteni lehet.

Másodlagos mezőt is lehet megjeleniteni. Ha ilyen szándékunk az elsődleges mezőkkel kapcsolatban is van, akkor az invertált file-okon kivül harmadlagos módon is tárolhatjuk. Meg lehet jeleniteni egy dokumentum megjelenithető mezői közül az összeset vagy a tetszés szerint kiválasztottakat.

A harmadlagos mezők formázott módon is tárolhatók. Egy egy közbeiktatott !-jel azt jelenti, hogy megjelenitéskor uj sor következik.

Az IFP adatbázishoz tartozhat még egy monohierarchikus tezaurusz is, mely tetszés szerint megjelenithető. Egy fogalomnak lehet egy generikusa /főlérendeltje/, több specifikusa /alárendeltje/, több szinonimája és több rokona.

A rokoni kapcsolat kommutativ, de nem tranzitiv. Elsődleges visszakeresésnél a kereső deszkriptorhoz opcionális módon hozzárendelhetjük azok szinonimáit is.

Az IFP-ben az adatbiztonság dokumentum és mező szinten valósitható meg. Minden dokumentumhoz, mezőtipushoz és felhasználóhoz hozzárendelhető egy Ø és 63 közötti szám. Ha a felhasználó biztonsági kódja kisebb, mint a dokumantumé vagy a mezőé, akkor az számára nem létezik.

Az IFP lehetőségei és elvi felépitése megegyezik a fejlett szöveges információkezelő rendszerekével - STAIRS, GOLEM, MISTRAL - de további előnyös tulajdonságokkal is rendelkezik:

- Az Államigazgatási Számitógépes Szolgálat /ÁSZSZ/ Honeywell számitógépén kis memóriában, gyorsan, olcsón üzemeltethető,
- terminálhálózaton keresztül hozzáférhető,
- használata rendkivül egyszerü.

Nézzünk meg egy alkalmazási példát.
Keressük meg azokat a cikkeket, amelyek Magyarország valamint a Szovjetunió és a KGST kapcsolataival foglalkoznak és magyar vagy orosz nyelven hozzáférhetők.

```
SEARCH ("MAGYARORSZAG" OR "SZOVJETUNIO") AND "KGST"
SELECT NYELV = "MAGYAR" OR NYELV = "OROSZ"


*IFP/RETR
>>RETRIEVE ENGAGED ,
>>SEARCH ENGAGED          03/02/32    16:47:48:512

  MAGYARORSZAG     000030
  SZOVJETUNIO      000507
  KGST     000201
     MINTERM 01 =  000036

>>RESULT AT END OF SEARCH =    000036
>>SELECT ENGAGED          03/02/82    16:47:49:477
>>RESULT AT END OF SELECT =    000011
>>RESULT FILE = R9
>>RETRIEVE FINISHED       03/02/82    16:47:50:116
>>RETRIEVE FINISHED
```

Az adatbázis 6000 cikk adatait tartalmazza. Ebből 507 szól a Szovjetunióról, 30 Magyarországról és 201 a KGST-ről. A Szovjetunióval vagy Magyarországgal és a KGST-vel 36 cikk foglalkozik. Ebből tizenegynek a nyelve magyar vagy orosz.

Az eredmény nem egész 2 másodperc alatt megjelent a képernyőn.

Nézzük meg e 11 cikkből az első kettő nyelvét és rövid tartalmi kivonatát:

```
FIELD NYELV, ANNOTACIO
ITEM 1 - 2

*IFP/DISPL
IFP PARLIBRARY - OGYK TESZT    03/02/82   DISPLAY

 000161      K003627
  A MU NYELVE
  OROSZ
  TARTALMI KIVONAT
   1976-80 SZOCIALISTA GAZDASAGI INTEGRACIO MECHANIZMUSA, SZOVJET
   OSSZEFOGLALO ERTEKELES, TERVKOORDINACIO ES ARUVISZONYOK,
   PENZVISZONYOK SZOCIALISTA GAZDASAGBAN, TUDOMANYOS, TECHNIKAI,
   TERMELESI EGYUTTMUKODES SZOCIALISTA INTEGRACIOBAN, SZOCIALISTA
   INTEGRACIO NEMZETKOZI SZERVEI ES KGST NEMZETKOZI KAPCSOLATAI,
   SZOVJETUNIO KULGAZDASAGI SZERVEINEK IRANYITASA, KGST TAGORSZAGOK
   GAZDASAGI KAPCSOLATAI FEJLETT TOKES, FEJLODO ORSZAGOKKAL,
   SZOCIALISTAKONYV

 000652      K003660
  A MU NYELVE
  OROSZ
  TARTALMI KIVONAT
   1973-80 KGST, TERMELES KONCENTRALASA ES SPECIALIZALASA, 1977-80,
   NDK ERTEKELES, NDK ES SZOVJETUNIO, EXPORT, IMPORT ARUSZERKEZETE,
   STATISZTIKA, 1976-80, NDK ES KELET-EUROPAI SZOCIALISTA ORSZAGOK
   ARUCSEREFORGALMA, 1973-76, SZOCIALISTASAJTO
```

Egyéb alkalmazási lehetőségek, amelyekre a rendszert kipróbáltuk:

- tudományos cikkek gyüjteménye a számitógépes szimuláció témaköréből;
- kóreset- és betegnyilvántartás;
- szabványok nyilvántartása;
- jogszabálynyilvántartás

Csicsman József: A SOLAR Adatszótára

Központi Statisztikai Hivatal

A KSH Számitóközpontjában fejlesztettük ki a Statisztikai On-line Adatbázislekérdező Rendszert a SOLART-t.

A rendszer alapvető célja, hogy a KSH adatbázisrendszerének a leggyakrabban használt részét interaktiv módon elérhesse a statisztikus felhasználó. Az adatok a rendszerbe a több éves tapasztalatok alapján kerülnek. Csoportositásuk alapvető szempontja az összehasonlithatóság, azaz az, hogy általában az adatok nem rendezhetők hierarchikus rendbe.

A feladat specialitása miatt nem volt lehetséges az ismert adatbázislekérdező rendszerek bevezetése, azaz miképp a teljes rendszert, az annak szerves részét képező Adatszótárát ujszerü alapokon kellet létrehozni.

A SOLAR felhasználója statisztikai fogalmak segitségével kommunikál a rendszerrel. Az adatbázis lekérdezése és karbantartása egymástól minden szinten független módon történik. Az adatelérés nem rekordszinten, hanem a relációs modelleknek megfelelően modul szinten történik. /A modul egy adott statisztikai mutató adatainak összessége./ A lekérdezések hatékonysága legnagyobb mértékben az adattár és a szótár elérésétől függ.

Az imént felsorolt feltételek határozták meg az Adatszótár tervezését, mely két egymástól elkülönülő részből a Logikai és a Fizikai Adatszótárból áll.

A Logikai Adatszótár megfelel a SOLAR metainformációs rendszerének, mely létrehozása és karbantartása batch üzemmódban történik. Az alrendszer 16 fogalomra épülő rekordtipust és azok kapcsolatait kezeli, s eredményül a hibátlan rekordokat tartalmazó un. R- és a rekordok kapcsolatait leiró K-file-t szolgáltatja. Ennek a rendszerkomponensnek összefüggésellenőrző és a file-ok karbantartó részének programozását a SZÁMKI munkatársai készitették el.

A Fizikai Adatszótár a már on-line környezetben használt Adatszótárból és az un. Modul-cimtárból áll.

Az Adatszótár speciális szerkezetü adatállomány, mely tartalmazza az egészében hálószerkezetnek megfelelő szótárrekordokat és az elérést meggyorsitó indextáblákat.

A szótárrekordok 18 tipusba sorolhatók, mely tipusok az egyes statisztikai fogalmaknak és a tárolt adatmoduloknak feleltethetők meg. Minden rekord két részből áll

- az egyik az információs rész, mely tartalmazza az adott adatbázis elem felhasználói és az adatlekérdezéshez szükséges információit
- a másik a kapcsolatrész, mely azt adja meg, hogy az adott rekord, mely más rekordokkal van kapcsolatban, azaz ezekkel a részekkel adható meg az adatbáziselemek kapcsolatrendszere.

A Modul-cimtár az adatmodulok fizikai cimeit tartalmazza, mely elérése az Adatszótáron keresztül történik. Rekordjai az egyes adatok időszakonkénti csoportositásában adják meg, hogy az Adatszótárban a definiált időszakokra létezik-e egy adott modul és ha igen, akkor az milyen fizikai cimen található.

A Fizikai Adatszótár létrehozása a Szótár-betöltő és cimtár inicializáló, a Modul-cimtár karbantartása az Adatbetöltő alrendszer feladata.

Az Adatszótár használata a SOLAR Kommunikációból felhivható Runtime rutinok és az Adatbetöltés adatmodul információ lekérdező rutinja segitségével történik. A Runtime rutinok két részre, a felhasználó informálódását elősegitő és a lekérdezés végrehajtását támogató rutinokra bonthatók.

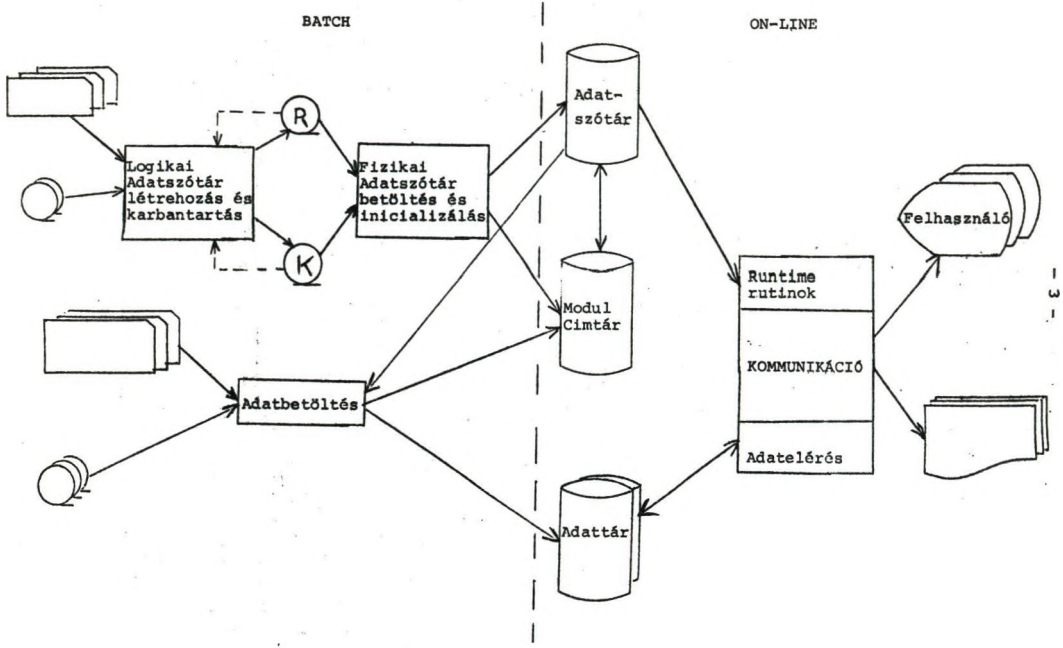
A felhasználó informálódását a SOLAR Kommunikáció külön részeként müködő un. Tájékozódás segiti elő, mely a munkavégzés bármely fázisában felhivható. A Tájékozódás Runtime rutinjai az adott rekord - adatbázis elem - szöveges leirását és a vele kapcsolatban levő többi elem tipusát és azonositóját adja meg. A rekordinformáció és a kapcsolat lista egymás utáni kérésével egyszerü

módon a teljes Adatszótár bejárható. Külön lehetőség az ugynevezett Vezérszavas keresés az egyes statisztikai mutatók "megtalálását" szolgálja, oly módon, hogy a mutatókat jellemző vezérszavak halmazán ad lehetőséget müvelet végzésre.

A lekérdezés végrehajtását támogató Runtime rutinok, melyek száma 12, a Kommunikáció egyes fázisaiban szolgáltatják a végrehajtást kiszolgáló speciális információkat.

Az itt igen szük keretekben ismertetett rendszerben elértük, hogy az Adatszótár bármely adata maximum két olvasással elérhető, és sikerült a szükséges tárolóhely minimalizálása is, amely fontos technikai feltétel volt ugyanis a jövőben cél, hogy a statisztikai adatbázisrendszer egyre nagyobb része elérhető legyen interaktiv módon.

A mellékelt ábra a SOLAR egyes rendszerkomponenseit és azok kapcsolatát szemlélteti.



A SOLAR rendszer vázlata

Petényi Andrea Ingeborg, SZKI

TELEDATA rendszer - általános ismertető

1. Bevezetés

Világviszonylatban uj tipusu számitástechnikai eszköz és szolgáltatás hódit, mely Angliában PRESTEL, a Német Szövetségi Köztársaságban BILDSCHIRMTEXT, más európai országokban VIEWDATA néven terjedt el és vált az üzleti és gazdasági - sőt számos helyen a privát - élet mindennapi eszközévé. Magyarországon ez a rendszer és szolgáltatás TELEDATA néven kerül bevezetésre.

Milyen uj lehetőségeket nyujt a TELEDATA?

A TELEDATA rendszer lehetővé teszi, hogy egyszerü TV-szerü terminálokról nyilvános póstai telefonhálózaton keresztül u.n. menüszerü lekérdezéssel számitógépes szolgáltatások választhatók, adatbázisok információi lekérdezhetők, megjelenithetők, esetleg módosithatók.

Hangsulyoznunk kell, hogy a TELEDATA a számitógépes adatbázisok kezelésének egy ujszerü lehetőségét biztositja. Az adatbázisoknak a TELEDATA rendszeren keresztüli kezelése rendkivül egyszerü, az adatok információhordozó erejét megsokszorozzák a szinek és grafikus lehetőségek. Minimális számitástechnikai ismeretekkel rendelkező egyének is könnyedén elsajátithatják használatát. A terminálok, melyekre a TELEDATA rendszer támaszkodik, Európa nagy területén csereszabatosak és természetesen jóval olcsóbbak, mint a hagyományos terminálok.
E tulajdonságok alapján számos olyan területen is lehetővé teszik a számitógépes szolgáltatások - köztük az adatbázisok - elérését, ahol a korábbi eseteken belül még nem volt mód a korszerü információs technika módszereinek alkalmazására.

A fennt vázolt feladatok végrehajtását, megvalósitását támogatja az SzKI-nál kifejlesztett eszköz és programrendszer, amely SZKITA néven került bevezetésre.

2. Az R-15/16 alapu zártkörü TELEDATA rendszer

Intézetünkben '82-ben indult meg az R-15/16 számitógép köré kiépitett zártkörü TELEDATA rendszer - SZKITA - kifejlesztése.

A rendszer kialakitásakor kettős cél lebegett előttünk:

- A rendszer hardware/software komponensei kulcsrakészen legyenek telepithetők, bármely megfelelő távadatfeldolgozási /TAF/ lehetőségekkel ellátott ESzR felhasználói környezetbe;
- A rendszer, mint szolgáltatás legyen igénybevehető az SzKI számitógépein.

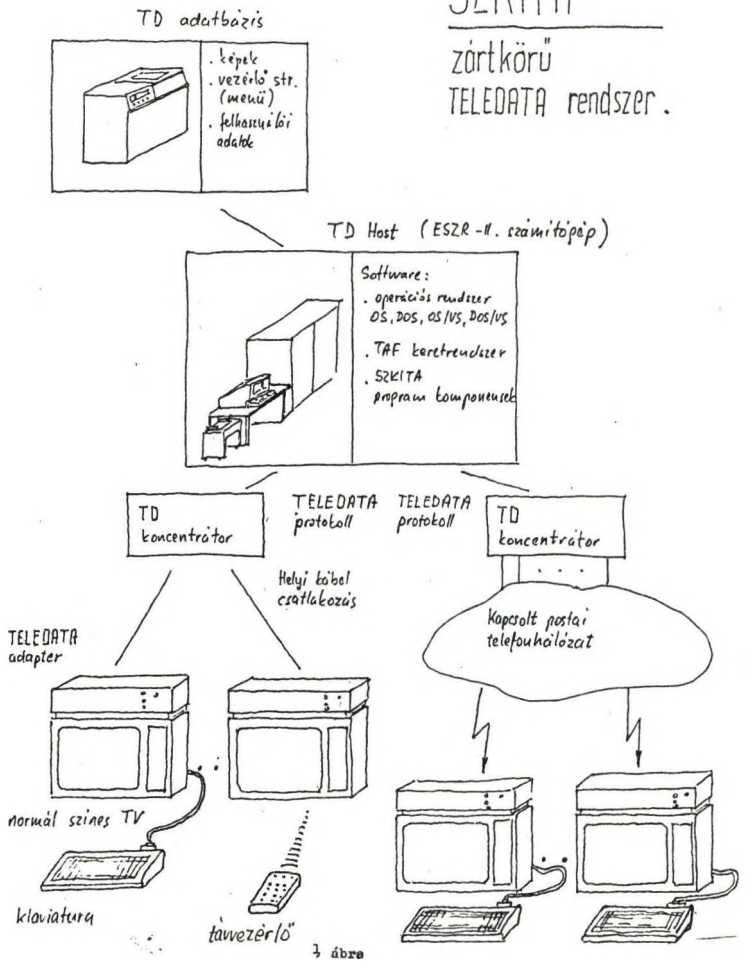
2.1. A SZKITA hardware eszközei

Az SzKI zártkörü TELEDATA rendszere az R-15/16 számitógépen került először alkalmazásra, de más ESzR sorozatu számitógépre is egyszerüen adaptálható.
Milyen eszközök szükségesek ahhoz, hogy a SzKITA rendszer valamely felhasználói igény kielégitésére konkrét alkalmazói környezetben kiépitsük?

. Teledata terminál
. Teledata koncentrátor
. telefonvonal
. Teledata szolgáltatást nyujtó számitógép /1. ábra/.

SZKITA

zártkörű
TELEDATA rendszer.



1 ábra

2.1.1. A TELEDATA terminál

A TELEDATA terminál az alábbi elemekből áll:

. Szines televizió készülék /pl. ORION/, amelynek segitségével a felhasználó nemcsak a normál TV müsor vételére képes, hanem igénybe veheti a TELEDATA szolgáltatásokat is.

. Viewdata adapter, amely magában foglalja a TELETEXT és a TELEDATA üzemmódot biztositó kiegészitő áramköröket, tápegységet és a telefonvonalra való kapcsoláshoz szükséges beépitett modemet /ORION termék/.

. Távvezérlő, mely lehetővé teszi vezérlő információk és parancsok, valamint numerikus input adatok bevitelét.

. Alfanumerikus billentyüzet, mellyel igény szerint ki lehet egésziteni a TD-terminált, s amely a teljes ABC-nek megfelelő alfanumerikus bevitelt teszi lehetővé /64 billentyüs változat, SzKI termék/.

A TD-terminál olcsó, ára kiépitéstől függően harmada, negyede egy hagyományos terminál árának.

2.1.2. A TELEDATA koncentrátor

A TD-koncentrátor az SzKI-nál kifejlesztett Teledata célberendezés, amely az IBM 3274-es control unit emulációjára támaszkodva szinkron adatátviteli vonalon kapcsolódik a TELEDATA feldolgozást végző számitógéphez. A számitógép és a koncentrátor közti logikai kapcsolatot az európai normákkal egyező BTX protokoll biztositja. A protokoll kompatibilis

a nyilvános TELEDATA rendszerben a központ és a külső csatlakozó számitógépek között rögzitett protokollal.

A koncentrátor egyidejüleg 16 TD-terminál kiszolgálására képes, melyek helyi kábelen, vagy kapcsolt telefonhálózaton keresztül kapcsolódnak hozzá.
Az adatátviteli sebesség a terminál felé 1200 vagy 4800 baud, a számitógép felé 75 baud.

3. A SZKITA programrendszer

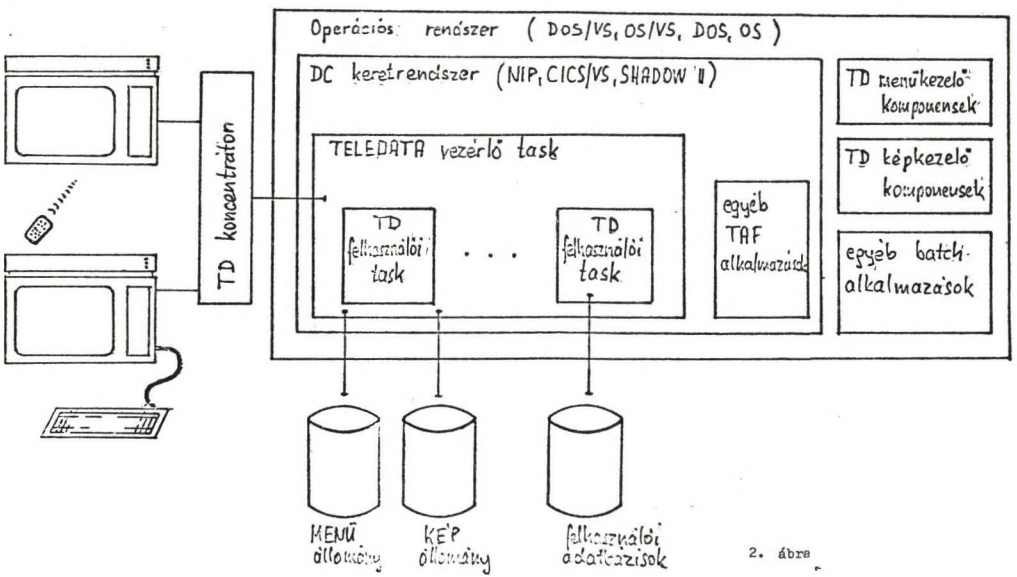
A SZKITA rendszer célja a TD funkciók megvalósitása, alkalmazói interface képzése. A programrendszer a korábban vázolt hardware eszközökkel az operációs rendszer /pl. DOS/VS/ és az adott alkalmazói környezetben müködő TAF keretrendszer /pl. SHADOW, NIP/ szolgáltatásaira épülve oldja meg a feladatot. Ennek során a klasszikus rendszer lehetőségeit a TD müveletek elvégzésének lehetőségével bővíti ki. A programrendszer felépitését a 2. ábra szemlélteti. Az ábrán feltüntetett rendszerben a TELEDATA szolgáltatás nem jelent kizárólagos szerepet, éppen ellenkezőleg, lehetővé teszi a távadatfeldolgozási alkalmazások egyidejü jelenlétét, üzemeltetését.

A TELEDATA alkalmazások kialakitását, üzembevitelét az alábbi komponensek teszik lehetővé:

. TELEDATA vezérlőprogram, amely a menü és a képek alapján gondoskodik a felhasználói választásnak megfelelő képek megjelenitéséről, felhasználói tranzakciók inditásáról, ezenkivül a felhasználói tranzakciókban igényelt TELEDATA I/O müveletek végrehajtásáról.

A TELEDATA vezérlő struktura /menü/ a vezérlési séma /gráf/, amely alapvetően meghatározza egy alkalmazás működési mechanizmusát.

SZKITA programrendszer



2. ábra

A vezérlő gráf csomópontokból és az ezeket összekötő vezérlési utakból áll. Egy csomópontot a tetszőlegesen választott azonositó, vagy a generálásnál kapott numerikus érték egyértelmüen meghatároz.
A csomópont, funkcióját tekintve:

. belépési pont jellegü /gate tipus/ - szerepe az alkalmazás azonositása, titkositás, speciális képek /üdvözlő-, bucsu-kép/ megjelenitésének hozzárendelése;

. informativ jellegü /menü tipus/ - a vezérlést kapva megjelenit egy csupán statikus /állandó/ információt tartalmazó képet; a választékban definiálja a vezérlésátadás irányát;

. aktiv jellegü /tranzakció tipus/ - a vezérlést kapva tranzakciót indit, az inditást megelőzően vagy a tranzakcióból egy dinamikus /változtatható/ adatmezőket tartalmazó képet küld. A tranzakció kijelölheti a vezérlésátadás irányát.

A választékban a vezérlő gráf bármely csomópontja megcimezhető.
A vezérlő struktura ezáltal lehet hierarchikus, vagy háló tipusu.

. A TELEDATA vezérlési struktura - menü - kialakitását, változtatását és archiválását végző programok.

. TELEDATA képek előállitását, változtatását és archiválását végző programok.

A SZKITA programrendszer adatai három adatbázisban helyezkednek el:

. A kép file, amely a terminálon megjelenithető képeket és az azokat leiró információkat tartalmazza.

. Menü file, amely tárolja a vezérlési strukturát, mely a képek egymásközti és a képek és tranzakciók közti kapcsolatot irja le.

. A harmadik a felhasználói adatbank, melynek tételeit a felhasználók lekérdezhetik, tranzakciók segitségével módosithatják.

Az alkalmazások kialakitása a SZKITA rendszerben három egymástól jól elkülönülő tevékenységből áll:

- az adott TELEDATA alkalmazás vezérlő strukturájának /u.n. menüjének/ meghatározása,
- felhasználói programok /tranzakciók/ előállitása az adott alkalmazáshoz tartozó adatok eléréséhez,
- a TELEDATA terminálon megjelenő képek összeállitása.

Ha az alkalmazás megfogalmazható olymódon, hogy egy felhasználói választástól függően csak a megfelelő informatív kép megjelenitésére van szükség, akkor a tranzakciók előállitása elmarad. Igy egy adott alkalmazásra vonatkozóan a rendszerprogram/alkalmazói program arány a hagyományos technikával megvalósitott rendszerrel szemben a rendszerprogram javára változik. Másképpen fogalmazva az alkalmazás kialakitásához szükséges funkciók nagyobb hányada érhető el "készen".

4. Professzionális személyi számitógépeken müködő zártkorü TELEDATA rendszer

Az előzőekben ismertetett alapelveket betartva Intézetünkben kifejlesztették a professzionális személyi számitógépen /PSzSz/ müködő zártkörü TELEDATA rendszert.
A PSzSz méreteinek megfelelően, szerényebb alkalmazások kialakitására megfelelő.
Az egyidejüleg kezelt terminálok száma: maximum 4.

A PSzSz-n alapuló alkalmazói rendszerek természetesen a protokoll betartása mellett kapcsolódhatnak a nyilvános TELEDATA rendszerhez.

Götl Győző - Farkas Gabriella
KSH Államigazgatási Számitógépes Szolgálat

PÁRBESZÉDES PROGRAMGENERÁLÁS ADATBÁZIS LEKÉRDEZÉSRE

1. Bevezetés

Az adatbázis lekérdezéseknél jogos igény, hogy bármilyen adatot el lehessen érni, ez az elérés gyors legyen, ne emésszen fel felesleges erőforrásokat és esetleg a felhasználó maga kezelje az egész eljárást. A Magyarországon használatos adatbázisokhoz általában van interaktiv lekérdező rendszer, de ezek a fenti igényeknek csak többé-kevésbé felelnek meg, mivel alkalmazásuk nagy erőforrást és speciálisan képzett programozókat igényel. Az előadás egy olyan rendszert mutat be, amely gépi erőforrást és jelentős programozói munkát takarit meg.

2. A gépi erőforrás-igény optimalizálása

Ha egy adatbázist ugy akarunk lekérdezni, hogy megfelelő paraméterezéssel bármely rekordot visszanyerhessük, akkor az azt megvalósitó program meglehetősen terjengős és sok ismétlődő részt tartalmaz. Másrészt az elkészült program futtatása nem gazdaságos, mert feleslegesen sok erőforrást igényel. /Pl. akkor, ha csak a hierarchia felső részén elhelyezkedő rekordokat akarjuk visszakeresni./
Ezzel szemben célszerü olyan programozási segédeszközt kifejleszteni, amellyel egyrészt min. memóriaigényü program állitható elő - lehetőleg interaktivan -, másrészt programozási ismeretekkel nem rendelkezők is használhassák. Azaz inputként használhatók legyenek a hétköznapi nyelvből vett szövegdarabok, outputként pedig adatbázis lekérdező utasitásokat, utasitáscsoportokat kapjunk, amelyekből végül komplett program álljon össze.

3. A programozói munka megtakaritása

Ilyen programozói segédeszköz megirására többféle lehetőség kinálkozik. Elterjedt software-es gyakorlat, hogy ilyenkor speciális uj rendszert hoznak létre, némi többletszolgáltatással. Viszonylag ritkán fordul elő, hogy már

meglévő software eszközökre támaszkodó megoldást keresnek.
Az előzőekben vázolt probléma megoldására is lehetett volna egy szövegkezelő rendszert kidolgozni, vagy egy preprocesszort irni. Ezzel szemben mi a makrogenerátornak mint szövegkezelőnek és preprocesszornak alkalmazása mellett döntöttünk. Makrogenerátor ugyan a legtöbb számitógépen van, de felhasználása a programozók körében nem szokásos. Ez azért is sajnálatos, mert a legegyszerübb adatfeldolgozó programok is tartalmaznak ismétlődő részeket. Ezeknek makrokkal való kiváltása mindenképpen élőmunka megtakaritásával járna.

4. Adatbázisbeli rekord visszakeresésére utasitás helyett makrohivás

Nézzük pl. az

láncot. Ha vissza akarnánk keresni az adott ország összes városát, azt a következő utasitásokkal tehetnénk meg:

```
MINDEN-VÁROS.
    FIND NEXT VÁROS WITHIN OV.
    IF DB-STATUS = 0502100 DISPLAY "ORSZÁG VÉGE"
    GO TO VEGE ELSE
    IF DB-STATUS NOT = o DISPLAY "HIBA AZ OV LÁNCON"
    DISPLAY DB-STATUS GO TO HIBAFELDOLGOZÁS.
    GET VÁROS.
```

Ugyanez makrohivással a következő lehet:

```
NEXT/MINDEN-VÁROS,VÁROS,OV,ORSZÁG,VÉGE,HIBAFELDOLGOZÁS/
```

vagyis csak az utasitás változó részeit kell leirni.
Példánkkal azt akartuk megmutatni, hogy adatbázist lekérdező programok irásakor érdemes makrokat alkalmazni, az általánosan használt makrodefiniciókat makrokönyvtárba tenni és a makrogenerátort mint preprocesszort alkalmazni.

5. Szöveges inputból makrohivás

A makrokönyvtárunkat más, programozási ismeretekkel, nem rendelkező felhasználó számára is elérhetővé tehetjük. Ehhez gondoskodni kell arról, hogy egy adott adatbázis-struktura adott szintjén a felhasználótól kapott szöveges információ alapján egy FIND makrohivás generálódjon. Ez a

- kérdéskiirás
- válaszfogadás
- válasz elemzés, FIND generálás
- folytatás, a következő kérdés kiirása

lánc automatizálását igényli.
Ennek a láncnak az elemeire makrokat lehet definiálni, amely definiciők makrohivásokat, főleg dinamikusakat tartalmazhatnak.
A mellékletben példán szemléltetjük a láncot megvalósitó programrészlet müködését.

6. Virtuális rekord kialakitásának lehetősége

Software-ünk felhasználási lehetőségeit tovább szélesiti az a tény, hogy saját programrészletek is beépithetők az interaktiv módon generált jobba az alábbi helyeken:

I-O SECTION-ben
DATA DIVISION-ben
file-ok megnyitásánál
az adatbázisstruktura bármely szintjén
az adatbázis rekord visszakereső utasitások után
a file-ok lezárásánál
az EXECUTE aktivitás file hozzárendeléseinél.

Lényeges szempont, hogy ezeken a helyeken elég megadni egy file leirást. /A Honeywell gépen ØØ SELECT-tel./ Gondoljunk arra, hogy milyen fontos ez. pl: DATA DIVISION-ben, ahol a virtuális rekordot definiáljuk, vagy az egész software interaktivitásának megitélésénél.
A saját programrészlet beépitésének a lehetősége biztositja azt is, hogy a visszanyert adatokon müveletek végezhetők, egyéb input file-ok is feldolgozhatók, az output pedig tetszőleges adathordozóra irányitható. Itt hivjuk fel a figyelmet arra, hogy a felhasználó a generált programot forrásnyelven saját területén tárolhatja, a későbbiekben pedig tetszés szerint tovább is fejlesztheti.

7. A módszer alkalmazhatósága

A módszer tetszőleges adatbázis lekérdező nyelvre, tetszőleges makrogenerátorral megvalósitható. A megvalósitás

mellett szól az is, hogy a software vásárlásoknál gyakran nem jut keret a lekérdezőrendszer megvásárlására, S ha van is ilyen lekérdező rendszer, azt még az adatbázissal dolgozó programozók közül is csak kevesen használják, mivel sajátos ismereteket feltételez.
Az adatbázis lekérdező rendszert Honeywell 66-os gépen IDS II adatbázisra készitettük el IMP-66 makrogenerátor alkalmazásával. Az IDS II a CODASYL koncepció alapján készült, ezért módszerünk könnyen adaptálható más, szintén ezen koncepció alapján készült adatbázisokra /pl: IDMS/.
Az a dán B.Svejgaard és P. Lindblad által GIER gépre készitett IMP-IV /Interpretetive Macro Processor/ meglehetősen pontos adaptációja. Ezt az adaptációt Zsombok Zoltán készitette.

A makrogenerátor adatbázisokon kivül egyéb programozási segédeszközök kidolgozásában is hasznosnak bizonyult. Többek között felhasználói rendszerek üzemeltetésére olyan job-inditó rendszert dolgoztunk ki, amelyet szöveges információ adásával a felhasználó maga kezelhet. Tapasztalataink szerint egyre több felhasználó igényli ezt az átfutási időt lényegesen csökkentő szolgáltatást. Makrogenerátor bármely programnyelvre preprocesszorként alkalmazható. /Lásd [2] -ben FORTRAN preprocesszor, mely ALGOL tipusu utasitások használatát teszi lehetővé/.

Irodalom

[1] Peter Lindblad: IMP-IV Kobenhavns Universitets Matematiske Institut Afd. for Inform. KUAIB 23.

[2] Zsombok Zoltán: IMP-66 1982.

Melléklet

IDS II adatbázisra IMP-66 makrogenerátorral kialakitott programozási segédeszközből a dialógust megvalósitó részlet:

```
'DIALOG'(CASE,99,,,,,
      (CMP,97,
         (CALL,ASK,"A FOLYTATAS TH (TALAJMINTA) VAGY TA (TALAJATLAG)
            VAGY TAK (ALLOKULTURA) VAGY TE (TECHNOLOGIA) LESZ ?")
            ,TM,TA,TAK,TE)
         (ADD,99,(CASE,97,256,30,31,32,33),99),,,,
         (CMP,97,
            (CALL,ASK,
               "A FOLYTATAS TMUV (TALAJMUVELET) VAGY "
                           "TRA (TRAGYAZAS) VAGY "
                           "NV (NOVENYVEDELEM) VAGY "
                           "GYB (GYEP BETAKARITAS) LESZ ?"),
            TMUV,TRA,NV,GYB)
         (ADD,99,(CASE,97,,30,31,32,33),99))
      (CMP,23,
         (CALL,ASK,"A KIVANT "
            (CASE,99,,
               MEGYE,
               GAZDASAG,
               TABLA,
               TALAJFELEMI,
               TALAJATLAG,
               ALLOKULTURA,
               "TECHNOLOGIA ALAP",
               TALAJMUVELET,
               TRAGYAZAS,
               NOVENYVEDELEM,
               "GYEP BETAKARITAS")
            " REKORDOT (VALASZ: OSSZES VAGY BIZONYOS)"),
         OSSZES,BIZONYOS)
      (CASE,23,,
         (CALL,SEAT,'INDENR'(CASE,99,,,13,15,17,18,14,16,32,33,34,35),
            R(CASE,99,,,13,15,17,18,14,16,32,33,34,35),
            (CASE,99,,,MG,GT,TM,TA,TA,TR,RM,R4,RM,RM),
            (CASE,99,,,MEGYE,GAZDASAG,
               TABLA,TABLA,TABLA,TABLA,
               TECHNOLOGIA,TECHNOLOGIA,TECHNOLOGIA,TECHNOLOGIA),
            (CASE,99,,,VEGE,
               MTNDENR13,MTNDENR15,MTNDENR15,MTNDENR15,MTNDENR15,
               MTNDENR16,MTNDENR16,MTNDENR16,MTNDENR16),
            VEGE),
         (CALL,ANY,BIZONYOSR(CASE,99,,12,13),
            R(CASE,99,,12,13),
            (CASE,99,,MEGYE,GAZDASAG),VEGE))
       (CALL,INCL,"SAJAT PROGRAMRESZLET AZ ELLEN SZINTEN:")
      (CMP,98,(CALL,ASK,KITRATJA?),IGEN)
      (CASE,96,,
         (CALL,DISP,(CASE,99,,12,13,15,17,18,14,16,32,33,34,35))))
```

Iványi Antal, Luderer Bernd, Szotnyikov A. N.
ELTE, Karl-Marx-Stadti Müszaki Főiskola, Moszkvai Állami Egyetem

Deszkriptoros információvisszakereső rendszerek paramétereinek optimális megválasztásáról

Az információvisszakereső rendszerek nagy részében /például bizonyos könyvtári rendszerekben/ a tárolt dokumentumok tartalmát deszkriptorok segitségével jellemzik. Az előadásban ismertetünk egy egyszintes és két kétszintes deszkriptoros rendszert, majd a hasonló rendszerek Saltontól [1] származó matematikai modelljét. Ezután a kétszintes rendszer paramétereinek megválasztására javasolt algoritmusokat mutatunk be, majd ezen algoritmusok hatékonyságát elemezzük szélsőséges esetekben, végül az egyik közelitő algoritmust hasonlitjuk össze az optimálissal átlagos esetben.

1. §. Konkrét rendszerek

Az INF nevü könyvtári információvisszakereső rendszer a Magyar Állami Földtani Intézet megbizásából készült 1971-72-ben az ELTE-n. Feladata a MÁFI könyvtárában lévő dokumentumok /könyvek, folyóiratcikkek, jelentések stb./ tartalom szerinti nyilvántartása, indexek készitése és adott tartalmu dokumentumok visszakeresése volt.

A dokumentumok tartalmát egy körülbelül 1000 deszkriptort tartalmazó tezaurusz segitségével jellemeztük. A deszkriptorok közül mintegy 70 hierarchikusan kapcsolatban álló földrajzi deszkriptor volt. A rendszer 10 programból állt, melyek néhány jellemző adatát az 1. táblázat tartalmazza

Program neve,	nyelve,	memóriaigénye szóban,	teljesitménye
INF1	PLAN	390	5 db/perc
INF2	FORTRAN	13600	
INF3	"	8100	
INF4	"	23450	52 db/perc
INF5	"	23250	20 db/perc
INF6	"	8370	
INF7	"	8640	
INF8	"	3840	
INF9	"	13820	
INF10	"	14230	

1. táblázat. Az INF rendszer progjamjainak néhány adata

Az INF2 az uj dokumentumok leirásának szintaktikus ellenőrzését és a hibátlan leirásoknak az állományhoz csatolását végezte. Az INF3 az állomány adott részére vonatkozó szerzői, témaszerinti és földrajzi indexeket készitett. Az INF5 KWIC-indexeket generált. Az INF6 segitségével a deszkriptorok logikai kifejezésével /AND, OR és NO müveletek megengedettek/ jellemezhető dokumentumok visszakeresésére volt lehetőség. Az INF7 a dokumentumok angol cime alapján automatikus tárgyszavazást végzett. Az INF9 a tezaurusz karbantartását, fejlesztését biztositotta. A többi program listázást és más segédfunkciókat látott el.

Az INF a visszakeresést a szekvenciális állományban lineárisan végezte [2].

A következő két rendszerben a dokumentumokat hasonló dokumentumokból álló zónákra osztják, és minden zónához meghatározzák az odatartozó dokumentumok tartalmát jellemző karakterisztikus vektort, és az igényvektornak a karakterisztikus

vektorokkal való összehasonlitása után csak a releváns zónákban való lineáris keresésre van szükség.

A RASZT nevü rendszer egy moszkvai kutatóintézet részére készült 1981-82-ben. Növényekkel kapcsolatos megfigyelési adatok kezelését végzi. A rendszer kb. 90000 megfigyelés adatait tartalmazza, melyek kb. 300 zónára oszlanak. A visszakeresési igények kb. 100 számot tartalmazó vektorok. Ezekben a vektorokban az AND, OR és NO logikai operátorok használhatók.

A BOL nevü rendszer egy moszkvai klinika részére készült a Moszkvai Állami Egyetemen, 1982-83-ban, és betegek ötéves megfigyelésére vonatkozó adatokat tartalmaz. A rendszer célja, hogy a betegség fejlődésének, a különböző gyógyitási módszerek hatásának elemzését elősegitse.

A BOL 10000 megfigyeléssorozat eredményeit kezeli, melyek 100 azonos méretü, a visszakeresési igények figyelembe vételével kialakitott zónában helyezkednek el.

2. §. Matematikai modell

Adott dokumentumok $T=\{t_0, \ldots, t_{n-1}\}$, igények $Q=\{q_0, \ldots, q_{r-1}\}$, deszkriptorok $D=\{d_0, \ldots, d_{m-1}\}$ halmaza, ahol $Q=2^D$. Továbbá, adott a dokumentumok leirásának mátrixa $[t_{ij}]_{n\times m}$, ahol $t_{ij}=1$, ha a d_j deszkriptor jellemzi a t_i dokumentum tartalmát, és $t_{ij}=0$, ha nem /$i=\overline{0,n-1}$, $j=\overline{0,m-1}$/.

Tegyük fel, hogy T a $T_0, \ldots, T_{z-1}$ /$1\le z\le n$/ zónákra van bontva, ahol $\bigcup_{k=0}^{z-1} T_k = T$ és $T_{k_1}\cap T_{k_2}=\emptyset$ /$0\le k_1<k_2\le z-1$/. Ilyen felbontás megadható például az $X=[x_{ik}]_{n\times z}$ tartalmazási mátrix segitségével, ahol $x_{ik}=1$, ha $t_i\in T_k$ és $x_{ik}=0$, ha $t_i\notin T_k$ /$i=\overline{0,n-1}$, $k=\overline{0,z-1}$/.

Adott X felbontás $[u_{kj}]_{z\times m}$ karakterisztikus mátrixát /amely a karakterisztikus vektorokat mint sorvektorokat tartalmazza/ igy definiáljuk: $u_{kj}=0$, ha egyetlen $t_i\in T_k$ dokumentum leirásában sem szerepel d_j, és $u_{kj}=1$, ha T_k-ban legalább egy elem leirása tartalmazza d_j-t. Ekkor

$$u_{kj}=\max_{0\le i\le n-1} t_{ij}x_{ik} \quad /j=\overline{0,m-1},\ k=\overline{0,z-1}/.$$

A visszakeresés ugy történik, hogy a q_ℓ igényt összehasonlitjuk minden zóna karakterisztikus vektorával, majd a releváns zónákban /ahol minden igénybeli deszkriptor megtalálható/ lineáris keresést végzünk.

A paraméterek /zónák száma és mérete, dokumentumok szétosztása a zónák között/ megválasztásának hatékonyságát az visszakeresési idővel jellemezzük, melyet a következőképpen definiálunk. A $q_\ell\in Q$ igényhez tartozó keresési időt X tartalmazási mátrix esetén $C_{\ell X}$ -szel jelöljük, és

$$C_{\ell X}=z+\sum_{k=0}^{z-1} g_k L_{k\ell}+\alpha\sum_{k=0}^{z-1} L_{k\ell},$$

ahol α egy uj zónában való visszakeresés előkészitésének /memóriába töltésének/ ideje, g_k a k-ik zóna mérete, azaz $g_k=\sum_{i=0}^{n-1} x_{ik}$, $L_{k\ell}$ pedig logikai változó, melynek értéke "igen"

/1/, ha $u_k \not\sim q_\ell$, és"nem" /0/ ellenkező esetben, azaz

$$L_{k\ell} = \begin{cases} 1, \text{ ha } u_{k_j} \not\sim q_{\ell_j} \quad / j = \overline{0, m-1} / \\ 0, \text{ ha van olyan } \quad / 0 \le j \le m-1 /, \text{ melyre } u_{k_j} < q_{\ell_j}. \end{cases}$$

Legyen most Q valószinüségi tér A σ-algebrával és $P/A/$ valószinüségi mértékkel. A $C_{\ell X}$ keresési idő várható értékét M_X-szel jelöljük. Ekkor $M_X = \sum_{\ell=0}^{r-1} P(q_\ell) C_{\ell X}$.

Ennek az M_X mennyiségnek, mint hatékonysági mértéknek az alapján fogjuk a paraméterek megválasztására javasolt különböző algoritmusokat összehasonlitani.

Ha a zónák számára és méretére vonatkozó z_{min}, z_{max}, g_{min}, g_{max} korlátok adottak, feltesszük, hogy $0 \le g_{min} \le g_{max}$ és $1 \le z_{min} \le z_{max}$. Ha ilyen korlátok nincsenek, akkor a $g_{min} = 0$, $z_{min} = 1$, $g_{max} = z_{max} = n$ értékeket alkalmazzuk. Ekkor a paraméterek optimális megválasztása a következő egészértékü programozási feladatra vezet.

Adott $n, m, z_{min}, z_{max}, g_{min}, g_{max}$ számok, $[t_{ij}]_{n \times m}$ mátrix és $P/q/$ eloszlásfüggvény esetén határozzuk meg azt az $X^* = [x_{ik}]_{n \times z}$ mátrixot, amelyre

$$x_{ik} \in \{0,1\} \quad / i = 0, n-1, \quad k = 0, z-1 /,$$

$$\sum_{k=0}^{z-1} x_{ik} = 1 \quad / i = 0, n-1 /$$

$$g_{min} \le g_k \le g_{max} \quad / k = 0, z-1 /$$

$$M_{X^*} \le M_X \quad /\text{minden lehetséges } X\text{-re}/.$$

Megjegyezzük, hogy ez a feladat egy [3]-ban kitüzött feladat továbbfejlesztett változata /ott z rögzitett, és az X-ben levő egyesek számát kell minimalizálni/.

3. §. A paraméterek meghatározására szolgáló algoritmusok

Az irodalomban a következő algoritmusokat találtuk.

Salton bebizonyitotta, hogy ha a dokumentumhalmaz rendezett, minden lehetséges dokumentumleirásnak pontosan egy dokumentum felel meg és az igénnyel azonos leirásu /egyetlen/ dokumentumot keressük, akkor M_X minimalizálásának feltétele $z = g_k = \sqrt{n} / k = 0, z-1 /$. Ezen az eredményen alapul A_{SZIM}

Szimmetrikus algoritmus A_{SZIM} [4]

$$z \approx g_0 \approx \ldots \approx g_{z-1} \approx \sqrt{n}.$$

A következő algoritmus adott q_{max} esetén a zónák számát a lehető legkisebbre választja /a zónaméret igy maximális/.

Maximális algoritmus A_{MAX} [3] $\quad z = \lceil n \cdot q_{max} \rceil$.

A [3]-beli optimalizálási feladattal kapcsolatos A_{MIN}.

Minimális algoritmus A_{MIN} [3]. Adott z és q_{max} esetén ugy osztja szét a zónák között a dokumentumokat, hogy az X-ben lévő egyesek száma minimális legyen.

Optimális algoritmus A_{OPT} [5]. Az előző rész végén leirt optimalizálási feladat megoldását szolgáltató algoritmus.

Kvázioptimális algoritmus A_{KV} [5]. A zónák számát ugy határozza meg, mint A_{OPT}, majd a zónák között az X-beli egyesek számát minimalizálva osztja szét a dokumentumokat.

Rendező algoritmus A_{REND} [5]. A zónák számát és méretét a

$$z = \lceil \sqrt{n} \rceil, \quad q_k = \lceil n/z \rceil \quad / k = \overline{0, z-2} /, \quad q_{z-1} = n - (z-1)\lceil n/z \rceil$$

formulák alapján határozza meg, majd rendezi a dokumentumokat a leirások, azaz a $/t_{i0}, \ldots, t_{i,m-1}/$ vektorok alapján, és T_0-ba az első q_0, T_1-be a soron következő q_1 dokumentumot teszi, és igy tovább.

4. §. Az algoritmusok hatékonysága szélsőséges esetekben

Tegyük fel, hogy adott m, n, q_{max}, $P(q)$ és $[t_{ij}]_{n \times m}$ esetén X_{MIN}-nel jelöljük azt a tartalmazási mátrixot, amelyet A_{MAX} és A_{MIN} segitségével, és X_{KV}-vel azt, amelyet A_{KV} segitségével kaptunk. Továbbá tegyük fel, hogy X_{SZIM}-hez ugy jutottunk, hogy a zónák számát A_{OPT}, a zónák méretét A_{SZIM}, majd a dokumentumok zónákra osztását A_{MIN} segitségével végeztük. Ekkor érvényesek a következő állitások.

1. TÉTEL [5]. Mindem valós B-hez megadhatók olyan m, n, $P/q/$, q_{max} és $[t_{ij}]_{m \times n}$ paraméterek, hogy teljesüljön

$$M(X_{MIN})/M(X^*) > B. \quad \boxtimes$$

2. TÉTEL [5]. Minden valós E-hez megadhatók olyan m, n, $P/q/$, q_{max} és $[t_{ij}]_{m \times n}$ paraméterek, hogy teljesüljön $M(X_{MIN})/M(X^*) > E$. $\boxtimes$

3. TÉTEL 5. Minden valós F-hez megadhatók olyan m, n, $P/q/$, q_{max} és $[t_{ij}]_{m \times n}$ paraméterek, hogy teljesüljön $M(X_{MIN})/M(X^*) > F$. $\boxtimes$

Tegyük fel, hogy rendszerünk paraméterei a következő tulajdonságokkal rendelkeznek. $b > 1$ egész szám, $m = 2^b$, $n = 2^m$, $r = 2^m$, $q_{max} = 2^b$ és

$$i = \sum_{j=0}^{m-1} t_{ij} 2^j, \quad l = \sum_{j=0}^{m-1} q_{lj} 2^l \quad / i = \overline{0, n-1}, \; l = \overline{0, r-1} /,$$

azaz a dokumentumleirások és igények mint kettes számrendszerbeli számok megadják a megfelelő leirás ill. igény indexét.

Ekkor a zónák száma illetve mérete a különböző közelitő algoritmusok szerint azonos:

$$z(A_{SZIM}) = z(A_{MAX}) = z(A_{REND}) = \sqrt{n} = 2^{\gamma},$$

$$g_k(A_{SZIM}) = g_k(A_{REND}) = n/z = 2^{\delta} \quad (k = \overline{0, z-1})$$

Ezek a számszerü adatok azt mutatják, hogy bizonyos körülmények között az emlitett közelitő algoritmusok azonos paraméterértékeket adnak.

5. §. A rendező algoritmus vizsgálata átlagos esetben

Tegyük fel, hogy a deszkriptorok száma m, $n = 2^m$, $r = 2^m$, γ adott egész szám $/0 \leq \gamma \leq m/$, $z = 2^{m-\gamma}$, $g_k = 2^{\gamma}$ $/k = \overline{0, z-1}/$. Továbbá az igények egyenletes eloszlásuak, azaz $P(q_l) = 1/2^m$ $/l = \overline{0, r-1}/$, és a dokumentumleirások, igények mint kettes számrendszerbeli számok megadják a megfelelő leirás ill. igény indexét.

Ilyen feltételek mellett érvényes a következő állitás.

4. TÉTEL [6] . Ha az X_{REND} tartalmazási mátrixot az A_{REND} rendező algoritmus állitotta elő, akkor

$$M/X_{REND}/ = (1/2)^{\gamma} 2^m + (4/3)^{\gamma} (3/2)^m + a\,(2/3)^{\gamma} (3/2)^m \quad \boxtimes$$

Említést érdemelnek a tétel következő következményei.

1. Következmény. Ha a 4. tétel feltételei mellett $a = 0$, akkor γ optimális értéke $\gamma_{opt} \approx C_1 \cdot m + C_2$, ahol $C_1 = \frac{\log 4/3}{\log 9/3} \approx$ $\approx 0{,}3$ és $C_2 = [\log[(\log 2)/(\log 4/3)]]/\log 8/3$, ahonnan $z_{opt} \approx 0{,}5 \cdot n^{0{,}7}$ és $(g_k)_{opt} \approx 2 \cdot n^{0{,}3}$ $(k = \overline{0, z-1})$. $\boxtimes$

2. Következmény. Ha a 4. tétel feltételei mellett $m = 2\delta$, $\gamma = \delta$, akkor legyen X_{SQRT} a $z = \sqrt{n} = 2^{\delta}$ zónaszámhoz tartozó tartalmazási mátrix. Ekkor

$$M/X_{SQRT}/ = 2^{\gamma} + 3^{\gamma} + a(3/2)^{\gamma} \text{ és } \lim_{\gamma \to \infty} M(X^*)/M(X_{SQRT}) = 0 \quad \boxtimes$$

Az A_{REN} által előállitott X_{REN} -re vonatkoznak a következő állitások is.

5. TÉTEL [7]. Ha a zónák száma $z = 2^w$ alakban rögzitett, és $w \leq 2$, akkor $M(X_{REND}) = M(X^*)$. $\boxtimes$

6. TÉTEL 7 . Ha a zónák száma $z = 2^w$ alakban rögzitett, és $w \geq C_1 \cdot m$, akkor $M/X_{REND}/ \leq 2M/X^*/$, ahol C_1 az 1. következményben szereplő állandó. $\boxtimes$

6. §. Összefoglalás

A kétszintes zónás deszkriptoros információvisszakereső rendszerek Saltontól származó matematikai modellje alapján vizsgáltuk a paraméterek meghatározására szolgáló algoritmusokat.

Mivel az optimalizálási feladat nagyon müveletigényes egészértékü programozási feladatra vezet, vizsgáltuk a különböző közelitő algoritmusoknak az átlagos válaszidőre gyakorolt hatását.

Megállapitottuk, hogy bizonyos ideális feltételek mellett a közelitő algoritmusok azonos eredményre vezetnek, mig szélsőségesen kedvezőtlen esetben a közelitő algoritmusok az optimálisnál sokkal rosszabb eredményt adnak.

Az utolsó két tétel azt mutatja, hogy bizonyos feltételek mellett a rendező algoritmus optimális, vagy ahhoz közeli visszakeresési időt eredményez.

Az elméleti vizsgálatokat gyakorlati tapasztalatok is alátámasztják. Korábban széles körben elterjedt a $z=\sqrt{n}$ zónaszám alkalmazása. A RASZT és a BOL rendszereken végzett kisérletek azt mutatták, hogy az 1. következménynek megfelelő $z \approx z_{opt}$ zónaszám alkalmazásával körülbelül egy nagyságrenddel csökkent az átlagos visszakeresési idő.

Irodalomjegyzék

1 Сэлтон Г., Автоматическая обработка, хранение и поиск информации. Москва, Советское радио, 1973.

2 Iványi A., Nyirádi L., Library information retrieval systems INF and FARMDOC, In: Conference on systems for information servicing of professionally linked computer users /Varna, May 23-29, 1977/, Bulgarian Academy of Science, p. 307-313.

3 Белобродский А. В., Реметников В. Н., Об одном способе поиска релевантных векторов, В кн.: Вопросы оптимизации и управления, Москва, Изд. МГУ, 1979, с. 59-63.

5 Ивани А. М., Сотников А. Н., Оценка эффективности некоторых поисковых алгоритмов в АИПС, Программирование, 1983 /в печати/.

4 Реметников В. Н., Сотников А. Н., Алгебраические свойства Σ-структуры и псевдорелевантные векторы., Вестник Московского Университета, Сер. Вычисл. мат. и кибернетики, 1982, No. 1, с. 70-75.

6 Ивани А. М., Сотников А. Н., Об оптимизации дескрипторных зонно-иерархических информационно-поисковых систем, Вестник Московского Университета, Сер. Вычисл. мат. и кибернетики, 1983 /в печати/

7 Ивани А. М., Лудерер Б., Об одной задаче целочисленного программирования, связанной с информационным поиском, Изв. АН СССР, Серия Техническая кибернетика, 1983 /в печати/.

Majtényi Edit

KSH Számitóközpont

Adatbázisadminisztráció a statisztikai információrendszerben

A statisztikai szervezetek fő feladata a begyüjtött adatok feldolgozása és információk szolgáltatása a lehető legrövidebb idő alatt.

A hetvenes évek elején alakult ki Hivatalunkban az a nézet, hogy a váratlan - ad hoc - információ igények kielégitésére a legjobb eszköz az adatbáziskezelő rendszer alkalmazása.

A statisztikai információ rendszerre a statisztikai mutatók és azok előfordulásának igen nagy száma, valamint a lehetséges kapcsolatok sokfélesége jellemző. Az információ igények állandóan változnak, szinte minden mindennel kapcsolatban lehet. Az adatbázis kezelésére az INFORMATICS cég által forgalmazott MARK IV rendszer alkalmazása látszott célszerünek, mivel a MARK IV az adatstrukturák rugalmas kezelését teszi lehetővé.

1. Az adatbázisadminisztráció kialakulása

A hetvenes évek elején az adatbázis fejlesztése, karbantartása és az adatbázisból való lekérdezés egyetlen, körülbelül 15 fős szervezet keretén belül történt. Az igények növekedése mind az adatbázisban tárolt témák számának, mind az ad hoc lekérdezések számának növekedésében jelentkezett; a régi szervezet nem volt alkalmas a feladatok ellátására.

Hivatalunkban az adatbázis környezetében müködő szervek a szakirodalomban leirtaktól némileg eltérően alakultak ki. Egy-egy statisztikai adatgyüjtés adatbázisba vitele 1-2 emberévnyi előkészitő tevékenységet igényel a témáért felelős statisztikai főosztály /továbbiakban adatbázisgazda/ munkatársainak bevonásával. A gyakorlatban hasznosnak bizonyult ezt a fázist az adatbázisadminisztrációtól elválasztani, oly módon, hogy az ujonnan bekerülő témák adatbázisba szervezését az Adatbázisfejlesztési osztály végzi.

Miután tisztázódott az adatbázis tartalma és a bekerülő adatok rendelkezésre állnak, az Adatbázisfejlesztési osztály elkésziti a Rendszertervet. A Rendszerterv tartalmazza az input adatok leirását, az adatbázis tartalmát, és tervezett fizikai szerkezetét. A kész Rendszertervet az Adatbázisadminisztráció megkapja véleményezésre. Az Adatbázisadminisztráció változtatási javaslatai általában az adatbázis fizikai szerkezetét érintik, hiszen az ezzel kapcsolatos tapasztalatok itt csapódnak le.

Általános igény, hogy az adatbázisba régebbi időszakok adatai is bekerüljenek. Ezekkel az adatokkal gyakran jelentkeznek kon-

zisztencia problémák. Előfordulhat az is, hogy az adatbázis gazda a betöltés után megváltoztatja igényeit - az ezt követő módositásokat szintén az Adatbázisfejlesztés hajtja végre. Az első betöltéstől számitott 1/2-1 évig az adatbázis kisérleti üzemmódban müködik az Adatbázisfejlesztési osztály felügyelete alatt. Ezen időszakban az adatbázist csak az adatbázis gazda használhatja. Ekkor készül el az információrendszer érintett részének tartalmi leirása, vagyis a témára vonatkozó meta-adatbázis-rész.

Az Adatbázisadminisztráció a kezdeti problémák kiderülése és kiküszöbölése után veszi át az adatbázist rendeltetésszerü használatra.

Az adatbázis jelenleg csak batch üzemmódban érhető el, s a statisztikusok nem látnak el számitástechnikai feladatokat - igy az adatbázisra vonatkozó ad hoc információ igények kielégitését az Adatbázis-lekérdezési osztály végzi.

A jelenlegi szervezetben tehát Számitóközpontunkban az ugynevezett adatbázisszolgálat három osztályból áll:
- Adatbázisfejlesztés,
- Adatbázisadminisztráció,
- Adatbázis-lekérdezés.

2. A Statisztikai Adatbázis Rendszer /STAR/ felépitése

A STAR két részből áll, a tárgy-adatbázisból és a meta-adatbázisból. A tárgy-adatbázis a tárgy-információ rendszerről tartalmaz adatokat, a meta-adatbázis pedig a meta-információ rendszerről.

21. A tárgy-adatbázis

Adatbázis rendszerünk 10 év óta müködik, néhány témában 13 éves idősoraink vannak. Az adatbázis főleg a gazdaságstatisztika területéről tartalmaz adatokat, a társadalomstatisztikai adatok hányada alacsony.

Az adatokat két szinten - elemi /mikro/ és
- aggregált /makro/
szinten tároljuk. Az aggregálás célja a gyorsabb lekérdezési lehetőségek biztositása.

Az adatbázis utolsó 6 évének adatai mágneslemezen a régebbi évek adatai a mágnesszalagos archivumban találhatók meg. /A mágneslemezen levő adatok mintegy 600Mbyte területet foglalnak el./ Az éves adatgyüjtések idősorainak hossza korlátlan. Az évközieket 3 évig tároljuk. A STAR-ban kb. 2400 statisztikai mutató van adatszolgáltatónként. Az adatszolgáltatók száma 1600-tól 8500-ig terjed. Az egyes mutatók előfordulása egy adatszolgáltatónál 1-től 5-600 lehet, mindezt még a témában rendelkezésre álló idősor hosszával is meg kell szorozni, hogy a STAR-ban található mutató előfordulások számáról képet kapjunk.

Az adatbázisok betöltése, karbantartása és lekérdezése a MARK IV segitségével történik. A MARK IV segitségével a többnyire hierarchikus adatstrukturák kezelése igen rugalmasan oldható meg - a file-ok /adatbázis témák vagy azok részei/ automatikus koordinálásával - a lekérdezések során.

A STAR-ban nincsenek pointerek, mivel a felhasználó minden elképzelhető aspektusban kérhet adatokat - minden lehetséges kapcsolatot pedig szinte lehetetlen, jobban mondva tul költséges lenne ábrázolni.

A file-ok koordinálása, azaz a kapcsolatok lekérdező futás során történő létrehozása a minden adatbázis file-ban megtalálható adatbázis-kulcs segitségével történik.

22. A meta-adatbázis

A statisztikai információ rendszerben mind az információ források mind az információ felhasználók száma igen nagy. A felhasználók nem minden esetben az információ gazdái - igy az "információra vonatkozó információ", vagyis a meta-információ nélkülözhetetlen.

Esetünkben a tárgy-információ rendszert leiró adatok, vagyis a meta-adatok
- egységes rendszert alkotnak,
- számitógéppel kezeltek,
- naprakészek,
- a felhasználó számára elérhetők, vagyis

meta-adatbázist képeznek.

A meta-adatbázis háromféle leirást tartalmaz:
- az információk szöveges leirása statisztikus felhasználók számára,
- adatszótár a software és a programozók számára,
- fizikai file-ok leirása a programozók számára.

221. A felhasználókat három szöveges leirás,
- a Tájékoztató Katalógus,
- a Mutató Katalógus és
- a Nómenklatura Katalógus segiti az információ kérésben.

A Tájékoztató Katalógus KWOC* indexes, minden vezérszóhoz hozzá van rendelve az összes érintett mutató és nómenklatura.

A Mutató Katalógus időSor szemléletü, a mutató azonositójával együtt megadja az adatbázisban tárolt mutatók szöveges leirását, az időSor kezdetét /és végét, ha lezárult/, a mutató mértékegységét, vonatkozási idejét, periodicitását, valamint azon nómenklaturák azonositóit, amelyek szerint a mutatót megfigyelik /pl. a létszámot gazdálkodó

*KEY WORD ON CONTEXT

szervezetenként a foglalkoztatottak korcsoportjai szerint./

A Nómenklatura Katalógus a Mutatókatalógusban hivatkozott nómenklaturák elemeinek szöveges felsorolását, valamint a nómenklaturák összefüggéseit tartalmazza.

A katalógusok alapján a statisztikus felhasználó kiválaszthatja az együtt feldolgozható mutatókat és ezekből az ugynevezett STAR Adatkérő Lapon tetszésszerinti táblákat kérhet az Adatbázislekérdezési osztálytól.

222. Az adatszótár a mutatókat és a kapcsolatokat tartalmazó file-ok leirását, valamint a nómenklaturák szövegeit és a karbantartáshoz szükséges tranzakciók definicióit tartalmazza. Az adatszótár hidat képez a statisztikusoktól a programozókon keresztül az adatbázishoz egy igen szigoru névkonvenció segitségével. /A katalógusban szereplő mutató vagy nómenklatura azonositó megegyezik az adatszótárbeli adatnévvel, és ebből egy egyszerü algoritmussal képezhető a mutatót tartalmazó fizikai file azonositója is./

223. A fizikai file-ok leirását a Fizikai File-ok Katalógusa tartalmazza. Ez a következőkből áll:
- file azonositója,
- egység tipusa,
- egység sorszáma,
- adatszervezés,
- elfoglalt lemezterület,
- rekord forma,
- rekord hossz,
- blokkméret,
- archiválás megtörtént-e,
- a biztonsági másolatok szalagszáma,
- jelszóval védett-e.

3. Az adatbázisadminisztrátor feladatai

Az adatbázisadminisztrátor /jelenleg 6 fő/ alapvető feladata a tárgy-, és a meta-adatbázis karbantartása.

31. A tárgy-adatbázis karbantartása

A tárgy-adatbázis karbantartása főként az adatbázis uj időszakok adataival való kiegészitését jelenti. Ennek előkészületei már jóval az adatbázis betöltése előtt megkezdődnek, amikor az adatbázisgazda bejelenti az adatgyüjtésben történt változásokat. A változás jelenthet uj mutatókat, megszünt mutatókat, vagy előfordulhat, hogy egy mutató tartalmi szempontból nem hasonlitható össze az előző évi mutatóval. Ezekben az esetekben az adatbázisadminisztrátor a régi mutatót lezárja, az ujnak pedig uj mutatóazonositót ad.
Ezután az alábbiakat kell megváltoztatni:
- az adatbázis-rész felépitése /esetleges/,
- a file definiciók,
- a tranzakció definiciók,
- betöltő programok,

- szöveg file-ok /esetleges/,
- Mutató Katalógus,
- Nómenklatura Katalógus.

/A Tájékoztató Katalógus ritkábban kerül karbantartásra./

Az adatbázisba ellenőrzött, javitott adatok kerülnek, ezért utólagos adatjavitásra ritkán van szükség. Gyakori problémát okoz viszont az egyes adatbázis témák között jelentkező inkonzisztencia, mely abból ered, hogy a különböző forrásokból származó, különböző adatgyüjtések csak külön-külön vannak ellenőrizve és eltérés jelentkezik az egyes adatgyüjtések adatszolgáltatói körében vagy az adatszolgáltatók azonositásában. A STAR központi része az adatszolgáltatói regiszter, melynek segitségével való ellenőrzés a konzisztencia problémákat hivatott kiszürni. A problémák kiküszöbölése, az egyes adatbázis témák gazdáival való egyeztetés szintén az adatbázisadminisztrátor feladata.

A gazdaságstatisztikában a megfigyelések nagy része szakágazat orientált, ami azt jelenti, hogy ha egy gazdálkodó szervezet szakágazata megváltozik, akkor az egyes évek szakágazati adatai többé nem hasonlithatók össze. A probléma megoldása a szervezeti változások kiszürése, az eredeti adatbázis homogenizálása.Az adatbázis két formában áll a felhasználók rendelkezésére: eredeti és következő év szerkezetére homogenizált. Igy láncindexek képezhetők minden év eredeti adatainak a megelőző év homogenizált adataival való összevetésével.

A homogenizálás végrehajtása mind az adatbázisadminisztrátor mind az adatbázisgazda számára igen sok nehézséggel jár. Az adatbázisgazdának pótlólagos adatokat kell begyüjtenie a szétvált szervezetekből, ami gyakran okoz problémát; az adatbázisadminisztrátor számára pedig a homogenizálás végigvezetése jóval időigényesebb és bonyolultabb feladat, mint az eredeti adatbázis betöltése.

Másfajta homogenizálási módszer szolgál a termék és település orientált adatbázisrész homogenizálására - itt csak összevonásokra van szükség, ezért a teljes idősor homogenizálható /pótadatok nélkül/ az utolsó év szerkezetére.

Az adatbázis helyes müködését az adatbázis karbantartása és védelme biztositja.

Az adatbázis jelszóval védett, módositásra csak az adatbázisadminisztráció jogosult, a lekérdező programozók csak olvasási jogosultsággal rendelkeznek,Jogosulatlan személyek nem érhetik el az adatbázist.

Az adatbázis biztonságát a mágnesszalagos archivumba elhelyezett másolatok szavatolják.

32. A meta-adatbázis karbantartása

Az adatbázisadminisztrátor legfontosabb segédeszköze a meta-adatbázis.

A metaadatok változásainak igen rövid idő alatt rendelkezésre kell állnia, hisz a különböző szintü felhasználók - Hivatalon kivüliek, hivatalbeli statisztikusok és programozók - ennek alapján tudják megfogalmazni adatkéréseiket, illetve programjaikat. Vagyis
- az adatszótár listák,
- a Fizikai File-ok Katalógusa,
- a Mutató Katalógus és
- a Nómenklatura Katalógus időben való szétküldése nélkülözhetetlen.

- . -

Összefoglalva: az adatbázisadminisztátor a Hivatal adatainak jó őrzője és gondozója kell legyen. Az adatbázis megfelelő üzemeltetése az adatbázis helyes használatának alapvető feltétele.

A meta-adatbázis pontossága és könnyü használhatósága jó elemzési lehetőségeket biztosit.

ADATVÉDELEM—FELHASZNÁLÓ-AZONOSITÁS

Márkus Gábor

MTA SzTAKI

1111 Budapest, Kende u. 13-17.

BEVEZETÉS

A time-sharing számitógéprendszerek alkalmazásakor rendkivül fontos, hogy a rendszert megóvjuk az illetéktelen felhasználóktól; ezért szükséges a felhasználók autentikusságának vizsgálata (azonositása). Természetesen a bevitt adatok, programok ellenőrzésére is szükség van — adat-azonositás —, ennek vizsgálata azonban tulnő ezen előadás keretein. A védelem szükségességét alátámasztják a számitástechnikailag fejlett országok számitógépes bünözéssel foglalkozó statisztikái (az anyagban előforduló adatok, eredmények referenciáit illetően ld. Márkus G.: *Megbizható felhasználó-azonositó eljárások* MTA SzTAKI Working Paper AP/21 Budapest, 1983).

MÓDSZEREK A FELHASZNÁLÓK AZONOSITÁSÁRA

Egyik legelterjedtebb módszer a *password*-ökkel való azonositás. Minden felhasználó, amikor először kapcsolódik be a rendszerbe, választ egy jelsorozatot (password) és ezt a felhasználó nevével együtt elhelyezeik a rendszerben egy táblázatban, a *password directory*-ban. Minden alkalommal, amikor a felhasználó be akar kapcsolódni, a rendszer megkérdezi tőle a password-öt, és akkor és csak akkor engedi őt bekapcsolódni, ha ismeri a nevéhez tartozó password-öt.

A password directory-hoz csak az azonositást végző program férhet hozzá, a felhasználók nem. Természetesen van egy személy, a *rendszeradminisztrátor*, aki jogosult e file tartalmát nemcsak olvasni, hanem megváltoztatni is.

E rendszerben a rendszeradminisztrátor ismeri a password-öket, bár semmi sem indokolja még azt sem, hogy módja legyen ismerni őket. Egy illetéktelen személy a következő módokon férhet hozzá egy felhasználó password-jéhez (feltéve, hogy ezt az információt a rendszeradminisztrátor nem árulja el):

(i) a felhasználó gondatlanságából (pl. továbbadja a password-öt, vagy könnyen kitalálható password-öt választ),

(ii) hozzájut a rendszerben tárolt információkhoz (pl. elolvassa, kinyomtatja a password-file-t),

(iii) lehallgatja a felhasználó terminálját a rendszerrel összekötő vonalat.

Minthogy a rendszer nem tud különbséget tenni két, ugyanazzal a password-del jelentkező személy közt, az (i) lehetőség semmiféle password-rendszerrel sem védhető ki. Más, a felhasználó biológiai jellemzőit (hang, ujjlenyomat,stb.) felhasználó rendszerekkel ez a probléma is kiküszöbölhető.

A továbbiakban foglalkozzunk először (ii)-vel. Ez volt az a gyengesége a hagyományos password-rendszereknek, amely arra a képtelennek tűnő követelményre vezetett, hogy a rendszerben ne tároljuk a password-öt, de mégis biztonsággal el tudjuk dönteni, hogy a felhasználó valóban a password-jével kísérelt-e meg bekapcsolódni. De ha a felhasználók neve mellett nem a password-jüket tároljuk, akkor mit? Először Needham javasolta, hogy a PW password helyett annak egy alkalmas transzformáltját, $f(PW)$-t tároljuk. (Itt megjegyezzük, hogy arra is lehetőség nyílik, hogy – megfelelő f transzformáció esetén – a rendszerben ne tároljuk a **felhasználók nevét.**) Az f függvényt ugy kell megválasztani, hogy $f(PW)$ ismerete esetén se lehessen PW-t meghatározni. Az ilyen tulajdonságu függvényeket nevezik egyirányu függvényeknek. Pontosabban: egy f függvényt *egyirányu függvény*nek nevezünk, ha minden, az f értelmezési tartományába eső x argumentumra *könnyen* ki tudjuk számitani az $f(x)$ értéket, de majdnem minden, az f értékkészletébe tartozó y-ra *számitástechnikailag megoldhatatlan* az $y=f(x)$ egyenlet x ismeretlenének meghatározása.

Vegyük észre, hogy a fenti definicióban megkövetelt nem-invertálhatóság alapvetően különbözik a matematikában általában használt fogalomtól, hiszen itt nem arról van szó, hogy az inverz kép ne legyen egyértelmü, hanem arról, hogy egy inverz képet gyakorlatilag ne lehessen meghatározni. A két fogalom közötti különbséget jól mutatja egy konsàns függvény, amely a szokott értelemben természetesen (ha $|D(f)| > 1$) nem invertálható, számitástechnikai szempontból pedig a legegyszerübben invertálható függvények egyike (a keresett x argumentum az értelmezési tatomány bármely pontja lehet).

Figyelmet érdemel a definicióban a következő három meghatározás is: *könnyen*, *majdnem minden* és *számítástechnikailag megoldhatatlan*. Mindhárom fogalmat definiálhatjuk precizen, a konvencionális módon. Pl. a *könnyü* és *megoldhatatlan* esetében a fogalmakhoz — definicióként — számitás- és memória-igény korlátokat rendelhetünk; a *majdnem minden* definiálásához pedig definiálhatunk az f értékkészletén σ-gyürüt vagy σ-algebrát és mértékeket, és igy e fogalom is precizen definiálható lenne. Azonban nem célszerü ez a konzervativ eljárás, mert esetleg jól használható egyirányu függvényeket zárna ki. *Könnyü* számitáson értsük azt, amit ma könnyen végre tudunk hajtani (ez később sem lesz nehezebb); a *majdnem minden* esetében maradjunk meg a szemléletes, köznapi jelentésnél; *megoldhatatlan*nak pedig tartsuk azt a számitást, amely megalapozott becslések szerint az elkövetkező néhány évtizedben megoldhatatlan lesz (ilyen számitás- és memória-korlátok: 10^{30} müvelet és 2^{100} bit memória). Ez utóbbi definicióhoz kapcsolódóan két dolgot kell megemliteni. Először: a *megoldhatatlan* definiálható a szokott módon, "örökérvényüen" is, ekkor a korlátok lehetnek pl. 10^{70} müvelet és 10^{60} bit memória (vannak olyan függvények, amelyek még e konzervativ definició értelmében is egyirányuak). Másodszor: előfordulhat, hogy egy addig egyirányunak tartott függvény inverzének kiszámitására egyszerü módot fedeznek fel. Ekkor a függvény természetesen már nem lesz többé egyirányu függvény, és igy minden e függvény egyirányuságára épülő rendszer egyik napról a másikra használhatatlanná válik.

Emlitésre érdemes az is, hogy még ha a függvény inverzének meghatározására nem is áll rendelkezésre direkt eljárás, a teljes halmazt kimeritő próbálgatást (teljes leszámlálásnak is nevezhetjük) minden függvény esetében megkisérelhetjük. Itt van szerepe a G.B. Purdy által definiált fogalomnak, a függvény degenerációjának:

$$d := \max_{Y_j \in R(f)} |\{X_i \in D(f) \; : \; f(X_i)=Y_j\}|$$

(feltételezzük, hogy létezik maximum). Purdy igazolta, hogy minél nagyobb egy függvény degenerációja, annál nagyobb valószinüséggel vezethet eredményre a teljes leszámlálás.

Egyirányu függvény lehet pl. egy bitmanipulációk sorozatával előállitott függvény (Evans, Kantorowitz, Weiss); egy prim-modulusu nagyfokszámu polinom-kongruencia (Purdy):

$$f(x) \equiv x^n + a_1 x^{n-1} + \cdots + a_n \pmod{P};$$

GF(q)-beli primitiv elem hatványozása (Pohling–Hellman); egységelemes gyürü feletti mátrix-hatványozás (ez az előbbi általánositása – Márkus).

Ez utóbbi esetben a felhasználó password-je az $n \times n$-es mátrixból és a kitevőből álló rendezett pár: (A,e); feltétel: az A mátrix különböző hatványainak száma nagy legyen és előforduljon hatványai között a megfelelő egységmátrix. Alkalmas mátrixokat viszonylag könnyen találhatunk. Legyen most a gyürü pl. $R = GF(2^{521})$. Ekkor a hatványozás *1042* mátrix-szorzást igényel R felett, mig az inverz (a gyürü feletti mátrix alapu logaritmus) kiszámitásához 2.6×10^{78} -szor ennyi számitás szükséges. 10^6 müvelet/sec sebességet feltételezve a logaritmus meghatározásához $n=1$ esetén több, mint 8.66×10^{66} év, $n=20$ esetén pedig több, mint 3.58×10^{70} év lenne szükséges (a hatványozás még ez utóbbi esetben is csak mitegy *0.5* másodpercet igényelne). E választással a mátrix-hatványozás a konvencionális definició szerint is egyirányu függvény.

A felsorolt egyirányu függvények felhasználásával konstruálható olyan felhasználó-azonositó eljárás, amelyben a password egy számpár. E rendszer azt is lehetővé teszi, hogy ne tároljuk a felhasználók neveit.

A password-rendszerek (iii) gyengesége akkor jelentkezik, amikor a központi rendszer és a felhasználói terminál távol van egymástól. Ekkor az őket összekötő vonal (ez legtöbbször telefonvonal) fizikai védelme nem gazdaságos, igy lehetőség van a továbbitott információ lehallgatására, meghamisitására, a továbbitás megszakitására. (Itt van különös jelentősége az adatazonositásnak.) Az ilyen password-rendszerekben password-ök sorozatát szokás alkalmazni. A sorozat megválasztására két mód kinálkozik: (1) mindjárt az elején meghatározzuk az összes PW_i-t, és $f(PW_i)$-t a rendszer mindvégig tárolja – ekkor a felhasználónak is, a rendszernek is sok információt kell tárolnia; (2) a felhasználó az i-edik bekapcsolódáskor küldi el a következő bekapcsolódásnál azonositóul szolgáló $f(PW_{i+1})$-et – ekkor lehetséges, hogy (kommunikációs hiba vagy illetéktelen beavatkozás miatt) $f(PW_{i+1})$ helyes értéke nem jut el a rendszerbe. L.Lamport olyan rendszert dolgozott ki, amely mentes e hátrányoktól.

Legyen f egy olyan egyirányu függvény, amelynek N-szeri szukszcessziv alkalmazása is könnyü, és legyen PW egy password. Ekkor a password-sorozat i-edik eleme:

$$PW_i = f^{N-i}(PW) \quad , \quad (1 \leq i \leq N)$$

(ahol f^k az f függvény k-szori szukszcessziv alkalmazását jelöli — feltételezzük, hogy ez értelmes). A sorozat elemei mind különbözőek (igy nem lehet egy esetleg lehallgatott password-del később a rendszerbe kapcsolódni), és $i>1$ esetén éppen a password szükséges a következő password azonositásához. Sőt PW_i értéke alkalmas bármely további PW_j , $j>i$ password azonositására is, miáltal lehetővé válik, hogy akkor se kelljen egyetlen password-öt se ismételten használni, ha valamilyen okból a rendszert ujra kell inditani; tehát még ilyenkor sem tud egy illetéktelen személy felhasználni egyetlen lehallgatott password-öt sem.

Az emlitett egyirányu függvények közül csak az utolsó nem használható itt közvetlenül, de alkalmas módoditással az is megfelelővé tehető (ha pl. R elemei megfeletethetők egész számoknak (a megfeleltetés legyen ψ), akkor $PW_i=A^e$ esetén lehet $PW_{i+1}=A^r$, ahol $r = e \cdot g(A^e)$ egy alkalmas g függvénnyel — g lehet pl. $\psi(A^e)$ legnagyobb eleme abszolut értékének vagy $\psi(\det|A^e|)$-nek egy polinom-kongruenciája, stb.).

ZÁRÓ MEGJEGYZÉSEK

Az eddig kidolgozott eljárások megbizhatósága a hosszu és teljesen értelmetlen password-ökön mulott (általában annál megbizhatóbbak, minél hosszabb a password). Egy felhasználó azonban a könnyen megjegyezhető (rövid és értelmes) password-öket szereti: a hosszut és értelmetlent elfelejti, ha pedig leirja vagy rögziti valamilyen (egy vagy több) adathordozón, akkor azt elveszitheti, lemásolhatják, ellophatják. Ugy tünik, ez az ellentmondás egyelőre nem oldaható fel. Részleges megoldására szolgálhat a fent leirt mátrix-hatványozásra épülő felhasználó-azonositó eljárás, mert ott — igaz hosszabb, de — értelmes szöveg is lehet password a megfelelő szintü biztonság megőrzése mellett.

A megvalósitást illetően az *inteligens terminál* tünik a legigéretesebbnek. Ez magában foglal egy mágneskártya/kazetta olvasó egységet és egy mikroszámitógépet, és igy elvégzi a felhasználó (adat) azonsoitását, pontosabban annak egy részét. Igy egy rendszerbe különböző felhasználók különböző módokon kapcsolódhatnak be, egy felhasználó pedig akár több rendszerbe is ugyanugy. Ekkor nem jelent nagy problémát a távoli terminál sem.

Bármiféle implementációra is fennáll azonban, hogy minél megbizhatóbb egy rendszer, annál költségesebb is. Ezért mindig az igényeket és lehetőségeket figyelembe véve, a veszélyeket mérlegelve egy kompromisszumos optimumot kell megvalósitani.

Dr Tankó József
Államigazgatási Számítógépes Szolgálat

Kiegyensúlyozott adatbázis alkalmazások

1./ Bevezetés

Az ÁSZSZ számítógépein részben ÁSZSZ fejlesztéssel is kialakított államigazgatási adatbázis alkalmazási rendszerek tapasztalatai fontos jelenségekre hívják fel a figyelmet. Ezek a rendszerek kiegyensúlyozottságával függnek össze. Kiegyensúlyozottságon azt értjük, hogy a rendszer különféle vonatkozásokban mennyire azonos, vagy eltérő szinten kerül megtervezésre és kivitelezésre. A kiegyensúlyozatlanság mindig kihasználatlansággal és gazdaságtalansággal jár.

A következőkben néhány fontos vonatkozást, tényezőt sorolunk fel, és pár konkrét adatbázisnál rámutatunk, hogy a kiegyensúlyozatlanság milyen törekvések következménye lehet és az mennyire a hatékonyság és gazdaságosság rovására megy.

Az elmondottak triviálisnak tűnhetnek, mégis számos konkrét példa bizonyítja, hogy komoly törekvések formájában is élnek olyan elképzelések - elsősorban a döntéshozók körében, akik csupán a rendszer egyes vonatkozásait ismerik kellően - melyek kiegyensúlyozott rendszer létrehozását megakadályozzák.

2./ Alapvető tényezők

A számítógépesítésnél az alkalmazási rendszer valódi színvonalát és eredményességét egy sor tényező határozza meg. Ezek közül soroljuk fel - véleményünk szerint - a legfontosabbakat. Ezek:

1. Hardver eszközök mennyisége és minősége.
   Ebbe beleértjük a háttértárak minőségét, mennyiségét, egyidejüségét, megbizhatóságát is.
2. Szoftver eszközök szinvonala és erőforrásigénye.
   Meghatározó a szoftver programozásának, üzemeltetésének, a felhasználásának komfortja és ára. Adatbáziskezelő nyelv, lekérdező nyelv léte, ill. alkalmazhatósága, stb.
3. Elérési eszközök léte és szinvonala.
   Ebbe értjük a diszpécserezést, a hálózat létét, a vonalak és végberedezések megbizhatóságát. A postai szolgáltatást úgyszintén.
4. A szolgáltatás szervezettségének foka.
   Ez a potenciális lehetőségek hasznosithatóságának szinvonalát jelenti, szakértelmet, gyorsaságot, kényelmet.

5. Alkalmazásfejlesztési felkészültség és kapacitás.
Ez a fejlesztők ismereteit, tapasztaltságát, felhasználó-orientáltságát és hatékonyságát is magába foglalja.

6. Felhasználói igény valódisága, eltökéltsége.
Az igény divatból, objektiv kényszerből származik-e, a felismerés tudatossága, érdekek egyensúlya ide értendő.

7. Felhasználói felkészültség és fogadó készség.
Itt a ténylegesen felhasználó és a felhasználást biztositó, pl. adatszolgáltató, személyek képzettsége, fegyelme és hozzáállása mutatkozik meg.

8. Megvalósitási költségszint és fedezet.
Itt beleértjük a helyes költségbecslést, a fedezet kellő időben való rendelkezésre állását a fejlesztéshez, valamint a költség viszonyát az elért felhasználói hozamhoz. Itt kell számításba venni a célberuházásokat is, beleértve a hardver és szoftver eszközöket.

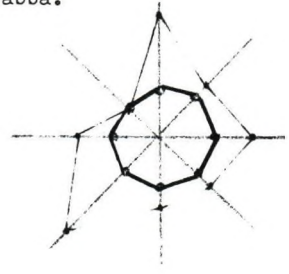
9. Üzemeltetési költség és fedezet.
Hosszútávon biztositható üzemeltetési költségszintet kell betartani és azt összevetni alternativ megoldásokéval. Itt hardver és szoftver bérleti költség is felléphet.

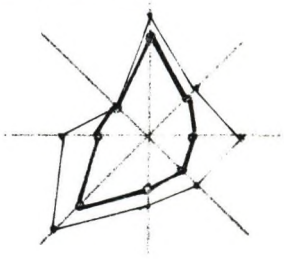
10. Üzemeltetési készség és felkészültség.
Államigazgatásban a számitógép üzemeltetés, a rendszerfejlesztés és a végfelhasználás független intézmények keretében valósul meg. Közöttük az alkalmazási rendszer egyedi kapcsolatot, koordinációt igényel, mind a létrehozás, mind a tartós üzemeltetés keretében. Főleg az üzemeltetés hosszútávú közös tevékenységének szinvonala lehet döntő az alkalmazási rendszer "beválásában".

11. Az alkalmazás célkitüzései.
A különféle szempontok, tényezők elérhető szintjei determinálják az elérhető célokat. A "feszitő", de még elérhető, az igényes, de hosszútávon gazdaságos célkitüzések, illetve az azoktól elmaradó, vagy azokat meghaladó célkitüzések fokozatai lépnek itt fel.
Az egeret szülő vajúdó hegyek, az asztalfiókba temetett csodarendszerek, a csodaváró kiábrándulások a tipikus jelzők a rossz és az egyre inkább beváló, a változó igényekhez rugalmasan alkalmazkodó a tipikus jelzők a jó rendszerekre.

A felsorolt tényezők bármeylike többféle szinten tervezhető és valósulhat meg. A mérés nehézségeitől eltekintve képzelhetjük a tényezőket egy sugársor sugarain ábrázolva egy pókhálószerü alakzattal, amelynek határát a maximális lehetőségek tüzik ki. Koncentrikus körök mentén az azonos tényezőszintek helyezkednek el. Egy konkrét megvalósitott alkalmazási rendszer annál kiegyensúlyozottabb,

minél közelebb áll egy koncentrikus körhöz az a vonal, amelyet az egyes tényezők kihasznált szintjei által kijelölt pontok összekötése eredményez a sugársornál. A rendszer annál magasabb szintü, minél nagyobb a kör sugara. Kiegyensúlyozatlan rendszer tényleges szinvonalát azonban a tényezők közül a minimális szintü határozza meg. Ugyanakkor általában a rendszer költségeit a legmagasabb tényezőszint szabja meg. Az optimális rendszer jellemzője az, hogy az alacsonyszintü tényezőknél a fokozatos továbbfejlesztésével válik egyre kiegyensúlyozottabbá.





3./ Kiegyensúlyozottságot gátló körülmények

Könnyen tudunk példát mondani arra, hogy egy tényező alacsonyszintü megvalósulása miatt a rendszer elérhető hozama jelentősen lecsökken, ami a relativ költség magas szintjét eredményezi. Csak néhány tipikus problémát említek, amely az, egyébként legnagyszerübb, rendszer szintjét is jelentősen leszállitja, ezáltal a kiegyensúlyozottságot gátolja:

- gyakori számitógép üzemzavar eszköz vagy üzemeltetési gyengeségek miatt,
- üzemeltetési nehézségek az aránytalanul nagy erőforrásigény vagy a programrendszer terjedelmessége miatt,
- a nehézkes, késedelmes, pontatlan adatszolgáltatás,
- a sokszorositás elhúzódása, az információtovábbitás lassúsága,
- a fejlesztés elhúzódása miatti elavulás,
- az információk visszatartása, a rendszer használatbavételének halogatása vagy takaréklángon való használat.

Az előzőkben kiemelt tényezők még maguk is összetettek, alkalmazási területenként eltérően jelentkezhetnek és további tényezők válhatnak kritikussá. A felsorolás célja inkább a mondanivalók illusztrálása volt.

Egy államigazgatási adatbázis létrehozása csak akkor célszerű, ha

- ésszerü fejlesztési ráforditásokkal elfogadható időn belül elkészül,
- ésszerü és rendelkezésre álló erőforrások használatával eléggé megbizhatóan üzemeltethető,
- reális igényt elégit ki, amelyet a felhasználók is felismertek és készek lesznek annak használatára.

Optimális rendszer kialakitásánál még egy fontos szempontot figyelembe kell venni, éspedig azt, hogy az egyes tényezők lehetséges szintjei fokozatosan változnak, növekednek. Itt két hibát követhetünk el:

- csak a mai szinttel számolunk és a növekedést nem fogjuk tudni kihasználni /befagyasztott rendszer/
- előrebecsült jövőbeni szinttel számolunk, amely nem valósul meg, igy rendszerünk legfeljebb "félárbócon" lesz használható és örökre kiegyensúlyozatlan marad.

4./ Néhány konkrét alkalmazás

A különböző tényezők különféle alkalmazási rendszereknél más-más formában jelennek meg. A kiegyensúlyozatlanság gyakran annak a következménye, hogy nem veszik tekintetbe a rendszer sajátos jellegzetességeit és ebből irreális tervek, elvárások fogalmazódnak. Három államigazgatási adatbázisnál szeretném példaként a - véleményem szerint - fő jellegzetességeket felsorolni:

- a népességnyilvántartási rendszernél,
- a jogszabálynyilvántartásnál,
- a nyugdijmegállapitó rendszernél.

A népességnyilvántartó rendszer fő jellegzetessége, hogy

- nagyszámú objektum /személyek/ meghatározott adatait kell tárolni, karbantartani, ami sok száz millió karaktert jelent,
- rendszeres nagytömegü adatváltozás történik területileg szétszórtan keletkező adatokkal,
- a tipikus adatszolgáltatás átfogó, a teljes adatbázis áttekintését igényli.

A reális hazai adottságok mellett ezekből következik, hogy kiegyensúlyozott rendszert célul tüzve

- a teljes adatállomány nem lehet egyszerre on-line
- az adatbázis aktualitási fokát nem elsősorban az adatbázisba való bevitel számitógépközeli gyorsasága dönti el,
- a szolgáltatásokat lehetőleg egy feldolgozási menetben kell teljesiteni /leválogatások egyszerre/,

- egyedi adatok hibavalószinüsége avulás miatt jelentős lehet.

A jogszabálynyilvántartási rendszer fő jellegzetessége, hogy

- szöveges objektumokat /jogszabályi tartalom/ kell tárolni,
- az adatigénylés egyedi, széles igénylőkörrel,
- spontán igények, gyors válaszidő igénnyel.

A kiegyensúlyozott rendszer megvalósításához a következőket is meg kell fontolni az erőforrások függvényében:

- a teljes jogszabály szöveget tároljuk-e, vagy csak tartalmi utalásokat /mit nyerünk a teljes szöveggel/,
- a jogalkalmazó közvetlenül fordul a géphez, vagy közvetve /mire képes, ill. hajlandó/,
- a jogalkalmazónak az információra /válasz/ maradandó formában van-e szüksége /nyomtatva/,
- a jogalkalmazónak valójában milyen hozzáférési mód és válasz-gyorsaság szükséges.

Nyilvánvaló, hogy

- ha nem nélkülözhető a jogszabályok nyomtatott gyüjteménye a jogalkalmazók asztalán, akkor felesleges a teljes szöveg tárolása;
- ha a válasz irásban kell és nyomtató lehetetlen minden jogalkalmazó szobájában, akkor jobb a közvetett géphezfordulás;
- amikor az adatbázis on-line, az intenziv lekérdezés optimális és nem a minél kevesebb számú;
- a terminál "ha maga használja" új technikai szemléletet kiván a jogalkalmazótól, a nyomtató pedig zajos és "kényes" és számitani kell a gyakori üzemzavarra is.

A nyugdijmegállapitó rendszer fő jellegzetessége, hogy

- vegyes jellegü adatokat kell tárolni /szövegesek, nem szövegesek, algoritmusok, programok, stb./,
- az adatok jelentős része átmenetileg /a nyugdijmegállapitás időszakára/ kell, hogy közvetlenül elérhető legyen,
- az adatokkal meghatározott körben /ügyintézők, ellenőrök/ foglalkoznak, akik megfelelően felkészithetők.

A fenti sajátosságokból következik, hogy

- a rendszer jellegzetes használata párbeszédes, tehát interaktiv;
- a terminálok száma és a rendszer terhelése jól tervezhető, kihasználás optimalizálható;

- a kiegyensúlyozottsághoz fontos elhatárolni a rendszer on-line végrehajtandó és elhalasztva /pl. éjszaka/ végrehajtható tevékenységeit, mert a jelentős adatmanipulációs háttértevékenység esetén a számitógép hamar telitődhet és a válaszidők megnövekedhetnek.

6./ Összefoglalás

A példákban felsorolt tényezők és szempontok figyelmen kivül hagyása, vagy helytelen számitásbavétele kiegyensúlyozatlan rendszerekhez vezetne, amelynek következményeként a rendszerek hozamai nem lennének arányban a ráforditásokkal és az egyáltalán lehetséges használhatóság is veszélybe kerülhetne. A nagyigényü, csábitó célkitüzések csalódásra vezethetnek, miután más tényezők hiánya valóraválásukat megakadályozhatja.

Az igénytelen megoldások viszont eleve csalódást okoznak a felhasználónak, aki többet vár a korszerü számitástechnikától. A lehető legmagasabb szinten megvalósitott, de kiegyensúlyozott, továbbfejleszthető rendszerek váltják csak be a hozzájuk füzött reményeket.

Dr.Kelen Lászlóné - Gilicze László
KSH ÁSZSZ

Osztott adatbázis kialakitása CII Honeywell Bull és TPA 11/40 számitógépeken

BEVEZETÉS

Az előadás egy országos hatáskörű intézet anyagforgalmazási rendszeréről, annak most folyó fejlesztéséről kiván beszámolni. A rendszerrel szemben - az intézet speciális helyzete miatt - támasztott igények egyenként is nehezen megvalósithatóak, összességükben pedig egymással ellentmondásosak. A legkevesebb kompromisszumot tartalmazó megoldás osztott rendszerben megvalósuló adatbázis alkalmazására vezet.

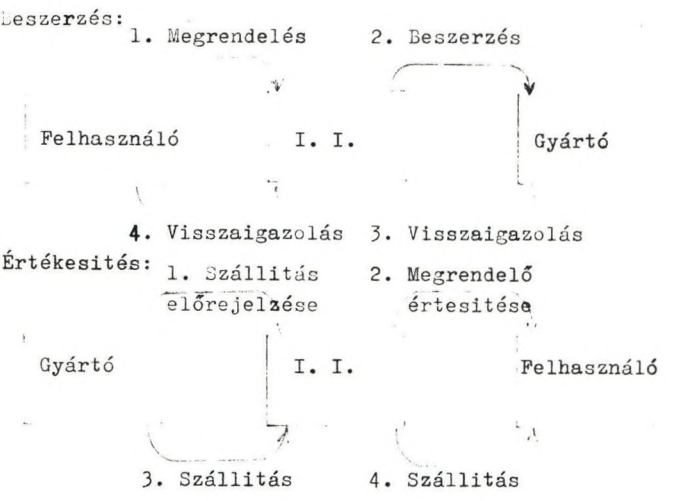
Az alábbiakban ismertetjük a rendszer fő funkcióit, ezek két gép közötti megosztásának szempontjait, a CII Honeywell Bull 66/20D és a TPA 11/40 számitógépeken rendelkezésre álló eszközöket, a kapcsolattartás módszerét és korlátait, és a megvalósitás eddigi tapasztalatait. Az előadásban részletesebben kitérünk a TPA - Honeywell kapcsolat által nyújtott, osztott rendszerek kialakitását támogató, távadatátviteli lehetőségekre.

A RENDSZER FŐ FUNKCIÓI

Az országban a rádióaktiv készitmények nyilvántartása és forgalmazása az MTA Izotóp Intézetének /a továbbiakban I.I./ feladata. A forgalom rohamos növekedése, a gyors, pontos ügyintézés igénye szükségessé tette az ügyvitel számitógépes automatizálását. A megvalósitás há rom, egymástól időben jól elkülönithető részből áll. Alapja egy alapadatnyilvántartó adatbázis /a továbbiakban OINY/. A második rész, jelen témánk, az izotópforgalmazás /továbbiakban IZOFOR/ elkészitése most folyik az ÁSZSZ kivitelezésében. A rendszer teljessé a számlázás automatizálásával válik.

Az OINY adatbázis tartalmazza az I.I. által, fennállása óta forgalmazott valamennyi szállitmány jellemzőit, és feladata, hogy a jövőben szállitott termékeket is nyilvántartsa. Igy ez egy állandóan bővülő állomány. Az adatok közötti kapcsolatok bonyolultsága igen nagy, szükség van a Honeywell IDS adatbáziskezelő nyelvének lehetőségeire.

A nyilvántartási rendszerhez kell illeszkednie az általunk tárgyalt forgalmazási rendszernek, amely a beszerzés és értékesitás alább vázolt ügymenetéből áll:



A számitógépes rendszer készíti az ügyvitelhez szükséges ürlapok, bizonylatok kitöltését, naprakész információszolgáltatást, statisztikai és leltári kimutatásokat, a forgalmazott termékekről, göngyölegekről, ügyfelekről.

A rendszerrel szemben támasztott kritériumok szigorúak, összetettek, és egymással gyakran ellentétes hatásúak voltak. Lehetetlen volt a hagyományos, centralizált rendszer keretében történő megvalósitásuk. Példaként felsorolunk néhány ilyen, egymással ellentétes kritériumot:

- a rendszer interaktivitásigénye oly nagy, hogy feltétlenül szükségessé teszi több terminál állandó on-line használatát;

- az adatbázis fizikai mérete oly nagy, hogy lehetetlen a tartós, nappali, on-line hozzáférés a lemezkapacitás szükössége miatt;

- a nyilvántartás egy állandóan bővülő állomány, míg a forgalmazás évente megújuló állományokat igényel;

- az adatok közötti kapcsolatok annyira összetettek, - ez a kódok és kulcsok felépitésében is tükröződik - , hogy csak igen megbizható adatrögzitéssel müködtethető a rendszer;

- az izotópforgalmazásra vonatkozó előirások változnak, az ügymenet nem eléggé tisztázott, gyakoriak a kivételes eljárások, igy csak rugalmas, könnyen változtatható rendszer fogadható el;

- a rendszer erőforrásigénye egyes szük keresztmetszetek esetében az üzemelés során kritikus terhelést okozhat, ezért csak gondosan megtervezett, és optimálisan kivitelezett müködhet üzemszerüen;

- gyakori a többpéldányos, előre nyomtatott leporellók használata, igy a nyomtatás a központi

nagyszámitógépen csak igen rossz hatékonysággal oldható meg.

A fenti követelmények ellentmondásossága miatt csak olyan kompromisszum valósitható meg, amely több, egymással laza kapcsolatban álló részrendszerből áll. A részrendszerek adatbázisainak illetve adatállományainak kapcsolatát célprogramok valósitják meg: karbantartó, válogató, és átviteli programok.

A rendszer kialakitásához kedvező lehetőséget az intézetünkben kifejlesztett TPA 11/40 CII Honeywell Bull kapcsolat nyújtott. Az I. I.-ben működő kisszámitógépen nemcsak a távoli futtatások és távoli nyomtatás valósitható meg, hanem fájlok átvitele is. Ezenkivül a TPA termináljairól dolgozhatunk a Honeywell tetszőleges interaktiv rendszereivel is. Ezek a különböző TAF módok egyszerre is használhatók a TPA különböző termináljairól, rugalmas összeköttetést teremtve a két számitógép között. Kényelmes lehetőség van a távoli nagygép és a helyi kisszámitógép közötti funkciómegosztás kialakitására.

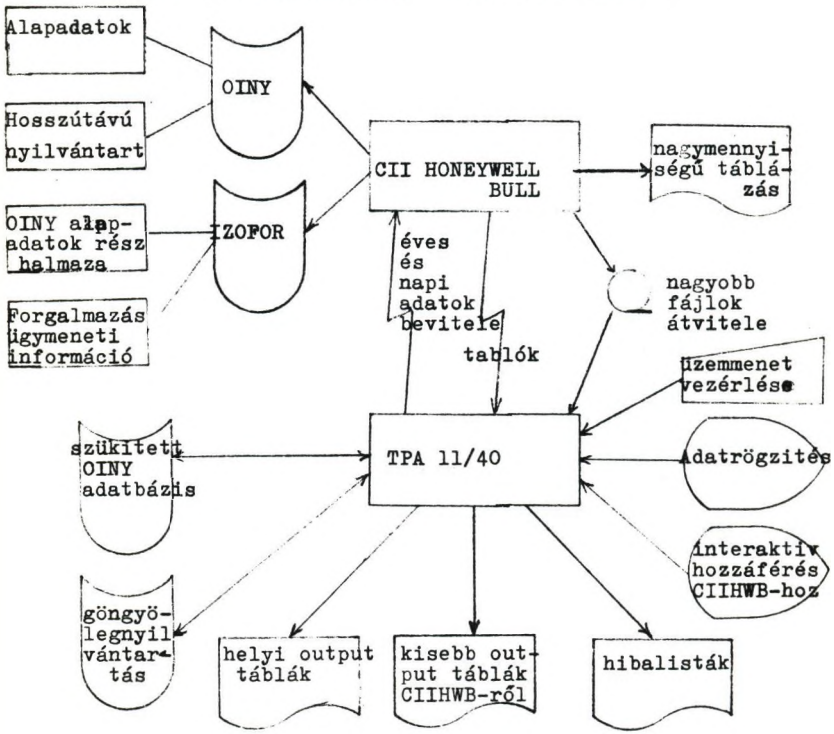
A KÉT GÉP KÖZÖTTI MEGOSZTÁS

A nyilvántartási és a forgalmazási rendszereket az archiválás idejének különbözősége miatt két független adatbázisban valósitjuk meg. Az alapadatnyilvántartásban /OINY/ végzett javitások után automatikusan átvezetjük a módositásokat az IZOFOR adatbázison is. Mindkét adatbázis a Honeywell-en van, tekintettel a jelentős háttértároló igényre.

Mig a nagy kapacitást /háttértároló, CPU idő/ igénylő müveletek a Honeywell-en futnak, a TPA-n valósulnak meg a nagy interaktivitást igénylő müveletek. Ezek során

gyakran van szükség felhasználói döntésekre, és igy leghatékonyabban a felhasználó környezetében végezhetők. Ilyen müveletek az adatbevitel, adatellenőrzés, részállományok listázása, nyomtatás többpéldányos, vagy előre nyomtatott bizonylatokra. A TPA 11/40 szoftver-ellátottsága /DECFORM/ lehetővé teszi, hogy az adatbázisba történő bevitel előtt már a kulcshibákat, és az adatok közötti tartalmi ellentmondásokat is kiküszöbölhetjük. Megjegyezzük, hogy korábban, csoportos adatrögzitéssel /INTERSCAN 2100/ történő bevitelnél ilyent nem lehetett megvalósitani, mert nem volt lehetőség a bevitel során az alapadatokhoz való hozzáférésre.

A feladatok megosztásának sémája a következő:



Az év végére tervezzük az IZOFOR adatbázis rendszertesztjét. Jelenleg a programozási munkák folynak.

Fontos tapasztalatokat szereztünk az egyes feladatok erőforrásigényeire vonatkozóan. Sajnos lényeges döntések és eszközök hiányában előzetes számításokat és kísérleteket csak késve tudtunk elvégezni, és ez az erőforrás-igény szempontjából néhány hátrányos megoldást eredményezett.

Az osztott rendszerben résztvevő eszközök nagy száma természetesen rontja a rendszer üzembiztonságát, hiszen a teljes rendszer üzemidejét az egyes rendszerek üzemidejének közös része adja. Sok a panasz a postai adatátviteli berendezésekre. Ezen a téren a karbantartás megszervezése a legnehezebb feladat, hiszen mintegy féltucat egymástól távol álló szervezet összehangolt munkája szükséges egy-egy hiba biztonságos elhárításához. Emiatt a két gép közötti kapcsolat laza, sok redundanciát tartalmaz, és a távoli vezérlésen és időnkénti interaktiv terminálhasználaton kivül lényegében file-átvitelre korlátozódik.

A fejlesztés során a legnagyobb nehézséget az jelenti, hogy olyan team munkáját kell megszervezni, amelyben nemcsak a szakterületek idegenek, hanem a szakértők is egymástól helyileg és szervezetileg is igen távoliak. Magas szintű koordináció hiányában az érdekek különbözősége is komoly akadályt jelenthet.

Az eddig szerzett tapasztalatok szerint ezzel a megoldással azonban sikerül kielégiteni egy igen bonyolult követelményrendszert. A feladatoknak a két gép közötti ilyen megosztásával, a helyi adatbevitel és rész-adatbázisok segitségével eleget lehet tenni a pontosság, várakozási idők, adatbiztonság igényeinek is.

Kovács Kálmán
MTA SZTAKI

PERSONAL DATA MANAGEMENT SYSTEM
MIKROSZÁMITÓGÉPES ADATBÁZISKEZELŐ RENDSZER

Az MTA SZTAKI-ban 3 éve foglalkozunk személyi számitógépeken használható adatbázis kezelő rendszerek fejlesztésével. A célrendszerek készitésén kivül általános célu adatkezelő rendszerek kidolgozásával is foglalkozunk. Ezek egyike a PDMS.

A PDMS kifejlesztésével két célt szeretnénk megvalósitani. Egyfelől egy könnyen megtanulható, egyszerüen kezelhető komplett rendszert adni a felhasználók kezébe, mellyel komolyabb előképzettség nélkül is felépithetnek és kezelhetnek adatbázisokat. Másrészt a professzionális felhasználók számára szeretnék egy olyan eszköztárat biztositani, mellyel az adatbázisokkal kapcsolatos tipikus feladatok gyorsan megoldhatók, s igy bonyolult célrendszerek készithetők.

A rendszer rövid ismertetése

A PDMS elsősorban olyan területeken használható jól, ahol sok szempont /kulcsok, feltételek szerint kell tudni egy nagy adatbázisállományban gyorsan keresni, és a kiválasztott információkat változatos módon megjeleniteni.

A rendszer egyszerre akárhány független adatbázis kezelését képes ellátni. Minden adatbázis
egy fő adat file-ból,
1-9 "segéd file"-ból,
tetszőleges számu, a fő file rekordjainak rendezését és kulcs szerinti közvetlen elérését biztositó VSAM strukturából,
tetszőleges számu, a felhasználó által definiált szelekciós kirtériumból, report formátumból, a PDMS nyelvén irt speciális tranzakcióból áll.

Az adatok között egyirányu hierarchikus kapcsolat valósitható meg: a főfile rekordjaiból a segédfile-ok rekordjai pointerek segitségével elérhetők.

A rendszer által kezelt adattipusok:
string /max. 250 char/
egész szám /max. 12 jegy/
valós szám /max. 15 jegy/
dátum /1900 jan. 1.-1999. dec. 31./
pointer /fő file-beli adat
segéd file-beli adat/

Egy-egy file azonos szerkezetü rekordokból áll. A fő file maximális rekordhossza 1020 byte, a segédfileoké 250 byte. Az egyes file-ok rekordszáma max. 65535. A főfile szükség esetén több diszken is átnyulhat. A mezők számának összege az összes file-okon nem haladhatja meg a 128-at. A nem string tipusu mezők értékét mindig tömöritve tároljuk.

A rekordok kulcs szerinti elérését egy VSAM /Virtual Sequential Acces Method/ rendszer biztositja. Kulcsként a fő file bármely mezője /string, szám, dátum, pointer tipusu/ vagy annak valamely része használható. Ha a kulcs nem azonositja egyértelmüen a rekordokat, használhatunk 1-3 kiegészitő kulcsot is, de ezeknek sem kell egyértelmüeknek lenni.

Minden rendezéshez egy index file tartozik. A kiválasztott rendezés index file-jának egy - az adatbázis méretétől függő - része memória rezidens, a maradékot egy speciális page-elési eljárás kezeli. A kulcs szerinti elérés sebessége a kulcstól egyáltalán nem függ, egy bizonyos rekordmennyiségen tul pedig a rekordszám növekedése sem befolyásolja, lényegében állandónak tekinthető. Közel egyértelmü kulcsok esetén a keresés /beszurás/ átlagos ideje - sok ezres rekordszám mellett - 1-2 másodperc. Nem egyértelmü kulcsok a rekordelérést lassitják, a szüséges idő arányos az azonos kulcsu rekordok átlagos számával.

Egy adatbázisban egyszerre akárhány ilyen rendezést definiálhatunk, ezeket a rendszer automatikusan karbantartja, és bármikor választhatunk közülük. Ezzel biztosithatjuk a különböző szempont szerinti keresések egyforma hatékonyságát.

Lehetőség van az adatbázisban nem tárolt változók használatára is. Ezek - az adatbázis adataiból és a már definiált változókból képzett - adatok elsősorban riportok készitésénél hasznosak.

Az adatok felvitelére, módositására, megjelenitésére standard eljárások vannak, melyeket a felhasználó szükség szerint átparaméterezhet, ezzel számára egyszerübbé, saját céljainak megfelelőbbé tehet.

Listák készitéséhez egy riport generátort használhat, mellyel tetszőleges formátumu printout-ot /oszlopos lista, levél, boriték cimzés, stb./ készithet. A riportban report-, page- és control header, ill. footer, valamint tetszőleges számitások definiálhatók.

Egy feltétel generáló program segitségével készithet szelekciós rutinokat, melyek a riportok készitésénél használhatók.

A válogatás használhat egy VSAM strukturát, mely kivánt kulcs szerinti elérést biztositja. A feltételekben szerepelhetnek a kulcs/ok/ra, az adatbázis mezőire és a definiált változókra vonatkozó logikai kifejezések.

Az adatbázis minden tranzakciójának van alapértelmezés szerinti input és output egysége /pl. adatfelvitel: klaviatura input, diszk file output, stb./ de ezek átdefiniálására is lehetőség van.

Az egységek:

klaviatura /input/
képernyő /output/
printer /output/
diszk /input, output/
RST 232/V24/ /input, output/

A fenti funkciók megvalósitásakor a felhasználónak semmilyen programozási vagy adatbáziskezelési ismeretekkel nem kell rendelkeznie, ezek a rendszerrel folytatott dialógus segitségével definiálhatók.

A pDMS parancsokkal vezérelhető, teljesen interaktiv rendszer. Egy parancs parancs-szóból, paraméterekből és opció/k/ból álló sorozat. A parancs-szó általában a végzendő tevékenységre utal, a paraméterek a tevékenység tárgyára, az opciók pedig a végrehajtás módjára vonatkozó információkat tartalmaznak.

Részletes magyarázat nélkül felsoroljuk a pDMS parancsait, használatuk szerint csoportositva:

parancsszó	paraméter	opció
HELP	-	[;P]
DIR	[n]	[;P]
STAT	-	-
DEF	DBS VSAM VAR REP COND CHAIN AUTO NEW MOD DISP	[;I=input file]
FIX		[;O=output file]
SHOW	FILE VSAM VAR CHAIN USER	[;P]

DBS	[adatbázis név]	[;N]
FILE	file név	-
ADD	-	[;I=input file] [;u]
MODIFY	{ . / n [-][m] / -n }	[;u] [;I=input file]
DELETE	{ . / n [-][m] / -n }	-
RESTORE	{ . / n [-][m] / -n }	-
LIST	-	[;P] [;u]
QUERY	{ . / n [-][m] / -n }	[;P] [;u] [;O=output file]
CLOSE	-	-
VSAM	[VSAM név]	-
GET	-	[;u] [;P] [;O=output file]
INSERT	-	[;u] [;I=input file]
REPORT	[riport név]	[;C=feltételnév], [;O=output file]
AUTO	-	-
CHAIN	parancslánc név	-
END	-	-

A parancsokkal /vagy parancsláncokkal/ nem megoldható... /feladatok megvalósításához saját tranzakciók irása szükséges, melyhez a pDMS adatbáziskezelő parancsaival kiegészitett BASIC nyelv használható.

A pDMS néhány felhasználása:
Gyógyszernyilvántartási rendszer
Raktári számlázó és készletnyilvántartó rendszer.
Vezetői információs rendszer
Munkaközvetitő rendszer

A rendszer környezete
Hardware: SYSTER, VARYTER, TRS-80 64 kbyte /Z80/
2-4 floppy drive
printer
Software: CP/M, NEWDOS, PASCAL, BASIC

I R O D A L O M

Hannák L. - Kovács K. - Lengyel Tamás: A personal computerek alkalmazási lehetőségei a kórházi adatnyilvántartásban és egyéb egészségügyi területen, 10. Neumann Kollokvium, Szeged, 1980, pp. 77-84.

Kerékfy P. - Ruda M.: Mikrogépek alkalmazása kórházi és rendelőintézeti információs rendszerekben /megjelenőben/, 11. Neumann Kollokvium, Szeged, 1982. dec. 6-8.

Kerékfy P. - Ruda M.: Microcomputer Application in Data Processing Systems /megjelenőben/, 8 th Winterschool of Operating Systems, Visegrád, 1983.jan. 31-febr.4. /Abstracts: p 18./

Kerékfy P. - Ratkó I. - Ruda M.: Microcomputer - based medical information systems /elfogadott előadás/, MEDINFO'83, Amsterdam, 1983, aug. 21-26.

Lengyel T.: A MEDICOR kötés- és terméknyilvántartási rendszere közlés alatt, 1983.

Vargha Dénes
OMFB KFIIR Kidolgozó Csoport

A GÉPI SZÖVEGFELDOLGOZÁS SZEREPE A JAPÁN 5. GENERÁCIÓS PROJEKT REALIZÁLÁSÁBAN

Sokan keresik arra a választ világszerte, hogy mennyire megalapozottak a japánok ambiciózus tervei egy uj, "intelligens" számitógépgeneráció létrehozására. A mesterséges intelligencia megteremtése /értsd: az értelmes viselkedés szimulációja, az intelligens válaszreakciók biztositása/ igencsak a kutatás stádiumában van ma még. Nem az elért eredmények lebecsülését jelenti, ha elismerjük, hogy méreteiben ezek a kutatások egyelőre csak nagyon korlátozott gyakorlati alkalmazást tesznek lehetővé.

A japánok célul tüzték, hogy az évtized végére a világszerte elért "laboratóriumi" eredmények birtokában /nem utolsósorban saját intenziv kutatásaik felhasználásával/ a számitástechnika alapjává tegyék a mesterséges intelligencia alapgondolatait, egységbe ötvözve a különböző területeken elért részeredményeket.

A laboratóriumi méretektől az üzemszerü alkalmazásokig legfőképpen nem azért nehéz eljutni, mintha elvi akadályai volnának a bevált módszerek szélesebbkörü alkalmazásának, mégcsak nem is a technikai lehetőségek növekedési iramával vannak problémák /a rohamos technikai fejlődés inkább elébeszalad a lehetőségek mindenkori kihasználásának/. A "szük keresztmetszetet" sokkal inkább a technikailag kiépithető rendszereknek megfelelő tudásanyaggal való feltöltése jelenti. Igy van ez még a mesterséges intelligencia kutatások eddig talán leg-

eredményesebb, a gyakorlatban leginkább bevált területén, az un. szakértői rendszerek kialakitásában is. A felhalmozódott emberi tudásanyag döntő többsége emberi nyelven leirva, szótárakban, enciklopédiákban, szakkönyvekben, szakcikkekben szunnyad, s csak részleges megoldást jelent az. hogy a hozzáférésben, a szöveges információk kikeresésében igénybe lehet venni a számitógép segitségét. Jelenlegi formájában az a tudásanyag a számitógép számára "élvezhetetlen", és óriási emberi munkaráforditással lehet csak hasznositható vá tenni.

A japán számitógép projekt sikerének kulcsa - s ezt ők igen világosan felismerték - az, hogy létre tudnak-e hozni egy emberi nyelven tanitható rendszert, amely egy "nyers", előzetes előkészités nélküli inputból, addigi ismereteire támaszkodva, ki tudja hámozni az ujabb és ujabb ismereteket, s asszimilálni tudja addigi tudásanyagához. Az emberi segitséget, beavatkozást természetesen, nem zárja ki egy ilyen rendszer alkalmazása - feltéve, hogy arra nem tulságosan gyakran kerül sor. Az alapvető kérdés tehát egy természetes nyelvi interface rendszer megteremtésének lehetősége, beleértve a természetes nyelv alkalmazásából származó pontatlanságok pontositását, a többértelmüségek kiszürését, a hiányos információk kiegészitését, stb./Az emberi közbeavatkozás is emberi nyelven feltett kérdésekre választási alternativákra adott válaszok formájában történhet./

Néhány adat arra vonatkozóan, hogy mennyire intenziv kutatómunka folyik Japánban egy /vagy több/ ilyen rendszer megteremtésére:

- Az 1982-ben Prágában megrendezett 9. Nemzetközi Gépi Nyelvészeti konferencián igen rangos mezőnyben az előadók 23 %-a japán volt /a társszerzőket nem számitva is 16 %/. /Egyedül az Egyesült Államokból volt több előadás; Japán után következett az NSZK, a Szovjetunió, Franciaország és Csehszlovákia./

- A Kyoto-i egyetemen kisérleti jelleggel ugyan, de rendszeresen folyik az INSPEC adatbázisából fizikai és matematikai témáju cikkek cimeinek gépi forditása, 93 %-os pontossággal, s egy japán nyelvü adatbázisrendszer gépi előállitása.

- Ugyancsak a Kyoto-i egyetemen implementáltak egy olyan rendszert, amely az elemzés hatékonyságának javitására nagy mennyiségü szöveg feldolgozásából tanulja a szabályok alkalmazásának optimális sorrendjét.

- 1978 óta sikeres kisérletek folynak Kyoto-ban egy LISP-nyelvü személyi számitógépre kialakitott rendszerrel, amely egy egyszerüsitett Montague-grammatika alapján elkészíti a leforditandó angol mondat λ-kalkulusát, és ennek kiértékelésével létrehozza a mondat japán nyelvü forditását.

- Az Osaka-i egyetemen japán szabadalmi bejelentések angolra forditását végzik egy interaktiv tanulórendszerrel, amely relációs adatbázisba gyüjti a kihámozott ismereteket, és hasznositja a további feldolgozásnál. /A rendszer sebessége jelenleg 3 japán szó/mp./

- A Hitachi cég heurisztikus angol-japán forditórendszer kifejlesztésén dolgozik. A prototipus rendszer egyelőre 10 ezer alapszót használ, ezekből felépitett idiómatárral és többszavas kifejezéstárral továbbá mintegy ezer számitógépes terminussal. A rendszer jelenleg haladó nyelvtanuló diákokkal veszi fel a versenyt számitógépes folyóiratcikkek forditásában, de remélik, hogy rövidesen elérik a szakértői szintet.

Az eddig elért eredmények fényében talán kevésbé tünnek elérhetetlennek 1990-re a japánok által kitüzött célok:

- a gépi forditás olyan szintü megvalósitása, amely 100 ezer szavas szótárral 90 %-os forditási pontosságot ér el, és nyomdakész szöveget állit elő 70 %-os megtakaritással;

- a gépi beszédmegértés segitségével fonetikus "irógép" kifejlesztése, továbbá szöveges /irott vagy szintetizált/ válaszadás, és 50-60 beszélő azonositása, 10 ezer szavas fonetikus szótárt használva az értelmi elemzésre és a félrehallások kijavitására;

- a kérdés-válasz rendszerek olyan prototipusának létrehozása, amely 5000 szó és 10 ezer következtetési szabály alkalmazásával szakterületenként képes a felhasználók igényeit kielégiteni.

Az előadás konkrét példán mutatja be a japánok által alkalmazott egyik elemző-fordító módszert.

Kádár István - Polák István
EFE Földmérési és Földrendezői Főiskolai Kar Székesfehérvár

SIKBELI ÉS TÉRBELI PONTHALMAZOK STRUKTURÁLT TÁROLÁSÁNAK MATEMATIKAI ÉS SZÁMITÁSTECHNIKAI KÉRDÉSEI

Bevezetés

Tekintsük a következő kérdést: "Mennyi információ van egy gépkocsiban?". Azt hihetnénk, hogy erre a többszörösen határozatlan és sokszorosan pongyola kérdésre még akkor sem lehet érdemleges választ adni, ha a gépkocsiba felejtett ismeretlen tartalmú aktatáskákat, újságokat és más egyéb információhordozókat már eleve kihagyjuk a számitásból. Akadt azonban még 1959-ben valaki, aki a fenti kérdést csak kicsit pontositva, de a vele kapcsolatos fogalmakat már nagyon absztrahálva, a gépkocsi ún. szerkezeti információtartalmának számitásához meglehetősen egyszerü elemi összefüggéseket kapott. /Jacobson,H.: The information content of mechanisms and circuits. Information and Control. Vol.2. 1959. 285-296/ Ha ugyanis feltételezzük, hogy egy gépkocsi nem más, mint m_1 darab 1-es tipusú, m_2 2-es és m_n n-es tipusú /ismert szerkezetü/ alkatrész céliránysosan elhelyezett kombinációja, majd az egyes alkatrészeket súlypontjaikkal helyettesitjük, akkor gépkocsi helyett az előadás cimében is szereplő olyan térbeli ponthalmazhoz jutunk, melyben az azonos tipusú alkatrészeket azonos, a különbözőeket szines pontok jelképezik. Jobb azonban, ha geometriai pont helyett egy-egy fizikai pontot /pl. egy kicsiny téglatestet/ képzelünk el minden alkatrész súlypontjában, hogy ezek megfelelő beállitásával a szóbanforgó alkatrész térbeli tájékozását is szemléltetni tudjuk, hacsak nem akarunk mindenáron a hatdimenziós tér geometriai pontjaihoz fordulni. A szerkezeti információ bitekben kifejezett értékét végülis a szines ponthalmaz /alkatrészhalmaz/ összeszerelés előtti és összeszerelés utáni rendezetlenségi mérőszámainak különbsége fogja megadni.

Többszinü /vagy egyszinü/ ponthalmazok persze nemcsak gépkocsi-alkatrészekből születhetnek és nemcsak arra valók, hogy valaki információtartalmukat kiszámitsa. A legkülönbözőbb mozgó vagy mozdulatlan térbeli objektumok alakjának, méretének, kölcsönös vagy abszolut helyzetének ismerete annyira fontos dolog, hogy az ún. geometriai modellezés, ill. a geometriai alrendszer csaknem minden adatbázis nélkülözhetetlen elemévé válik. A ponthalmaz-modell ez esetben az egyes objektumok geometriai jellemzőit kivánja kiemelni, minden egyéb, más tulajdonságát elhanyagolva. A geometriai modellezés egyik főkérdése: a ponthalmazok egyértelmü leirásához szükséges ún. helyzeti vagy koordináta-információ hatékony tárolásának és visszakeresésének elvi és módszertani tisztázása. Ehhez a feladathoz kivánunk majd előadásunkkal is hozzájárulni, szemelvényeket mutatva be az e téren végzett kisérleteinkből. A kérdéskör azonban annyi új és szokatlan probléma vizsgálatával jár együtt, hogy itt kénytelenek vagyunk csupán ezek felsorolására szoritkozni.

Az alábbiakban megpróbáljuk mondanivalónkat a cimnek megfelelően két csoportra: matematikai és számitástechnikai kérdések csoportjára bontani, habár tisztában vagyunk azzal, hogy közöttük éles határvonal nem húzható, hiszen a szá-

tástechnika kis túlzással kisérleti, ill. gyakorlati matematikának, a matematika pedig a számitástechnika elméleti oldalának is tekinthető. Munkánkat gondolatébresztésre, vitára ingerlésre szántuk, ezt pedig csak úgy érhetjük el, ha nem kész, nem kerek, nem tévedhetetlennek tartott gondolatokat közlünk, hanem hipotéziseket, elképzeléseket vegyitünk tapasztalatokkal, kisérleti eredményekkel, véleményekkel úgy, ahogy az életben is mindig megelőzi, sőt kiváltja a fejlődést az alkotó fantázia, az óhaj, az igény és általában csak sokkal később követi a megoldás és a megvalósitás.

Matematikai kérdések, hipotézisek, tapasztalatok, vélemények és eredmények

1. Az információtartalom egy közlemény terjedelmének, hosszának alsó határa. Elméleti fogalom, de a gyakorlat számára megközelitendő és megközelithető célt mutat, memóriahelymegtakaritásra, sőt néha feldolgozási idő-csökkentésre is ösztönöz. Végtelen sok úton közelithető meg, nyelvtől, kódrendszertől, mondanivalónk sorrendjétől teljesen függetlenül. Az a tény, hogy végtelen sok optimális nyelv létezik, még korántsem jelenti, hogy minden nyelv optimális. Csupán arra figyelmeztet, hogy nincs "egyedül üdvözitő út", nincs olyan "legjobb nyelv" /kódrendszer, számrendszer, koordinátarendszer, struktúra stb./, mely egyeduralomra tarthatna számot változó körülmények mellett is. Lehet, hogy adott viszonyok egyértelmüsithetik a választást, de a mai világban vagy a tervezett jövőben hol vannak, hol lesznek adott, változatlan viszonyok?

Egy térbeli ponthalmaz teljes információtartalmát felölelő leirás, közlemény hossza, memóriaigénye - optimális kódolás esetén - nem függ attól, hogy milyen /derékszögü, poláris, görbevonalú stb./ koordinátarendszert használunk, hogy a koordinátarendszert hogyan tájékozzuk, hogy a koordinátákat milyen számrendszerben adjuk meg, hogy egy-egy pont rögzitéséhez egy, két vagy három koordinátát használunk, hogy differenciált vagy integrált tárolási strukturát alkalmazunk stb. Az optimális kódolás ugyanis egy olyan "csodaszer", mely minden ellentmondást fel képes oldani.

2. Ponthalmazok részére három alapeloszlás /alapstatisztika/ ismeretes. /Vilenkin, N.J.: Kombinatorika. Bp. 1971. 102./ Az egyenletes eloszlás és a függetlenség mindhárom alapfeltétele. A Maxwell-Boltzmann /vagy KK/ statisztika: különböző /vagyis egyedi azonositókkal ellátott, például sorszámozott/ pontoknak ugyancsak különböző /koordinátájú/ ponthelyekre vonatkozó eloszlása, pontosabban az eloszlás sürüségfüggvénye, melynél az ugyanazon ponthelyen lévő /azonos koordinátájú/ pontok számát semmi sem korlátozza. A Bose-Einstein /vagy NK/ statisztika: egyféle /egyedi azonositóval nem rendelkező/ pontoknak különböző ponthelyekre vonatkozó eloszlása, ahol az azonos koordinátájú pontok mennyiségét az előbbi statisztikához hasonlóan semmi sem korlátozza. A Fermi-Dirac /vagy NKl/ statisztika: mint az előző, de egy ponthely legfeljebb egy pontot tartalmazhat, vagyis azonos koordinátájú pontokat nem türő eloszlás.

Fenti statisztikák a statisztikus mechanikában leggya-

koribb jelenségek általánositásaként születtek, és korántsem meritik ki az összes hasznos esetet. Kiegészitésül ide kivánkozik még három statisztika. A KN-statisztika értelemszerüen a Bose-Einstein statisztika duálisa, melynél különböző pontok meg-nem-különböztetett ponthelyekkel kerülnek kapcsolatba minden megkötés nélkül. A KN1-statisztika esetében egy - egyébként ismeretlen ponthelyre - legfeljebb egy azonositott pont kerülhet, vagyis semmivel sem ad több információt, mint mikor a pontok n és a ponthelyek m számát megadjuk. Az NN-statisztika tartalma nem más, mint a ponthelyek gyakorisságeloszlásának leirása a bennük lévő pontok darabszáma szerint.

A nagyszámú hasznos értelmezés és alkalmazás közül itt csak néhányat emlitünk meg. Az NK-statisztika a népsürüségtérkép tartalmával egyenértékü, mig az NK1 ugyanennek maximalista változata. A KN-statisztika pl. az egy helyen /közös lakásban/ élés relációjának leirására vagy a társtulajdonosi reláció megadására szolgálhat. A hatféle statisztika közül maximális információkapacitással a KK-statisztika rendelkezik, mely pl. már egyedi /népesség-, lakás-, ingatlan- stb./ nyilvántartásnak is alapjául szolgálhat. Az eddig még nem emlitett KK1-statisztika a KK azon maximalista megoldása, melynél a ponthelyek finomitásával elérhető, hogy a nyilvántartott személynek, dolognak, tárgynak külön-külön helye, cime, koordinátája legyen. Az NN1-statisztika tartalma sem többet, sem kevesebbet nem mond a KN1-nél, amennyiben a pontok azonositása folyamatos sorszámozással történik.

Érdemes lenne a ponthalmaz-statisztikákat még tovább, olyan mélységig kiépiteni, hogy reális célként lehessen kitüzni: minden statisztikai nyilvántartás az egyre bővülő igény mértékében egyre inkább egyedi nyilvántartássá nőhessen át és forditva. A lehetőségtipusok modellezése /a fenti nyolc alapstatisztika tovább-bővitése/ a "fix pontokat" rögzitené ehhez a munkához.

3. "Amikor a tizes számrendszerben felirt többjegyü szám értelmét magyarázni akarjuk, ezt polinom alakban szokás felirni. Ilyenkor a számot a 10 alapú hatványoknak egy-egy számjeggyel szorzott szorzatösszegeként állitjuk elő, a kitevők fogyó sorrendjében felirva. A polinomalakból úgy kapjuk meg a helyiértékes alakot, hogy a számjegyeket szorosan egymás mellé irjuk, s minden mást elhagyunk. A polinomalakot és a helyiértékes alakot összehasonlitva észrevehetjük, hogy a tizes rendszerben leirt többjegyü számok valójában három müveletet rejtenek magukban: összeadást, szorzást és hatványozást, ezeket pótolja a sorrend és az a megállapodás, hogy a hatványalap mindig 10. De a müveleti jelek hiánya ellenére itt valóságos müveletek szerepelnek." /Ruzsa I.: A matematika és a filozófia határán. Bp. 1968. 55./

Az algebrában az előjel, a zárójelek, a többjegyü számokat szétválasztó szóköz, vessző, pont vagy pontosvessző mind-mind mellőzhetők, mivel ezek funkciója külön jel alkalmazása nélkül is biztositható, betölthető ahhoz hasonlóan, ahogy a többjegyü számok poziciós irásmódjánál a ténylegesen ki nem irt, de értelmezett helyiértékek, müveleti jelek - fenti idézet szerint - hallgatólagosan értelmezhetők. Igy

jönnek létre a sign-free /előjel-mentes/, bracket-free /záró-jel-mentes/, comma-free /változó kódhosszúságú/, coordinate-free /koordináta nélküli/ stb. rendszerek. Azt a tényt, hogy minden sokjegyü szám egy-egy fastruktúra, forditva is lehet értelmezni, vagyis minden fastruktúrát egy-egy sokjegyü számmal lehet kódolni. Ez a megállapitás akkor is érvényes, ha a fastruktúra egy egész ponthalmazt kiván ábrázolni, sőt akkor is, ha pontok helyett más geometriai alakzatok /pl. egyenes szakaszok, egyszerü sikidomok vagy térbeli alakzatok/ szerepelnek.

Megtehetjük például azt is, hogy a fastruktúra pontonkénti /és nem - ahogy szokásos - koordinátánkénti/ összeállitása előtt a sik vagy a tér pontjait a közöttük értelmezett geometriai vagy vektormüveletekkel együtt homomorf módon leképezzük a számegyenesre. A sokjegyü szám képzése ezután ugyanugy történhet, mint valódi skaláris számok esetében. De eljárhatunk ugy is, hogy a sikot vagy teret közvetlenül parkettázzuk /azaz számsikot ill. számteret hozunk létre/ és skaláris mennyiségeket vagy koordinátákat egyáltalán nem használunk.

"Maguk a matematikai és logikai törvények sem immunisak a revizióval szemben, ha egész fogalmi sémánk lényeges egyszerüsitése válik azükségessé. Igy a matematika és a logika törvényei minden "szükségszerüségeik" ellenére hatálytalanithatók." /Quine, W.O.: A logika módszerei. Bp. 1968. 18-19./

Számitástechnikai kérdések, tapasztalatok, lehetőségek, vélemények és eredmények

1. Egy ponthalmaz információtartalmát sokféleképpen irhatjuk le. A leirás módszerétől függetlenül minden esetben pontok és ponthelyek viszonyát adjuk meg többé vagy kevésbé áttekinthető formában. A hagyományos leirásnak két - egymással ellentétes, de egymást kiegészitő - alaptipusa a következő. A "mi-hol-van" nyelv mindegyik mondata egy-egy pont helyzetét egy vagy több koordinátával /ponthely-sorszámmal/ adja meg, ezért a leirás hossza a pontok számával és a ponthelyek számának logaritmusával arányos, azonban a pontok /egyedi vagy csoportos/ azonositóinak számától független. A "hol-mi-van" nyelv mindegyik mondata egy-egy ponthely tartalmát a pontok azonositóival adja meg, ezért a leirás hossza a ponthelyek számával és a pontazonositók számának logaritmusával arányos, azonban a pontok számától független.

Ha semmit sem tudunk a ponthalmazról, akkor a fenti leirások mellett még bizonyos rendszerparamétereket /eloszlások sürüségfüggvényeit stb./ is tárolni szükséges, éspedig másokat az első és másokat a második nyelv esetében. Ezek figyelembevételével igazolható - ami egyébként az információakkumuláció alaptörvényéből is következik -, hogy a kétféle nyelven megfogalmazott ponthalmaz-leirás információtartalma pontosan egyenlő, ennélfogva optimális kódolás /optimális azonositó- és koordinátarendszer/ esetén a közlemény hossza is közel egyenlő kell legyen. A gyakorlatban kialakult eljárások azonban messze esnek az optimumtól, ezért e két nyelv hatékonysága is erősen különbözik.

A "mi-hol-van" nyelv a geodézia, a "hol-mi-van" nyelv

a kartográfia évszázadok óta használt nyelve. Volt idő tehát megmerevednie. Egyébként is a memóriahellyel való takarékosság a számítógépek megjelenése előtt a papirral való takarékosságot jelentette, és hol van az a hivatal, amelyik a papirral takarékoskodik? Az igazat megvallva, a kartográfia esetében nem is szabad a papirral takarékoskodni, mert ez általában csak a szemléletesség, a kifejezőkészség kárára történhet. De vajon helytálló-e ez a megállapitás a digitális papir, a számitógépmemória, a digitális térkép esetében?

Lehet a memóriát digitális számitógép esetében is "rajztáblának" tekinteni, és az analitikus vagy koordinátageometria számitási módszerei helyett ábrázoló geometriai eljárásokat, szerkesztéseket alkalmazni. /Frolov, Sz.A.: Kibernetika i inzsenernaja grafika. 2. kiad. Moszkva. 1974. / Ilyenkor a ponthalmazt és az ugyancsak ponthalmaznak tekintett egyenes vagy görbe vonalakat két /esetleg három/ dimenziós logikai tömbként tárolják, ahol az egyes pontokat zérus háttérben elhelyezett 1-es számjegyek /logikai értékek/ képviselik. Különböző szinü vagy szinfokozatú pontok esetén már aritmetikai tömbre és kettőnél több kódszámra van szükség. A grafikus képernyők és a grafikus tervező rendszerek megjelenésével az ábrázolás fenti módja terjed, sőt a képfeldolgozásban input nyelvként csaknem egyedüli, output nyelvként is nagy jelentősége van.

A rajztáblán ábrázolt pontok, vonalak mögött lévő háttérszint /vagyis az üres papirt/ csaknem teljes egészében meg lehet takaritani, ha az előbb tárgyalt "hol-mi-van" nyelvről a "mi-hol-van" nyelvre térünk át. Előbbinél ugyanis a szükséges memória mértéke csak a ponthelyek számától függ és a pontok, vonalak mennyiségétől független, mig utóbbinál forditott a helyzet: a memóriaigény a pontok számával egyenesen arányos. Azok a külföldi és hazai tapasztalatok, amelyek nagyterjedelmü ponthalmazokkal, ill. az azokban lévő nagyszámú /milliós nagyságrendü/ pontokkal végzett kisérletekből származnak, azt mutatják, hogy a fenti két alapnyelv mind hatékonyság, mind kezelhetőség szempontjából szélsőségesen rossz tulajdonságú. Közös hibájuk, hogy struktúrálatlanok és hogy a ponthalmazok információs strukturájához nehezen illeszthetők. Ezért a gyakorlatban gombamódra fejlődtek ki az olyan kompromisszumos nyelvek, melyeknél a társszükséglet sem a ponthelyek, sem a pontok számával nem arányos, de nem is független tőlük. A síkban a 4-es fa /quadtree /, térben a 8-as fa /octtree/ a legkézenfekvőbb megoldás, mindkettőnél igaz, hogy a tárigény a a ponthelyek ill. pontok számának csak a logaritmusával arányos. Mi a 9-fa és a 27-fa megoldást részesitve előnyben, ezekhez fejlesztettük ki az un. nanovektoriális ill. alfavektoriális /koordinátamentes/ sikbeli ill. térbeli számrendszerünket és ezekben végeztünk struktúrált ponthalmaztárolási kisérleteket.

2. Ahogy az anyag jelenléte befolyásolja a tér geometriai szerkezetét, úgy kell hogy befolyásolja egy-egy új pont bevonása vagy régi elhagyása a ponthalmaz "koordinátajegyzékének" nemcsak a hosszát /sorainak számát/, hanem maguknak a koordinátáknak a szerkezetét, számszerü értékét is. A befolyás persze lokális kell legyen, mely alatt azt értjük,

hogy a változás helyétől távolodva, a változás hatásának statisztikai értelemben egyre inkább gyengülni kell. Einstein tanitása szerint anyag tér nélkül és tér anyag nélkül nem létezik, hisz a tér az anyagnak egy bizonyos vonatkozású absztrakciója. Az is következik ebből, hogy kisebb anyag kisebb, nagyobb méretü pedig nagyobb teret kiván. Ettől függetlenül azonban adathordozó adat nélkül, ill. adat információ nélkül nagyon jól elképzelhető, sőt az is igaz, hogy nagy adathordozón a kevés adat is elfér, hisz "az információ sem nem anyag, sem nem energia ..." - mondaná Wiener Norbert. Azonban ez esetben Einsteinnek van igaza Wiener Norberttel szemben. Lehet ugyanis olyan véges koordinátarendszert /véges teret/ szerkeszteni és a gyakorlatban előnyösen alkalmazni, melynek információkapacitása a benne ábrázolt pontok információigénye szerint változik. Ha például egy újabb pont valamelyik régebbik pont közelében születik meg, kevesebb teret, vagyis információt kér maga számára, mint az, amelyik távolabb lát napvilágot. Ez teljesen érthető is azok számára, akik sohasem tanultak geodéziát és nem fertőzte meg őket a maradiság szelleme /Taylor,E.F.-Wheeler,J.A.: Téridő-fizika.Bp.1974.Mese a földmérőkről c. fejezet/, hisz a közelebbi pontot kevesebb számjegyből álló koordinátával lehet bekapcsolni a legközelebbi régi ponthoz, mint a távolabbit. Az ilyen "anarchikus"-nak nevezett megoldást azonban utasitások és szabályzatok tiltják még a számitógép részére is, márpedig változtatási szabadság nélkül sose lesz "rend" az "anarchiából". Mert tudvalevő dolog, hogy csak addig anarchia a "káosz", amig szükségességét fel nem ismerjük és el nem ismerjük, de felismerése és elismerése után a káoszt anarchia helyett rendnek nevezzük. A gondolatmenet tanulsága: az anyag és a tér viszonyát még tudati szinten, koordinátarendszerek tervezésénél és üzemeltetésénél sem szabad megsérteni.

A dolgok belső logikáját bántjuk meg azzal, ha a sikot nem sikkal és a teret nem térrel, illetve az eseményeket nem eseményekkel mérjük. A terület, a térfogat ill. az ürtartalommérés jól bevált mértékegységei követendő példát szolgáltatnak a korszerü helymérés azaz helymeghatározás struktúrájára, sőt ennek kialakitása érdekében közvetlenül általánositathók. Ugyanis a terület- és térfogatmérés szokásos formáját /struktúráját/ változatlanul átvéve, csupán információtartalmának gazdagitásával egy objektum területén és térfogatán kivül annak alakját, méretét, helyzetét stb. egyetlen sokjegyü számmal integrált módon le lehet irni.

A meglévő rendszerekkel szembeni aggályainkat az alábbi kérdésekben foglaljuk össze:

Mennyire takarékos egy olyan rendszer /antilakótelep vagy antitemető/, ahol egy nemlétező /még meg sem született vagy már elpusztult/ pont számára is /azaz hazánk minden négyzetcentiméternyi darabkájára/

ugyanannyi számjegyből álló koordinátapár ill. -hármas van fenntartva /tartalékolva/, mint a létező /élő/ pontok számára?

Mennyire gazdaságos egy olyan rendszer, ahol az igen gyakran használt pontok számára is

ugyanannyi számjegyből álló azonositók, sorszámok vannak használatban, mint az igen ritkán használt

pontok számára?
Mennyire illeszkedett egy olyan rendszer információ- és adatstruktúrája, ahol geometriai azonositók gyanánt kicsiny objektumok esetében is
ugyanannyi számjegyből álló geokódot terveznek, mint az igen nagy objektumok esetében?
Mennyire következetes egy olyan rendszer, ahol skalár, vektor ill. mátrixmennyiségek kódolásához nem a legtermészetesebb megoldást: skalár, vektor ill. mátrix számjegyeket alkalmaznak, hanem még a számitógépek korában is
ugyanazon skalár számjegyekből alkotott, megkövesedett skalár számrendszerekhez ragaszkodnak?

Befejezés

Korunkban nem feltétlenül az az igazi tragédia, ha az ember olyas valamit akar, ami eleve lehetetlen. Épp ellenkezőleg: az, ha olyasmit akar, ami eleve lehetséges. Éspedig azért, mert a lehetséges többnyire nem más, mint elfogadott szabvány, futószalagon gyártott tömegcikk, nincs belőle testre és lélekre szabott méret, egyetlen konkrét igényt sem elégit ki. Ezért nem lehet igazán rendeltetésszerüen használni. És ahelyett, hogy a lehetetlen /értsd: ez idő szerint lehetetlen/ elérésére sarkallna, olykor még a lehetséges utáni vágyat is kioltja. Holott a lehetetlen akarása nélkül éppenséggel lehetetlen az a valami, aminek közkeletüen haladás a neve.

Ezért a realizmus csak a megalkuvók, a maradiak értelmezésében lesz azonos a fennálló elfogadásával. Mert aki a fennállót elfogadja, az az adott valóságot tiszteli, másszóval "megkövült véleményeket" tisztel. A valóság azonban minden alkotó számára a lehetőség határán túl kezdődik, ezért a realizmus nem más, mint tisztelet helyett irónia.

Nagy állami nyilvántartásaink nemcsak tudománytalanok, hanem egyenesen tudományellenesek. A kivételek igen ritkák. A bevallott /deklarált/ célkitüzések és a helyi, csoportérdekek között a dilatáció egyre mélyül, és ezt a végtermékek is erősen megsinylik. A korszerüsitést legtöbbször a forma kicserélésével hajtják végre és jó, ha a tartalom megmarad. Rosszabbik esetben ugyanis a formaváltás a tartalom rovására történik. Néha csak a cégtáblát cserélik ki "adatbázis"-ra - mint pl. az ingatlannyilvántartásnál - de a régi forma, régi tartalom megmarad, ill. megmaradna, ha különféle ürügyek cimén - gépi, anyagi, személyi korlátok - tovább nem szegényedik. Csak rá kell nézni egy 50-100 évvel ezelőtti térképre és egy korszerünek nevezett mairal a pontosságromlás ill. -rontás mellett a tartalom szegénysége mindenki számára nyilvánvaló. A tudomány öntörvényü fejlődése persze akkor sem szakad meg, ha a pragmatikus gyakorlat teljesen figyelmen kivül hagyja, viszont a mulasztások halmozódása végső soron súlyos lemaradáshoz vezet.

Nyilvántartásaink másik rákfenéje az egység, egységesség fogalmának hamis, káros értelmezése. Mielőbb vissza kellene adni ezeknek eredeti, hasznos értelmezésüket. "Harcolni kell az ellen, hogy mindent egy kaptafára húzzunk, és harcolni kell minden olyan kisérlet ellen, amely felülről akar egyformaságot bevezetni. ... Az egységet az alapvető, a fontos, a lényeges kérdésekben nemcsak nem zavarja, hanem biztositja a változatosság a részletekben, a helyi sajátosságokban ..." /Bozsik V.: Emberszabású szervezet. Bp. 1983. 7./

Alexits Gy., Kerékfy P., Rákóczi F., Ruda M.
MTA SZTAKI

MICRO - SHIVA
NÉHÁNY MIKROGÉPES ADATFELDOLGOZÁSI ESZKÖZ ISMERTETÉSE

E dolgozatban néhány mikrogépalkalmazási tapasztalat felvázolása mellett bemutatunk egy konkrét mikrogépes adatfeldolgozási rendszert /micro-SHIVA/. Tapasztalataink elsősorban egészségügyi információs rendszerek köréből származnak /1-3, 5, 6/, de más mikrogépes fejlesztésekre is hivatkozhatunk /4,7,8/.

A mikrogépes adatfeldolgozás - és általában a mikrogépfelhasználás - a személyes hozzáférhatőség miatt lényegesen különbözik a hagyományos számitógépfelhasználás módjától. Éppen ezért a mikrogépfelhasználás módjától. Éppen ezért a mikrogépfelhasználók számára a klasszikus számitástechnikai eszközökön tul olyan lehetőségeket kell létrehozni, amelyek kihasználhatóvá teszik a személyes gépfelhasználás előnyeit, figyelembe véve a jelenleg elérhető mikroszámitógépek kis kapacitását.

A szerzők által fejlesztett Micro-SHIVA rendszer /mikrogépes Sokoldalu, Hatékony, Intelligens, Vizuális Adattár/, ezeknek a szempontoknak a figyelembevételével készült.

Fő célkitüzéseink a következők voltak:

Olyan rendszert kivántunk létrehozni, amely különböző számitástechnikai ismeretek nélkül is biztonságosan kezelhető. Ugyanakkor az ember-gép kapcsolat lehetőleg egyszerü és gyors legyen. Emellett arra is törekedtünk, hogy a kis gépkapacitásból - viszonylag kis sebesség, kis térkapacitás - adódó hátrányokat semlegesitsük.

1. A kényelmes és biztonságos munka támogatása

A munka hatékonyságának egyik legjobb biztositéka egy "vizuális ember-gép kapcsolat" lehet. A felhasználó és a számitógép kapcsolatát jól áttekinthető, képernyőre vetitett formátumok /adatlapok, táblázatok, ábrák/ segitik. Ha ezek a formátumok "full screen" módban kezelhetők, akkor egy valóban dinamikus ember-gép kapcsolat eszközeivé válhatnak.

Néhány példa:

Az adatfeldolgozás egyik legfontosabb momentuma az adatok gépre vitele. Ha az adatok nem valamilyen közvetlenül gépre vihető formában jönnek létre, akkor legcélszerübb, ha az adatokat szolgáltató személy - pl. kódlapok kitöltése helyett - közvetlenül egy a képernyőre vetitett "kódlapra" viszi az adatokat. Igy közvetlenül részt vehet a gépi feldolgozásban, ami az adatszolgáltatás pontosságát, értékét nagymértékben növeli.

Ugyanugy egy rendszer futtatása, az elvégzendő feladatok kijelölése is segithető. Egy bonyolult feltételrendszer kijelölése és a számitógép számára történő pontos leirása igen munkaigényes és sok hibalehetőséget magában rejtő feladat. Ha azonban ez egy jól áttekinthető adatlapnak /képernyőn történő, ellenőrzött/ kitöltésével történik, akkor a feladat nagymértékben leegyszerüsödik.

Egy számitógép feldolgozásban szereplő adatok megjelenitéséről /printer, képernyő/ is fontos szempont a megfelelő, könnyen kezelhető formátum biztositása. Képernyő esetén ez a formátum mozgását, a képernyőnél nagyobb méretü formátumon történő navigálást is szükségessé teszi.

A felsorolt célok érdekében ltérehoztunk egy képernyőformátum szerkesztő és megjelenitő rendszert. Ez a Micro-SHIVA egyik alapvető eleme. A képernyőszerkesztő alrendszer egy full screen editor, amely adatmezők kijelölését is lehetővé teszi. A megszerkesztett és lemezen tárolt képernyőformátum felhasználását, a formátumon keresztül megvalósuló ember-gép kapcsolat létrehozását egy formátummegjelenitő alrendszer biztositja. Ez teszi lehetővé az ellenőrzött adatbevitelt, a futtatási feltételrendszerek - szintén ellenőrzött - leirását, a tárolt adatok megjelenitését.

2. A hatékony müködésmód biztositása

Az ember-gép kapcsolat hatékonyabbá tétele mellett a számitógép müködés hatékonyságának fokozása is fontos cél. A mikrogépek központi egységének viszonylag kis sebessége személyes használat esetén elfogadhatóvá válik, ha gépi kódra forditott programokat használunk. Kritikusabb kérdés a szokásos floppy háttér kis sebessége és kis kapacitása. A kis átvételi sebesség miatt - ha gyors müködést kivánunk elérni - le kell mondanunk a bonyolultabb file-szervezési lehetőségekről.

Egy másik szük keresztmetszet a floppy lemezek kis tárkapacitása. /A több Mbyto-os hajlékony lemezek és a winchester tárak jelenleg még nem hozzáférhetők./ A kisteljesitményü olcsó mikrogépeknél néhány száz kbyte háttérkapacitással kell számolni. Különösen fontossá válik igy a tömör adattárolás.

A Micro-SHIVA rendszerben a gyors müködés érdekében a programok Z80 assemblerben készültek. A file-kezelést leegyszerüsitettük, kulcstáblázatokat csak a memóriában tárolunk. A tárterület optimális kihasználása érdekében intenziv adattömöritést alkalmazunk. Ez a tömörités adatfüggő. Az adat tipusától és értékétől függően más és más módon történik.

3. Összetett rendszerek problémái

A kiskapacitásu floppy tár hátránya talán a több file-os /egy lemez terjedelmét meghaladó/ rendszereknél mutatkozik meg a legjobban. A terjedelmes rendszerek feldolgozása megoldható az adatbázis több lemezen történő elhelyezésével ugy, hogy egy-egy munkafázishoz szükséges adatok egyidejüleg elérhetők legyenek. Meg kell azonban szervezni a nem egyidejüleg elérhető adatbázis-részek módositásainak szinkronizálását. A micro-SHIVA egy központi adatnyilvántartó file segitségével biztositja a változók szükséges átvitelét a különböző adatbázis-részek között. Igy lehetővé válik, hogy egymástól független felhasználók, különböző időpontokban bevitt adatai valamennyi /illetékes/ felhasználó számára hozzáférhetők legyenek.

Hivatkozások

1. Alexits Gy., Kerékfy, P., Rákóczi F., Ruda M., Egy többkomponensü kardiológiai gondozási rendszer megvalósitása kisteljesitményü mikroszámitógépen. I. Egészségügyi Informatikai Vándorgyülés, Szekszárd, 1983. május 5-6.

2. Hannák L., Kovács K., Lengyel T., A personal computerek alkalmazási lehetőségei a kórházi adatnyilvántartásban és egyéb egészségügyi területeken. 10. Neumann-kollokvium, Szeged, 1980. Előadás kivonatok, 77-84.

3. Kerékfy P., Ruda M. Mikrogépek alkalmazása kórházi és rendelőitnézeti információs rendszerekben /megjelenőben/. 11. Neumann-kollokvium, Szeged, 1982. dec. 6-8.

4. P. Kerékfy, M. Ruda: Microcomputer Application in Data Processing Systems, 8 th Winterschool of Operating Systems, Visegrád, 1983. jan. 31-febr. 4., Abstracts, p. 18.

5. P. Kerékfy, I, Ratkó, M. Ruda, Microcomputer-Based Medical Information Systems, MEDINFO 83, Amsterdam, 1983. aug. 21-26.

6. P. Kerékfy, M. Ruda, Medical Information Systems on Desktop computers, Conference on Computer-Based Information Servicing, Várna, 1983. okt. 3-8.

7. Kovács K., pDMS mikroszámitógépes adatbáziskezelő rendszer, II. Neumann Kongresszus, Székesfehérvár, 1983. nov.

8. Lengyel T., A MEDICOR kötés- és terméknyilvántartási rendszere a HP85A professzionális personal computeren, II. Neumann Kongresszus, Székesfehérvár, 1983. nov.

Ungár János
VIDEOTON Elektronikai Vállalat
Számitástechnikai Gyár
Fejlesztési Intézete (VFI)

Szoftver eszközök adatbázis kezelő rendszerek megbizhatóságának javitására és adatai integritásának megőrzésére

Tartalom:

1. Bevezető

Valamely számitástechnikai rendszer megbizhatóságát meghatározhatjuk a berendezéseknél, készülékeknél szokásos módon: azaz ugy, hogy menynyire folyamatosan képes nyujtani a felhasználó által elvárt szolgáltatásokat /1./.

Az adatbázis kezelő rendszerek (ABKR) legjellemzőbb sajátsága, hogy egy szervezet teljes - általában igen nagy értéket képviselő - adat erőforrását teszik hozzáférhetővé a szervezet valamennyi érdekelt felhasználója számára.

Nyilvánvaló, hogy ezért az ABKR-től a felhasználók magasfoku megbizhatóságot várnak el.

Már az ABKR tervezésekor nagy figyelmet kell szentelni a megbizhatósági kérdéseknek. Számba kell venni a lehetséges hibákat, előfordulásuk gyakoriságát, kezelésük lehetséges módozatait.

Csak kellően megbizható hardverre és alapszoftverre /2.2/ szabad ABKR-et felépiteni.

Az ABKR szerkezetét ugy kell megválasztani, hogy az alkalmas legyen a megbizhatósági követelmények kielégitésére (pl. program és adat redundancia) /1./

Itt kell megemliteni, a teljesség kedvéért, hogy a megbizhatóságnak része a programok helyessége is.

A továbbiakban kizárólag az ABKR szoftverének sajátos megbizhatósági igényeiről lesz szó.

A megbizhatósági követelmények nagy mértékben függenek attól, hogy egyidejüleg egy vagy több felhasználó dolgozik-e az adatbázison, és hogy kötegelt vagy tranzakciós üzemmódban történik-e a feldolgozás.

Különös figyelmet érdemel a megbizhatósági szempontból legkényesebb többfelhasználós tranzakciós üzem, amely egyre inkább előre-tör az ABKR alkalmazásában.

Az ABKR legfontosabb megbizhatósági kritériuma az adatok épségének fenntartása.

Az adatok épségével kapcsolatban a következő meghatározásokra lesz szükség:

Az adatok "helyes" állapota azt jelenti, hogy az adatbázis a legfrissebb adatokat tartalmazza, és törölt adatok már nincsenek benne.

Ennél szükebb fogalom a "konzisztens" állapot, ami eltür az ép állapothoz képest bizonyos adatveszteséget, de megköveteli a felhasználói (!) konzisztencia feltételek teljesülését /3./.

Az adatok még helyes felhasználói programok futtatása esetén is megsérülhetnek, ha párhuzamosan futó programok azonos adat egyedeken irási ütközésbe kerülnek, illetve ha valamilyen hiba következtében a program vagy az ABKR müködése félbeszakad.

Az ABKR-nek tehát kezelnie kell a párhuzamos programfuttatásból származó ütközéseket, és képesnek kell lennie arra is, hogy a félbeszakadt program vagy üzem az adatok épségének kézbentartásával ujraindítható legyen.

A párhuzamosság és az ujraindithatóság problémaköre tehát az adatok ép - ségének megóvásán keresztül szorosan kapcsolódik egymáshoz, és igy tervezésüket is össze kell hangolni.

A továbbiakban szeretném röviden bemutatni a két problémakört, és algoritmus választást javasolni kisszámitógépes esetre.

2. Párhuzamos programfuttatás

Ha egy felhasználói program egyedül és hibátlanul lefut, akkor az adatbázist konzisztens állapotból konzisztens állapotba viszi át.

Ha azonban nem egyedül fut, összeütközésbe kerülhet más programokkal, amelyek ugyanazon adat objektumot érik el, és legalább egyikük irni kiván.

Ilyenkor a müveletek között tisztázatlan versenyhelyzet alakulhat ki, ami a következő konzisztencia sérülésekre vezethet:

Irás- olvasás ütközéskor nem lehet tudni, hogy az olvasó a módositás előtti, vagy utáni értéket kapta-e meg.

Irás-irási ütközéskor pedig csak az utolsó módositás jut érvényre, a korábbiak elvesznek.

Szükség van tehát valamilyen ütemezőre, amely gondoskodik arról, hogy a programok párhuzamosan, átlapoltan futva se sérthessék meg a konzisztenciát, azaz a soros futással egyenértékü eredményt adjanak.

Az ütemezésnek két alapvető módszere van: a zár /4./ és az időbélyeg /5./ kezelés, de javasoltak vegyes megoldásokat is /6./.

A zár módszer lényege a következő:

A programnak irási, illetve olvasás előtt az elérni kivánt adat objektumra irási, illetve olvasási zárat kell kérnie. Az ütemező a zárat csak akkor adja meg, ha az nem ütközik a már megadott zárakkal, ellenkező esetben a program az illető adatobjektumhoz rendelt várakozási sorba kerül.

Az ütemező csak akkor müködik helyesen, ha a program az első zár felszabaditási müvelet után nem kér több uj zárat (kétfázisu zár) /4./.

A zár módszernek az a legnagyobb előnye, hogy a zár szorosan kapcsolódik az általa védett adat objektumhoz.

Ez lehetővé teszi az ütközések pontos felfedését, és ezen keresztül a lehető legmagasabb foku párhuzamosság elérését.

Hátránya a zármódszernek a kölcsönös kizárás lehetősége, azaz előfordulhat, hogy programok egymás által lefoglalt adat objektumokra kérnek zárat.

Az időbélyeg módszer lényege abban áll, hogy az ütemező a programokat indulásukkor szigoruan monoton növekvő számozással látja el - ezt a számozást az órához szokás kötni -, és az ütköző müveleteket a számozás sorrendjében engedi érvényre jutni /6./.

Az időbélyeg módszernél kölcsönös kizárás nem fordulhat elő.

Komoly problémát jelent viszont, az ütközések felfedése. Az erre a célra publikált algoritmusok (pl.SDD-1) igen bonyolultak, az alkalmazási tapasztalat még kevés.

Az időbélyegek tárolása és kezelése sulyos többletterhet ró a rendszerre.

Kisszámitógépes rendszereknél ezért véleményem szerint a zármódszert érdemes alkalmazni.

2.1. Zár mechanizmus

A párhuzamos program futtatás tehát a kétfázisu zár algoritmusra épül.

Az algoritmus könnyen kiterjeszthető elosztott ABKR-re is, de ilyenkor a többszörös előfordulásu adatokra bizonyos kiegészitéseket kell bevezetni /6./.

Az algoritmus kényes pontja a zárak hatóköre, azaz az általuk védett adat objektum nagysága.

Kis hatókör a zárak tulszaporodására vezethet, nagy hatókör pedig csökkentheti a párhuzamosságot.

Érdemesnek látszik többféle hatókörü zár alkalmazása (rekord, állomány, stb.), igy a felhasználói programok tervezésekor a lehető legkedvezőbb válaszidőt lehet elérni.

A különféle hatókörü zárak hierarchiája ugyancsak kihasználható /7./.

A zárak alkalmasak arra is, hogy sérült adatbázis részt a teljes helyreállitásig védjenek, lehetővé téve ezáltal más, ép részek elérését, a csökkentett értékü szolgáltatás folyamatosságát.

2.2. Kölcsönös kizárás felfedése

Az ABKR-re az a jellemző, hogy a programok által igényelt adat objektumok meghatározása nem a programok indulásakor, hanem futás közben történik /8./.

Igy nincs mód a kölcsönös kizárás megelőzésére, vagy elkerülő stratégiák alkalmazására.

Marad tehát a felfedés és feldolgozás.

Kölcsönös kizárás felfedésére egyszerü, elosztott esetre is kiterjeszthető algoritmus létezik.

Fel kell venni a rendszer ugynevezett állapot gráfját. A gráf csomópontjai a programok és az adat objektumokat képviselő erőforrások. Ha egy erőforrás egy programé, az erőforrás csomópontból élet vezetünk a program csomópontjába. Ha egy program egy erőforrásra vár, élet vezetünk a program csomópontjából az erőforrásba.

Könnyen belátható, hogy ha az irányitott gráfban gyürüt találunk, akkor ez kölcsönös kizárást jelent.

A kölcsönös kizárást ugy oldhatjuk fel, hogy a gyürüben levő valamelyik programot visszagörgetjük, azaz módositásait töröljük.(ld.3.2.), és

lefoglalt adat objektumait felszabaditjuk.

A visszagörgetendő áldozat kijelölésére sokféle módszer született: a legkisebb költséggel visszagörgethető, a legfiatalabb (időbélyeg módszer), a legalacsonyabb prioritásu program, stb.

Az áldozat ujrainditása lehet az ABKR vagy a felhasználó (operátor) feladata.

3. Ujrainditás

Ha valamely program vagy az egész rendszer futása hiba következtében (feszültség kimaradás, hardver vagy szoftver hiba, kölcsönös kizárás, stb.) félbeszakad, a program vagy a teljes rendszer futását ugy kell ujrainditani, hogy az adatok épsége kézben tartható legyen.

Az adatok épsége a legtöbb esetben elsőbbséget élvez a szolgáltatás folyamatosságával szemben, azaz el kell fogadni az adatok helyreállitásával kapcsolatos többletterheket. Kivánatos, hogy a helyreállitás a lehető legkevesebb adatveszteséggel járjon.

Nagybiztonságu ABKR-től elvárható, hogy a helyreállitás konzisztens állapotot eredményezzen.

Egy program az adatbázis konzisztenciáját definició szerint megőrzi.

Ez azt jelenti, hogy a programnak vagy valamennyi módositása érvényes, vagy egyetlen egy sem (kétfázisu módositás). A félbeszakadt programnak tehát valamennyi módositását törölni kell.

Előfordulhat azonban, hogy a program által módositott adat egyedet közben valamely másprogram is kiolvasta, vagy akár módositotta. Ennek az lesz a következménye, hogy az utóbbi programot is törölni kell, és a dominó hatás miatt egyre tetemesebb adat veszteség léphet fel.

Az ütemezőnek tehát az ujrainditási követelményeket is figyelembe kell venni /7./!

A dominó hatást kiküszöbölhetjük ugy, hogy előirjuk, hogy ha egy program módositott egy adat objektumot, akkor ezt az adatot más program a módositott program befejeztéig ne módosithassa.

A zár módszernél ez azt jelenti, hogy irási zár csak a program végén szabadul fel.

Igy is előfordulhat azonban, hogy valamely program olyan adatot olvas, amelyet később más program módosit.

Ennek viszont az a következménye, hogy az ujrainditáshoz a rendszer futása közben rendszeresen ellenőrző pontokat kell felvenni.

Ujrainditáskor vissza kell gurgetni mindazokat a programokat, amelyek a hibe pillanatában, az utolsó ellenőrző pont óta, vagy éppen az ellenőrző pont felvételekor futottak, és ujrafuttatni mindazokat, amelyek

előtt befejeződtek.

Mindez feleslegessé válik, ha azt is megköveteljük, hogy ha egy program beolvasott valamely adat objektumot, akkor azt más program az olvasott program befejeztéig ne módosithassa.

A zár módszernél ez ugy biztositható, hogy olvasási zár sem szabadítható fel a program végéig.

Ezzel a programokat tökéletesen elszigeteltük egymástól, visszagörgetésük egymástól független, és ellenőrző pontok felvételére sincs szükség.

Az előbbiekben vázolt ujrainditási követelményeket az ismert ujrainditási módszerek közül a legteljesebben az adatbiztonsági naplózás elégiti ki /3./.

3.1. Adatbiztonsági naplózás

Az adatbiztonsági naplót a visszagörgetés, illetve az ujrainditás használja fel.

Tartalmaznia kell a program inditásokat és befejeződéseket, a zár müveleteket, és az adatbázis módositásokat.

A módositásokat a módositás előtti és a módositás utáni mágneslemez kép irja le.

Az előbbire a visszagörgetéshez, az utóbbira a lemez helyreállitáshoz van szükség.

Ha a programokat egymástól teljesen elszigeteltük, akkor a módositás előtti lemez képekre csak a program inditásától annak sikeres befejeztéig, vagy visszagörgetéséig van szükség. Tárolásukra legmegfelelőbb a mágneslemez (elérési idő, adatmennyiség).

A megbizható ABKR-nek fel kell készülnie fizikailag sérült lemez helyreállitására is.

Erre két módszer kinálkozik. Az első az, hogy az uj lemezre korábbi, konzisztens állapotu (általában mágnesszalagos) mentést töltünk, majd felvisszük rá az azóta végzett módositásokat a módositás utáni lemezképek alapján.

A módositás utáni lemezképekre tehát két adatbázis mentés között akármikor szükség lehet. Általában nagytömegü adatról lévén szó, tárolása mágnesszalagon történhet.

A másik módszer a lemez egységek és az adatok fizikai megkettőzése, és a módositások folyamatos vezetése mindkét adat példányon. Ekkor a helyreállitás lemez másolatra redukálódik, a módositás utáni lemez képek naplózására nincs szükség.

3.2. Visszagörgetés, hideg és meleg ujrainditás

Visszagörgetést kezdeményezhet maga a program, ha például a gyüjtött adatokban inkonzisztenciát észlel, de kérhet visszagörgetést az ütemező is, pl. kölcsönös kizárás feloldására. Visszagörgetésre van szükség ujrainditáskor is, sőt a programok teljes elszigetelése a "meleg" ujrainditást a hiba pillanatában futó programok visszagörgetésére egyszerüsiti.

A visszagörgetés maga ugy történik, hogy a programhoz tartozó módositás előtti lemez képeket forditott időrendben vissza kell irni az adatbázisba.

"Hideg" ujrainditásra csak akkor van szükség, ha fizikai adathordozó sérülés történt. Ilyenkor helyre kell állitani a sérült lemezt, ezután pedig meleg ujrainditás következik.

4. Hivatkozások

/1./ B. Randell et al.:
Reliability Issues in Computing System Design
ACM Computing Service (CS) 10/2 1978. pp. 123-166.

/2./ ACM CS 8/4 1976. Special Issue
Reliable Software II. Fault-Tolerant Software

/3./ J.S.M. Verhofstad:
Recovary Techniques for Database Systems
ACM CS 10/2 1978. pp. 167-196.

/4./ K.P.Eswaran et al.:
The Notions of Consistency and Predicate Locks in a Database System
Communications of the ACM (CACM) 19/11 1976. pp. 624-633.

/5./ L. Lamport:
Time, Clocks and the Ordering of Events in a Distributed System
CACM 21/7 1978. pp. 558-565

/6./ P.A.Bernstein, N.Goodman:
Concurrency Control in Distributed Database Systems
ACM CS 13/2 1981. pp. 185-222.

/7./ J.N.Gray et al.:
Granularity of Locks and Degrees of Consistency in a Shared Data Base
in G.M.Nijssen (ed.)
Modelling in Data Base Management Systems
North Holland Publishing Co. 1976. pp. 365-394.

/8./ S.S.Isloor, T.A.Marsland:
The Deadlock Problem: An Overview
Computer 13/9 1980. pp. 58-72.

ifj. Krekó Béla
Országos Tervhivatal
Számítástechnikai Központja

Adatmodellezés a népgazdasági tervezésben

Adatmodellen az adatok valamilyen absztrakt képét, az adatok egy meghatározott szemléletmódját értjük.

Közismert, hogy ugyanazon adatokra az aktuális céltól, a modellező izlésétől, a választott gondolati kerettől stb. függően igen különböző modellek adhatók, melyeket többféleképpen szoktak tipizálni. Ilyen modelltipusokat jelentenek pl. az ANSI/SPARC struktura külső, belső és fogalmi sémái /mely felosztásban a modellezési cél tükröződik/,vagy pl. a relációs, hálós, entity-relationship stb. modellek /itt a felosztás inkább formai szempontból történik /. Az is közismert, /éppen az ANSI/SPARC jelentés nyomán/, hogy egy alkalmazási környezetben általában több adatmodellre van szükség, mely tény magával hozza a különböző adatmodellek közötti megfeleltetések problémakörét.

Adatmodellezésen a továbbiakban az egy alkalmazási környezethez tartozó adatmodellek és megfeleltetéseik megadásának, kezelésének problémakörét értjük.

Nyilvánvaló, hogy a kérdéskört erőteljesen befolyásolja az adatmodellek és kapcsolataik leírásának formája:milyen metafogalmakkal dolgozunk, milyen matematikai modellt képzelünk mögéjük. Ebben természetesen tükröződniük kell az alkalmazási környezet sajátosságainak.

Az irodalom [2,4,6,9,10] szinte kizárólag olyan alkalmazási környezetekkel foglalkozik, ahol a fogalmi séma az entitás, attributum, reláció metafogalmakra támaszkodik, a külső, ill. belső sémák pedig viszonylag egyszerüen feleltethetők meg a fogalmi sémáknak /pl. a tárolás általában valami rekordszerü egységekben történik, egy entitás-tipusnak rendszerint egy rekordtipus felel meg, az attributumoknak mezők, stb./

Vannak azonban olyan alkalmazási környezetek, ahol ezek a konstrukciók nem, vagy csak nehézkesen alkalmazhatók. Előadásunkban azt szeretnénk bemutatni, hogy makroökonomiai környezetben más tipusu /más fogalmakra épülő/ adatmodellek a "természetesek"; ezek a modellek a külső, belső és fogalmi sémák leírására egyaránt alkalmasak; kézenfekvő módszereket kínálnak a sémák közötti megfeleltetések leírására; és elég egyszerüek ahhoz, hogy számítógéppel kezelhetők legyenek.

Fogalmi sémák

Ha erősen összevont, feldolgozott adatokkal dolgozunk, akkor általában igen nehéz az entitástipusok feltárása. Ezek az adatok a gazdasági élet heterogenitását elfedő, erős absztrakciók eredményei.

Ha a kézenfekvően felismerhető entitástipusokra akarunk támaszkodni, akkor áttekinthetetlenül sok entitástipusunk lesz /melyek felmérése ráadásul speciális szakértelmet igénylő, igen nagy feladat/. Ahhoz, hogy a modell áttekinthető legyen, első lépésben sokkal absztraktabb entitástipusokkal kell dolgoznunk, és ezek fokozatos konkretizálásával kell eljutni a "köznapi" entitástipusokhoz. Az adekvát általánositás azonban igen nehéz feladat, melyhez sajnos a közgazdasági elmélet sem ad sok segitséget.

A gyakorlatban az a megoldás alakult ki, hogy csak néhány kézenfekvő entitástipust definiálnak/ezeket szokás nomenklaturákba foglalni/, és a rajtuk /illetve belőlük felépithető összetett entitásokon/ értelmezett attributumok értékeinek tekintik az adatokat. Ennek általában az attributumok elég nagy száma és a köztük fennálló bonyolult összefüggések a következménye /ezeket szokták mutatókatalógusokba foglalni /. A vállalati mérlegbeszámolókban pl. egy vállalat egy adott évi tevékenységét /összetett entitás/ többezer attributum jellemzi.

Az entitástipusok "elrejtését", "elhallgatását" megkönnyiti /illetve lehetővé teszi/ az aggregálás, melynek során entitások /relációik és attributumaik segitségével képzett/ halmazaira vonatkozó mértékek kiszámitásáról van szó.

A gyakorlatban az aggregált adatok definicióját általában nem adják meg explicit módon, hanem valamilyen rövidebb névvel helyettesitik /pl.: x vállalat létszáma t időpontban/. Itt nincs explici hivatkozás a "dolgozó" entitástipusra, ill. a "z dolgozó munkahelye x vállalat a t időpontban" relációra. A "létszám" igy önálló attributumként, és nem egy halmaz számosságaként jelenik meg.

Az eddigiekben azt bizonygattuk, hogy makroökonomiai környezetben általában igen nehéz egy, a vállalati alkalmazásoknál megkivánt minőségü fogalmi séma /"a vállalat egy modellje"/ kidolgozása.

Más oldalról az is igaz, hogy az adatmodellezés eszköztára sincs kellőképpen felkészitve aggregált, származtatott adatok leirására. Ez a kérdés általában csak a lekérdezéseknél kerül szóba. Az igy nyert adatok azonban már nem tartoznak a modellezendő adatbázishoz. Megoldási kisérletek találhatók [1,3] -ben.

Ilyen körülmények között célszerü abból kiindulni, hogyan látják adataikat a felhasználók /közgazdászok, tervezők, statisztikusok/. Fogalmi szinten általában a következőképpen gondolkoznak:

- egy adat nem más, mint egy "tartalom=érték" kifejezés, ahol az érték egyetlen /valós/ szám.

- az adatok tartalma általában több részből tevődik össze. Például:
  - mutató: az entitások, ill. kapcsolataik mely jellemzőjének értékéről van szó /milyen tipusu az adat/
  - a szóbanforgó entitást, ill. kapcsolatot leiró ismérvek /attributumok/,ill. ezek értékei
  - egyéb, az egyértelmü meghatározáshoz szükséges információk /pl. a vonatkozási időpont vagy időszak,az adat forrása, mértékegysége, stb./

- adatok bármely halmaza csak akkor nyujthat reális /konzisztens/ képet a valóságról, ha az értékek között fennállnak bizonyos /csak a tartalomtól függő/ konzisztencia feltételek /aritmetikai egyenlőségek és egyenlőtlenségek, röviden: egyenletek/.

Egy fogalmi séma definiciójából tehát annak kell egyértelmüen kiderülnie, hogy

- mi a megengedett adattartalmak halmaza
- a megengedett adattartalmak /pontosabban: a hozzájuk tartozó értékek/ között milyen konzisztencia-feltételeknek kell fennállniuk.

Ezen fogalmakat a következőképpen tehetjük operativvá /számitógéppel kezelhetővé:

- az adattartalmat azonositjuk az adat megnevezésével /mely a fentiek szerint több részből álló, összetett objektum/
- a konzisztencia-feltételeket adatmegnevezésekből felépitett aritmetikai egyenletekként reprezentáljuk /ezek szintén összetett objektumok/.

Ez az adatmodell tehát tulajdonképpen nem más, mint egy olyan egyenletrendszer /egyenlőségek és egyenlőtlenségek halmaza/, melynek változói az adatmegnevezések. Ezért a továbbiakban egyenletrendszer /ER/ modellnek fogjuk nevezni.

Egy ER modell definiciója tehát nem más, mint egy megnevezéshalmaz és egy egyenlethalmaz definiciója.

A halmazok definiálása általában nem közvetlenül /elemeik explicit felsorolásával/ történik. Ennek két oka van:

- a halmazok elemszáma százezres, milliós nagyságrendben mozoghat: az elemenkénti felsorolás praktikusan lehetetlen
- általában lehetőség van arra, hogy nagyságrendekkel kevesebb információból állitsuk elő őket megfelelő müveletek /generálási szabályok/ felhasználásával.

A kiinduló információkat a megnevezések ill. összefüggések "épitőköveiként" felhasználható egyszerü és összetett objektumok halmazaiként lehet megadni.

Gyakran előfordul, hogy ezen halmazok közül bizonyosakat nem ismerünk előre /vagy tudjuk, hogy pillanatnyilag adott terjedelmük később változni fog/, de tudjuk, hogy bármely időpontban ki kell hogy elégitsenek bizonyos integritási feltételeket.

Egy ER modell definiciója tehát a következőkből áll:

- absztrakt modell: - generálási szabályok /megnevezésekre és egyenletekre/
  - integritási feltételek
- konkretizálás: az ismert halmazok megadása elemfelsorolással

Ez a leirási mód lényegében nem más, mint a matematikai közgazdasági modellek leirási módjának alkalmazása az adatmodellezésre. Ez persze nem véletlen: modelljeink adatokról "beszélnek",ebben az értelemben tehát adatmodellnek is tekinthetők.

Nézzük meg példaként az input-output elemzés egyszerüsitett modelljét /a szokásos leirási módban/:

Változók:

- x_i: az i szektor btto outputja / $i \in I$, I a szektorok halmaza/
- y_i^f : az i szektor netto outputja az f felhasználási célra /$f \in F$, F a felhasználási célok halmaza, $I \cap F = \emptyset$/
- x_{ij}: a j szektor inputja i szektor outputjából /$i,j \in I$/

Feltételek:

$$x_i = \sum_{j \in I} x_{ij} + \sum_{f \in F} y_i^f \quad ; \quad i \in I$$

$$x_i, y_i^f, x_{ij} \geq 0 \quad ; \quad i,j \in I, \; f \in F$$

Indexhalmazok:

- $I = \{ba, ko, \ldots\}$
- $F = \{fo, fh, \ldots\}$

Az itteni változóazonositók /x_{ba}, y_{ba}^{fo} . . . / a mi terminológiánkban adatmegnevezések, az egyes feltételek pedig egyenletek. A definició "változók" része /a szöveges magyarázattól eltekintve/ nem más, mint a megnevezésekre vonatkozó generálási szabályok együttese: ez alól csak az "$I \cap F = \emptyset$" kitétel a kivétel, mely viszont integrálási feltétel.
A "Feltételek" rész az összefüggésekre vonatkozó generálási szabályok halmaza. A két rész együttesen definiálja az absztrakt modellt.

Az "Indexhalmazok" rész a konkretizálás.

Eddigi definiciónk semmi megszoritást nem tartalmaz az adatmegnevezésekre vonatkozóan, a gyakorlat számára tul általános. Szerencsére vannak olyan egyszerü speciális esetek, melyeket viszonylag széleskörüen alkalmaznak:

- az adatmegnevezések maximált hosszuságu szimbólumsorozatok. Ilyen megnevezésekhez jutunk a szokásos indexes matematikai jelölésmód linearizálásával, vagy hierarchikus elrendezések felhasználásával. Ezt a modellt "klaszszikus ER" modellnek /KER/ nevezzük
- az adatmegnevezések azonos hosszuságu szimbólumsorozatok, tehát az adatok egy /szimbólumokkal indexelt/ tömb elemeinek tekinthetők. Ezt a modellt tömb modellnek /T/ nevezhetjük. A tömb dimenzióit ismérveknek, az indexek értékeit ismérvértéknek szokás nevezni. Általában nem minden indexkombináció értelmes, ezért tulajdonképpen egy "cakkos tömbről" /jagged array/ van szó.

Könnyen belátható, hogy bármely KER modellhez megadható /"természetes módon"/ egy vele ekvivalens T modell, tehát a T modellek nem speciálisabbak a KER modelleknél.

Külső sémák

A felhasználók adataikat általában numerikus táblázatokba, táblázatsorozatokba, táblázatsorozatok sorozataiba, többdimenziós tömbökbe, hipertömbökbe és ehhez hasonló strukturákba elrendezve "látják". A tömbökre, ill. az egyes dimenzióértékekre maguk választotta névvel hivatkoznak. Az adatok közötti összefüggések tömbök,"tömbrészek" közötti aritmetikai egyenletekként jelennek meg. Ezek az adatstrukturák /a dimenziók egymásutániságát rögzitve/ hierarchikus strukturáknak is tekinthetők, a felhasználó adta nevek pedig adatcsoportok egy nagyobb adatcsoporton belüli "relativ nevének". Minden adat megnevezhető egy, a hierarchián belüli helyét kijelölő névsorozattal. Ha például adatainkat táblázatokból álló állományoknak képzeljük el, akkor egy adatot egy ⟨állománynév, táblázatnév, sornév, oszlopnév⟩ szimbólumsorozattal tudunk megnevezni. A KER modell tehát alkalmas a külső sémák leirására. A tömbökbe rendezhetőség olyan "szabályosságok" formájában jelenik meg, melyeket generálási szabályok formájában lehet kihasználni. /Ahhoz hasonló dolgoról van szó, mint amikor tömbök azonosságát vagy a velük végzett müveleteket definiáljuk a tömbelemek felhasználásával./.

Belső sémák

A belső séma fogalmát a szokásosnál absztraktabb értelemben használjuk. Az adattárolásból csak annyi érdekel bennünket, hogy milyen információk megadásával érhető el egy tárolt numerikus érték /mi az adat "absztrakt cime"/, illetve, hogy

milyen aritmetikai összefüggéseknek kell fennállniuk az egyes "címeken" tárolt adatok között. A tárolás adatstrukturái - más környezethez hasonlóan - tervezési, makroökonomiai környezetekben is meglehetősen hasonlítanak a külső sémák adatstrukturáihoz. Esetünkben tehát általában hierarchikus, ill. "tömbszerü" strukturákról van szó. Ilyen tárolási strukturákkal dolgoznak a szocialista országok tervhivatalaiban kifejlesztett adatkezelő rendszerek, a francia tervezést szolgáló LEDA és ARGOS programcsomagok, de még a vállalati tervezés számára készült un. "pénzügyi tervezési rendszerek" /financial planning systems/ is.

A KER modell alkalmazhatóságáról tehát lényegében ugyanazt lehet elmondani, mint a külső sémák esetén.

A sémák közötti megfeleltetések

A sémák megfeleltetése a népgazdasági tervezés körülményei között jóval általánosabb dolog, mint a külső, ill. belső sémák leképzése a fogalmi sémára. Ennek az a fő oka, hogy praktikusan lehetetlen egyetlen, mindent átfogó fogalmi séma kidolgozása. A külső séma - fogalmi séma - belső séma viszonyok helyébe tehát a különböző sémák egymásban való definiálhatóságának viszonya lép. Az S_1 séma akkor definiálható S_2-ben, ha S_1 minden adatmegnevezéséhez hozzárendelhető egy S_2-beli aritmetikai kifejezés, oly módon, hogy S_1 egyenletei következményei S_2 egyenleteinek. Magát a megfeleltetést nevezzük definiciónak. A definicó speciális egyenletrendszernek tekinthető, megadására tehát ugyanazok a technikák szolgálhatnak, mint az ER modell egyenletrendszerének megadására.

A sémák és megfeleltetéseik leirása, kezelése

Ezeket a kérdéseket a terjedelem adta korlátok között csak futólag érinthetjük. A T modellek és kapcsolataik leirására kézenfekvő a matematikai logika, ezen belül a logikai programozás, konkrétan a PROLOG formalizmusa [7].

A kezelési funkciók egy része /konzisztenciavizsgálat, lekérdezések/ magán a PROLOG-on /mint következtető rendszeren/ megoldható, más részük PROLOG leirások azonos átalakitását igényli . Ez utóbbiak közül egyenlőre csak a tárolt adatok visszakeresésének /pontosabban: "címük" meghatározásának/ problémája terén értünk el eredményeket [6].

Az ER modell esetén igéretesnek látszik egy nyelvtani megközelités is [5].

Irodalom

[1] Asztalos D., Koltai T., Krekó B.: AMETIST: a meta-information system. Proc.of Int. Sem. on DBMS's, Schwerin, GDR, 1981.

[2] Chen, P.P./szerk./: Entity-Relationship Approach to System Analysis and Design. North-Holland, 1980.

[3] C.S.Dos Santos, E.J. Neuhold, A.L. Furtado: A Data Type Approach to the Entity-Relationship Model.In [2] .

[4] Frost, R.A.: SCHEMAL: Yet Another Conceptual Schema Definition Language. The Computer Journal, 3,228-233/1983/

[5] Gerevich L.: A Keretmodell formalizmusának egy megvalósitása. OTSzK munkaanyag, 1983.

[6] Halassy B.: Adatmodellezés, adatbázis-tervezés. SZÁMOK, 1980.

[7] Kiss Z., Krekó B., Őrszigety Gné.: A pénzügyi adattár tartalmi leirása. OTSzK munkaanyag, 1983.

[8] Krekó B.: Számitógépes adattárak tartalmi nyilvántartása. OTSzK munkaanyag, 1982.

[9] Nijssen, G.M./szerk./: Modelling in Data Base Management Systems. North-Holland, 1976.

[10] Nijssen, G.M./szerk./: Architecture and Models in Data Base Management Systems. North-Holland, 1977.

VIDEOTON adatbáziskezelők számítógéphálózatban

Békéssy Péter - Szalay Imre
SZÁMALK Kisszámítógép Főosztály[x]

Összefoglaló

A DMS60 és DMS600 adatbáziskezelő rendszerek VIDEOTON számítógéphálózathoz való kapcsolásának megvalósitásáról számolunk be az előadásban. Ez egy munkahelyről több távoli adatbázis elérését teszi lehetővé időben megosztva.

Szólunk az illesztések koncepciójáról, munkamódszerünkről és az alkalmazás eddigi tapasztalatairól is.

x A munka a VIDEOTON megrendelésére készült, a fejlesztés eredménye a VIDEOTON tulajdona.

1. Előzmények

A folyóirat publikációk gyakorisága alapján elmondható, az osztott adatbáziskezelés napjaink aktuális témája. Eddig leginkább csak elméleti részeredmények születtek, s a nálunk fejlettebb számítástechnikával rendelkező országokban is ritka a ténylegesen üzemelő, "igazi" osztott adatbázis.

Hazánkban az elosztott adatfeldolgozást alacsonyabb intelligenciával megoldó rendszerek lehetnek csak napirenden. Gyakorlati tapasztalatok szerezhetők földrajzilag osztott adatbázisokkal /távoli adatbázis elérések/ a számítógéphálózatok megbizhatósága, a végfelhasználók számítástechnikai szervezettsége terén. Erre a bázisra épitve lépcsőzetes fejlődés lehetséges: filetranszfer megvalósitása, majd a konzisztencia problémák megoldása után tranzakcióinditás az aktuális adatbázison kivüli adatbázisok lekérdezésére, módositására. Ezek a résztapasztalatok, az elméleti eredmények beérésével, a külvöldi osztott adatbázisok megismerésével vezethetnek el idővel az igazi osztott adatbáziskezelés problémájának itthoni megoldásához.

E rövid okfejtés mindjárt érthetőbbé válik az alábbi project történetének ismertetésével.

A Videoton Számitástechnikai Gyár és a Számitógépalkalmazási Kutató Intézet /jogutódja 1982. óta: Számitástechnika - Alkalmazási Vállalat/ 1978-ban elhatározta, hogy a Videoton által gyártott hardver berendezésekre épitve közösen kifejleszt egy komplex adatbázis /adatátviteli rendszert/, mely alkalmas földrajzilag elosztott többtelephelyes nagyvállalatok termelésirányitási rendszereinek támogatására.

A fejlesztési koncepció abból indult ki, hogy a termelésirányitáshoz szükséges adatok a funkcionális szervezeti egységenként rendelkezésre állnak. A termelésirányitáshoz szükséges döntésekhez azonban sokszor nem elegendő a lokális részadatbázis adatainak ismerete, ezért biztositani kell valamennyi többi részadatbázis elérését. Ezt a távoli elérést kommunikációs alrendszer megvalósitásával kivánták megoldani. Mindez találkozott a Videoton számitógépes termelésirányitásának rekonstrukciós törekvéseivel, s igy lehetővé vált, hogy a létrehozott rendszert a Videoton referencia alkalmazásként üzemeltesse.

A fejlesztés VNS /Videoton Network Service/ néven futott a következő feladatmegosztásban:

- A Videoton vállalta a felelősséget a hardver komponensekért, továbbá a felhasználói adatbázisok és a hozzájuk tartozó tranzakciós rendszerek kidolgozásáért.

- A SZÁMKI /SZÁMALK/ Távadatfeldolgozási Főosztályán dolgozó team alkotta meg a kommunikációs alrendszert, az NSS-t /Network Service Support/.

- Jelen beszámoló szerzőinek feladata volt a francia licenc alapján a VIDEOTON által forgalmazott DMS60, majd DMS600 adatbáziskezelő rendszereknek az NSS-hez való adaptálása.

2. Az illesztés

Az NSS X-25 elvek szerinti rétegekből felépülő csomagkapcsolt hálózati szoftver, amely az un. csomóponti /node/ számitógépeken fut. Az alkalmazások futtatására az un. host gépek szolgálnak. A DMS60/600 CODASYL tipusu hálós strukturáju tranzakcióorientált kisgépes adatbáziskezelőrendszerek szerepelnek a hálózati alkalmazásokként. Ezek a szoftver termékek a hozzájuk tartozó felhasználói kézikönyvekből, külső specifikációkból ismerhetők meg alaposabban.

A már üzemelő hálózati konfiguráció a géptipusok feltüntetésével az 1. ábrán látható.

A kommunikációs alrendszer az adatbáziskezelő számára meghatározza a terminálokkal való forgalmazás szabályait, és a forgalmazott csomagok formáját, az un. protokollt.

Az adatbáziskezelő szerkezetének vizsgálata megmutatta, hogy egy adatkezelő és egy adatgyüjtő részrendszerre osztható. Előbbi valósitja meg az adatok változtatását /update/, elhelyezését, utóbbi az interaktiv tranzakciók felügyeletét, a több terminál egyidejü kiszolgálását. A földrajzilag osztott adatkezelés megszoritás épp azt jelenti, hogy az adatbáziskezelőnek csak az adatgyüjtő része változik ugy, hogy alkalmazzá válik a hálózatban lévő terminálokon való adatgyüjtésre, tranzakció futtatásra is. Igazi osztott adatbázis esetén az adatkezelő alrendszer sem maradhatna változatlan, hiszen a host gépeken elhelyezkedő adatbázisok között ekkor adatkapcsolatok és összefüggések is lennének.

Esetünkben egymástól független adatbázisok müködnek, viszont a hálózatba kapcsolt tetszőleges terminálról e független adatbázisok bármelyike, de egyidejüleg csak egyike, elérhető.

Az adatbáziskezelő szerkezete ugy változik az illesztés következtében, hogy az adatgyüjtő alrendszer helyére egy hálózatkezelő alrendszer kerül.

A hálózatkezelő alrendszer funkciói:

- kapcsolatfelvétel a hálózattal és a terminálokkal
- csomagforgalmazás
- csomagtartalmak összeállitása: konverzió a protokoll szerinti formátumra és vissza.
  /Ez a lényegi változtatás a DMS60/600 strukturájában/

- kapcsolatbontás egy terminállal, a hálózattal és uj terminál fogadására való felkészülés

Alapelv volt az illesztés megvalósitásakor, hogy a felhasználói interfész minimálisan módosuljon. A valóban nem jelentős változások oka, hogy a terminálokat a hálózatban egy általánosabb célra készült terminálkezelő vezérli.

3. Néhány szó a munkamódszerről

Munkánk végeredménye a távoli terminálokon lefutó adatkezelő tranzakciók formájában csak a hálózat legtulsó végén levő szoftver réteg által jelenik meg ténylegesen. Mig azonban eddig eljut, sok párhuzamosan fejlesztett szoftver rétegen kell a csomagoknak keresztül jutnia. Ettől ugy függetlenitettük magunkat, hogy mindkét alkalommal közvetlenül az adatbáziskezelőre épitettünk egy szimulátor réteget, amely a távoli terminálokra érkező csomagokat mind megjelenitette az operátori konzolon, illetve mód volt a terminálokról érkező csomagok összeállitására is az operátori konzolról.

Jól müködő részfunkciók egymás utáni sorozatával jutottunk el a kész rendszerhez. Ha egy részcélt beprogramoztunk, akkor csak ennyi változtatást végeztünk a müködő rendszeren, a továbbfutó összes ágakat nyomkövetési pontokkal lezártuk. Ha az uj részfunkciót sikerült belőnünk, a nyomkövetési pont jelölte ki a következő részcélt. A nyomkövetési pontok segitették a rendszer módositási helyei teljes halmazának megtalálását.

Irodalom

1. I. Földvári, P. Rajki, L. Simonfai, R. Szentes:
   COMNET - 2000
   5th Conference on the Theory of Operating Systems,
   Visegrád 1979.

2. P. Rajki, R. Szentes, Z. Ujvári:
   An Overview on the VIDEOTON-SZÁMKI Networking
   Project 3rd Hungarian Computer Science Conference,
   Budapest, 1981.

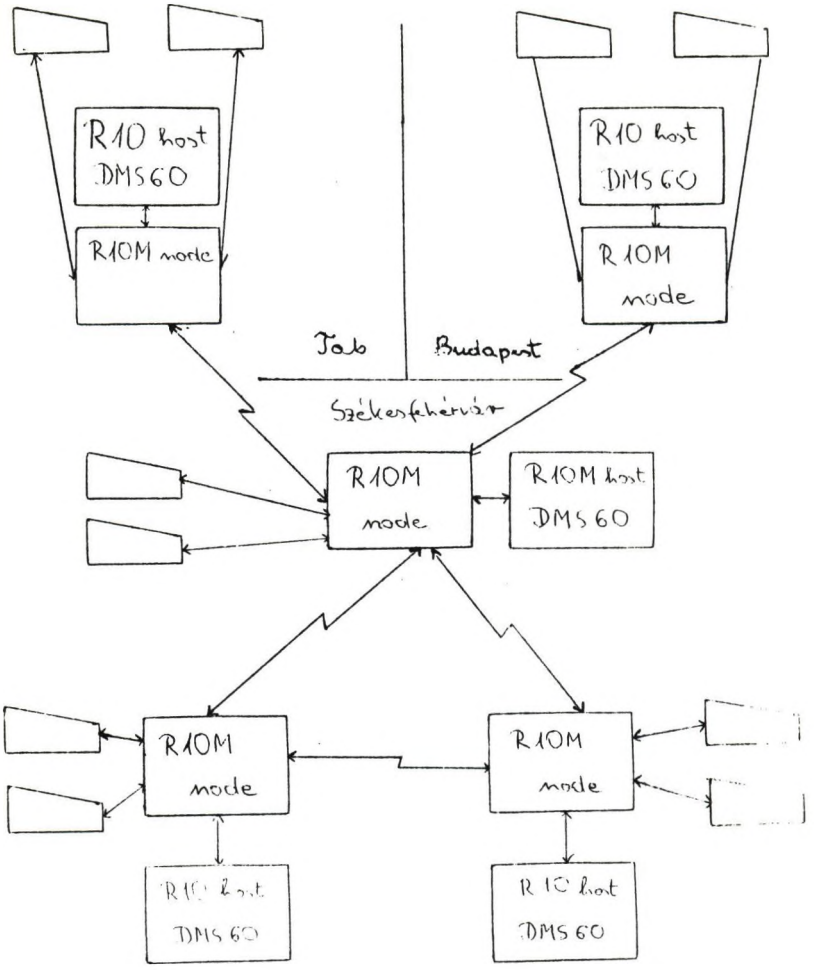
3. P. Rajki, R. Szentes, Z. Ujvári:
   VIDEOTON Network System General Overview
   COMNET '81 Conference Working papers /2-21 p./

4. Rajki P., Szentes R., Ujvári Z.:
   VIDEOTON kisszámitógépes hálózat általános ismertetése

5. Ungár János: Adatbázis-kezelő rendszerek a VIDEOTON
   számitógépein
   Információ-Elektronika 1981/6.

6. Baffia L. - Ungár J.: DMS 60 adatbáziskezelő rendszer
   Struktura, 1979/9. /153-168. o./

7. Békéssy P. - Szalay I.: DMS 60 az NSS 0 verziójában
   SZÁMKI-VIFI Bp. 1981.



A VHS üzemelő változata

1. ábra

Várnai Katalin
ELTE Számitóközpont

Dinamikus állományszerkezetek

Előadásunkban nagyméretű, dinamikusan változó adatállományok kezelési módszereiről kivánunk összehasonlító áttekintést adni. Az utóbbi néhány év szakirodalma - elsősorban az általunk is idézett szerzők tolléból - néhány, a klasszikus hashing módszerének tovább fejlesztésével kialakitott eljárást ismertet. Ezek az algoritmusok az egyes rekordok kulcs szerinti közvetlen elérését - az egyes szerzők által választott környezetben - jó tárolókihasználtság mellett biztositják. Szeretnénk azonban az ismert módszereket egységes szempontok szerint vizsgálni, azonos környezetben összehasonlitani. Ezért a következő alapfeltevésekkel élünk:

1. Az adatállományban előforduló rekordok számáról nincs előzetes információnk, és ez a szám az adatállomány élete során tág határok között változhat rekordok törlésével és beszurásával.

2. A rendelkezésre álló központi memória kicsi az adatállomány méretéhez képest és nem növelhető az adatállomány méretének növekedésével.

3. Minden rekordnak van egy egyértelmű azonositója, amely alapján a rekordot az állományból ki kell keresni.

Jól érzékelhető, hogy a "klasszikus" adatszervezési módszerek, pl. az indexszekvenciális, láncolt vagy akár a direkt file-ok ilyen feltételek között nem hatékonyak. A "hatékony" fogalom pontosabb használata érdekében definiáljuk a tárolókihasználás és elérési szám fogalmát egy adott módszerhez a következőképpen:

$T(n)$ = n rekord átlagos összhosszának és az n rekord tárolásához és a kulcs szerinti elérés megvalósításához szükséges tárolóhelynek hányadosa /kitöltési tényező/

$F(n)$ = egy rekord megkereséséhez átlagosan szükséges hozzáfordulások száma az n rekordot tartalmazó állományban

Egy adatszervezési módszer hatékonysága jól jellemezhető azzal, hogyan viselkedik $F(n)$ és $T(n)$ a rekordszám valamilyen tartományában. Ha pl. találnánk olyan módszert, amely a rekordok számától függetlenül teljesitené a

$$0 < C_1 < T(n) \qquad \text{és} \qquad F(n) < C_2$$

feltételeket, ezt valóban dinamikusan - azaz bármilyen méretü állományra egyformán jó hatékonysági mutatókkal - működőnek nevezhetnénk. Ilyen módszert nem sikerült találni, az ismertetésre kerülők csak a rekordszám valamilyen /a rekordok, kulcsok vagy a központi memória méretéből adódó/ többékevésbé elfogadható korlátozása mellett teljesitik a fenti egyenlőtlenségeket "értelmes" C_1, C_2 konstansokkal. Mindazonáltal - korlátaik ismeretében - ezek a módszerek a gyakorlatban jól használhatóak, hogy melyik mikor, az a továbbiakból szándékunk szerint kiderül.

II.

Az ismertetésre kerülő módszerek közös jellemzői - és ezeket ezért együtt is tárgyaljuk - a következők:

1. A rekordokat a klasszikus hashing-nél is alkalmazott csomagokban /bucket/ tároljuk.

2. Ha a file telitődik,uj csomagot kapcsolnak hozzá, az uj csomag eléréséhez szükséges információkat egy táblázatban tárolják. A táblázat /directory/ minden ismertetésre kerülő módszerben egy bináris fa reprezentációjának tekinthető.

Az egyes algoritmusok közötti különbség a fa tárolási módjában van, ill. abban, hogyan definiálják a file telitődését.

Ha az állományhoz tartozó tartalomjegyzéket egy m db bináris fából álló erdővel ábrázoljuk, egyelőre eltekintve annak konkrét tárolási módjától, könnyen megérthetjük a dinamikus hashing módszerek közös elvét. A fák levelei a rekordelőfordulásokat tartalmazó csomagokat reprezentálják. Egy adott kulcsból először is /ha az erdő több fából áll/ az $\{1,2,\ldots,m\}$ halmazra való leképezés segitségével meghatározzuk, hogy a rekordot tartalmazó csomag melyik fa levele. Ezután kiinduló értékként használva a rekord kulcsát, elő[illegible] egy bináris sorozatot, amely megmondja, hogy milyen [illegible] csomaghoz. /A kulcshoz tartozó bi[illegible] előállító algoritmus lehet pl. egy pszeudo-[illegible] -generátor./ Most már ebben a csomagban megkeresheti[illegible] a rekordot - ha ez a feladat -, illetve beilleszthető[illegible], ha még elfér. Ha egy beillesztés során a fentiek szerint meghatározott csomag betelik, az alábbi eljárást követjük:

1. nyitunk egy uj üres csomagot, ezt hozzáfüzzük a féhoz, mint az éppen betelt csomagot reprezentáló csucs jobboldali fiát;

2. a betelt csomagot szintén hozzáfüzzük a féhoz,mint a keletkezett csomag testvérét;

3. a kulcsból meghatározható bináris sorozat alapján minden

rekordról döntünk, hogy átkerül-e a jobboldali új csomagba, vagy marad a helyén /ehhez semmi mást nem kell tenni, mint a bináris sorozatot egy elemével tovább figyelembe venni - tehát az eddig egy csomagban tárolt rekordokhoz tartozó elérési ut a fán az utolsó előtti lépésig továbbra is közös marad/.

Rekordok törlése során ha két testvérben a rekordok száma együttesen annyira lecsökken, hogy már egy csomagban is elférnek, a két testvért a fenti algoritmus megforditásával összemásolhatjuk.

Knuth [2]-ben bebizonyitotta, hogy a csomagok telitettsége ln 2(≈0,69) körül várható. Ha a csomagok nem osztódnak azonmód, amint beteltek, hanem a klasszikus hashing-ből ismert módon tulcsordulási területeket rendelünk hozzájuk, ez az arány kb. 0,1-el javitható az elérési idő számottevő romlása nélkül.

Az előadásban röviden ismertetésre kerülnek a legalapvetőbb dinamikus hashing módszerek, ugy mint: "dynamic hashing" /[3], [8]/, "extendible hashing" /[1], [6]/, "linear hashing" /[4], [5], [9]/ ill. ezek különböző változatai.

III.

Megvizsgáljuk, hogyan viselkedik az F(n) és T(n) függvény n különböző nagyságrendü értékeire.
A gyakorlatban előforduló feladatok illusztrálására három példát dolgoztunk ki /1., 2., 3.táblázatok/. A példák számolása során olyan számitógépet tételeztünk fel, amelynek mind belső -, mind háttérmemóriája 32 biten cimezhető.

Az elméleti számitásokból ill. számitógépes szimulációk alapján kapott eredményeket a 4. táblázat foglalja össze. A táblázatban szereplő képletek a tárolókihasználás, elérési szám és az indextábla méretének nagyságrendi becslését adják. Már a nagyságrendi becslésekből látszik, hogy a "dynamic hashing"és a "linear splitting" módszerek csak abban az esetben javasolhatók alkalmazásra, ha a tartalomjegyzék kiinduló feltételezéseinkkel ellentétben a központi memóriában tárolható. Ha az elérési szám alacsonyan tartása nem elsőrendü fontosságu, a "linear splitting"módszer még ott is alkalmazható, ahol a "dynamic hashing" már a belső memória korlátai miatt nem. Ha a fenti feltételek nem teljesülnek, marad az "extendible hashing" alkalmazása. Ez konstans elérési számot produkál. A kitöltési tényező értke elméletileg 0-hoz tart egyre, mindazonáltal a gyakorlatban, a mai számitógépek mé-

reteit figyelembe véve általában lehet olyan csomagméretet választani, ami megfelelő tárolókihasználást eredményez.

csomagméret / rekordszám	b=1o	b=2o	b=5o
$1o^5$	35o	175	7o
$1o^6$	35oo	175o	7oo
$1o^7$	35ooo	175oo	7ooo

1. táblázat. Példák a "dynamic hashing" indextáblájának méretére /Kbyte/.

csomagméret / rekordszám		b=1o	b=2o	b=5o	F(n)
γ=2	$1o^5$	3o	15	6	1,7
	$1o^6$	3oo	15o	6o	
	$1o^7$	3ooo	15oo	6oo	
γ=3	$1o^5$	2o	1o	4	2,1
	$1o^6$	2oo	1oo	4o	
	$1o^7$	2ooo	1ooo	4oo	

2. táblázat. Példák a "linear splitting" indextáblájának méretére /Kbyte/ és az elérési számokra, ha az indextábla a központi tárban van / a módszer egyik paramétere, l. [9]/.

csomagméret / rekordszám	b=1o	b=2o	b=5o
$1o^5$	o,68	o,68	o,68
$1o^6$	o,57	o,61	o,65
$1o^7$	o,47	o,54	o,62
$1o^8$	o,39	o,48	o,59

3.táblázat. Példák az "extendible hashing" kitöltési tényezőjének alakulására.

	indextábla mérete	F(n)	T(n)
"dynamic hashing"	$\frac{C_1 n}{b^2 \ln 2}$ [2]	$\log_2 \frac{n}{b}$	$\frac{\ln 2}{\frac{C_1}{C_2} + 1}$
"extendible hashing"	G(n) [6]	<3 [1]	$\frac{n}{G(n)+Cn}$
"linear splitting"	$\frac{C_1 n}{\gamma^2 b}$ [9]	Cn	$>C(\gamma)$

4. táblázat.

A közölt képletekben szereplő változók jelentése a következő:

n - rekordszám,
b - csomagkapacitás,
C_1 - a cim hosszától és a pointerezés megvalósitásától függő konstans,
C_2 - C_1/B , ahol B a csomag hossza byte-okban kifejezve,
G(n) - olyan függvény, amelyre teljesül: $G(2n) = 2^{1+\varepsilon} G(n)$,
$C(\gamma)$ - egy γ-tól függő konstans.

Irodalomjegyzék

[1] R.Fagin, J.Nievergelt, N.Pippenger, R.Strong, "Extendible hashing - A fast access method for dynamic files", ACM Trans. Database Syst., vol. 4/1979/ pp. 315-344.

[2] D.E.Knuth, The art of computer programming, vol.3: Sorting and searching. Addison-Wesley, Reading, Mars.,1973.

[3] P.Larson,"Dynamic hashing", BIT, vol. 18/1978/, pp.184-201.

[4] P.Larson, "Linear hashing with partial expansions" Proc. 6th Int. Conf. on Very Large Data Bases, Montreal 1980, pp. 224-232.

[5] W.Litwin, "Linear hashing: A new tool for file and table addressing", Proc. 6th Int. Conf. on Very Large Data Bases, Montreal 1980, pp. 212-223.

[6] H.Mendelson, "Analysis of extendible hashing", IEEE Transactions on Software Engineering, vol. 8./1982/, pp.611-619.

[7] J.K.Mullin, "Tightly controlled linear hashing without separate overflow storage", BIT 21/1981/, pp.390-400.

[8] K.Ramamohanarao, J.W.Lloyd, "Dynamic hashing schemes", The Computer Journal, vol. 25/1982/, pp.478-485.

[9] M.Scholl, "New file organizations based on dynamic hashing" ACM Trans. Database Syst., vol. 6/1981/, pp.194-211.

Velkei Sándor, SZKI

Teledata bázisu alkalmazói rendszerek

Előljáróban meg kell emliteni: az előadást azzal a nem titkolt céllal terveztük, hogy a figyelmet a TELEDATA rendszerre irányitsuk, bizonyitva alkalmasságát vállalati információs rendszerek kialakitására.

Meg kivánjuk győzni a szervezőket, hogy további alkalmazásokban számitsanak /épitsenek/ a TELEDATA rendszer szolgáltatásaira, s meg kivánjuk győzni az alkalmazókat, hogy igényeljék, mint adott esetben a legalkalmasabb megoldást, a TELEDATA rendszer szolgáltatasait.

A TELEDATA rendszer alkalmassága vállalati környezetben történő felhasználásra

A TELEDATA tipusu alkalmazásokat általában mint póstai, vagy póstai hálózaton elérhető szolgáltatásokat mutatják be. Eddigi fejlesztéseink alapján ugy tünik, zártkörü TELEDATA rendszerek alkalmazása, belső, vállalati információs rendszerekben hamarabb realizálható, mint a fentiek.

A TELEDATA rendszer biztositotta előnyök a ma általánosan elterjedt képi megjelenitéssel szemben:

- az alkalmazás fejlesztési költségei alacsonyabbak, gyorsabban hoz eredményt, mint más rendszerek - a vezérlő program számos funkciót átvállal az alkalmazás vezérlésében. Legszembetünőbb példája ennek, mikor az alkalmazás csak informativ jellegü statikus információt kezel /népességnyilvántartás, utazási ajánlatok, ... stb./;

- a számitógépes szolgáltatás egyszerü eszközökkel, laikusok számára is, különösebb előképzettség nélkül hozzáférhetővé válik - infra távvezérlő, általánosan elterjedt szines televizió;

- szines képmegjelenités - a szines kép vizuális élményt jelent /főként laikusok számára meggyőzőbb, fölkelti a figyelmet/; a szinekkel a tartalom kiemelhető, hangsulyozható;

- grafikus megjelenités - hasonlóan az előbbiekhez, a grafika a megjelenités hatását fokozza, grafikonok ábrázolásával a kiértékelhetőséget növeli;

- számitógépes háttér - a müködő alaprendszerek fejlettségétől függően a vállalati adatbázisokat bekapcsoló, dinamikus, naprakész információ megjelenités a fenti formában;

. a terminálok ára alacsony, alkalmasint megjelenése bizonyos helyeken esztétikusabb más berendezéseknél - utazási irodák, pályaudvar, tanácsterem, dolgozó szobák, ... ;

. működtetés távvezérléssel - a szines televizióknál ismert technika felhasználásával.

A fölsoroltaknak, főként az egyszerüségnek ára van, ami az alábbiakban jelentkezik:

. kötöttségek a képszerkesztésben és az információ forgalomban - többnyire az alkalmazott protokollból származó megkötések;

. az ismert képernyős terminálóknál szükebb, egyidejüleg megjelenithető információ mennyiség.

Mindezeket összevetve, az alkalmazási kisérletek alapján megállapitható, hogy számos alkalmazási terület van, ahol az előnyök fokozottan jelentkeznek és a korlátozások igényesebb tervezéssel háttérbe szoríthatók.

TELEDATA bázisra épülő alkalmazások tervezésének sajátosságai

TELEDATA alkalmazás esetünkben vállalati adatok megjelenitését jelenti, speciális lehetőségekkel rendelkező szines képernyős terminálokon, előre meghatározott összetételben, egy előre definiált vezérlési séma által meghatározott sorrendben.

Egy TELEDATA alkalmazás tervezése és kiépitése két lépésben történik:

. az alkalmazói rendszer kiszolgáló környezetének tervezése és létrehozása;

. az alkalmazói rendszer tervezése és létrehozása.

Egy TELEDATA környezet az alkalmazói rendszer számára a következő környezeti elemek kialakitását jelenti:

. képállomány tervezése, szerkesztése és feldolgozása,

. vezérlő struktura /menü/ tervezése és generálása.

A TELEDATA képeket alkalmazói információk alapján grafikus formában tervezni kell. A kép tartalmazhat az információtartalomra, a rend-

szerben elfoglalt szerepére utaló jegyeket /főként grafika, vagy szín, stb./; statikus, a szerkesztésnél fölvitt adatokat és/vagy dinamikusan változtatható adatmezőket: a választékot a választást segitő szövegrésszel, jelezve a vezérlésátadás módját és irányait.

A tervezett képek szerkesztő terminálon nyerik el végleges formájukat. A különböző alkalmazásokban felhasznált képeket a TELEDATA rendszer egy közös kép adatbázisban tárolja. A kép adatbázis töltését, karbantartását szervizprogramok biztositják.

A TELEDATA vezérlő struktura /menü/ a vezérlési séma /gráf/, amely alapvetően meghatározza egy alkalmazás működési mechanizmusát. A tervezés egyik legfontosabb lépése a vezérlő gráf szimbólikus leképezése.

Egy alkalmazás vezérlő strukturája leiró nyelv segitségével definiálható. A menü gépi reprezentációját, a leirás alapján TELEDATA szervizprogramok állitják elő és helyezik el a közös menü adatbázisban.

A vezérlő struktura tervezése az alkalmazás tervezés fontos eleme. Megfelelő szerkezet kialakitásával a tranzakciók logikai felépitése lényegesen egyszerüsithető. Ez egyrészt előnyt jelent a kivitelezésben /egyszerü, gyors kivitelezés/; másrészt előnyt jelent a végrehajtásban /válaszidők és az erőforrások leterhelésének csökkenése/.

A szüken vett alkalmazói rendszer elemei:

- tranzakciók /programok/,
- alkalmazói adatbázisok.

A tranzakciók olyan alkalmazói programok, amelyek kielégitik a TELEDATA rendszer és annak telekommunikációs környezete által szabott feltételeket, s egyben fölhasználják ezen rendszerek szolgáltatásait.

Az alkalmazói adatbázis állhat egyszerü, hagyományos állományokból, vagy épülhet valamilyen adatbázis kezelő rendszerre.
/Több szempontot figyelembevéve a tárgyalt alkalmazás hagyományos állományokra épül./

TELEDATA bázisra épülő alkalmazás - Borsodi Szénbányák Vezetői Információs rendszere

A Borsodi Szénbányák vezetői információs rendszerének fejlesztése az CMFB - BSz - SzKI között fennálló kutatás-fejlesztési szerződés keretében indult. A rendszer kidolgozásához 1983. februában u.n. nag.,vonalu rendszerterv készült. A nagyvonalu terv alapján kerül sor a vezetői információs rendszer részletes, a Számitástechnikai Koordinációs Intézetben kifejlesztett SZKITA - TELEDATA rendszerben történő kivitelezésére.

A BSz vezetői információs rendszere egy központi adatbázisra épülő TELEDATA bázisu lekérdező rendszer, amely az alábbi modulokból épül fel:

- termelési modul,
- szénbányák jellemzői modul,
- vállalati eredmények modul,
- anyaggazdálkodási modul,
- beszerzések és árváltozások modul,
- létszám-, bér-, müszak-, teljesitmény-modul.

További csoportositásban, a rendszer tartalmaz

- kiszolgáló elemeket, amelyek a központi TELEDATA adatbázis feltöltését, karbantartását végzik /a TELEDATA adatbázis hagyományos állomány-kezelésére épül/,
- lekérdező elemeket, amelyek a TELEDATA szolgáltatásokra épülve, definiált összetételben és kapcsolatrendszerben az információt megjelenitik.

A kivitelezés modulonként, folyamatosan történik. A kivitelezésben a felek együttmüködnek oly módon, hogy a software keretek és a lekérdező rendszer elemeinek kidolgozása az SzKI, mig az adatbank karbantartó, feltöltő modul-k elkészitése a BSz feladata.

A vezetői információs rendszerből elsőként az anyaggazdálkodási modul készült el.

Az anyaggazdálkodási modul a vállalat gazdálkodása szempontjából lényeges anyagfelhasználás és készlet érték adatokat kezeli.

A kiszolgáló modulok a vezetői döntések alapján és a vállalat egyéb információforrásaiból létrehozzák az adatbázist képező állományokat, valamint gondoskodnak azok folyamatos feltöltéséről és karbantartásáról.

Az anyaggazdálkodási modul elemei:

- anyagfelhasználás ág:
  kezeli a napi-, havi-, üzemenkénti anyagfelhasználás értékeket.

- készletnyilvántartási ág:
  a készletmozgások kvázi naprakész nyilvántartására épülve kezeli az aktuális havi-, raktári- és fatelepi készlet értékeket.

- kiemelt anyagok felhasználása és -készlete, villamosenergia-felhasználás ág:
  kezeli a gazdálkodás szempontjából fontos anyagok havi felhasználás és pillanatnyi készlet érték adatait, valamint a villamosenergia-felhasználást.

Az alkalmazói rendszer havi adatokat az akutális évben az aktuális hónappal bezárólag és az elmult évben a teljes évet tekintve; napi adatokat az aktuális hónapban és az azt megelőző hónapra tárol.

Az alkalmazás jellemző tulajdonsága, hogy közvetlen módon mindig aktuális adatokat szolgáltat, de lehetőséget biztosit egy választott dátum alapján is a lekérdezésre, ha az adott terminusra vonatkozóan létezik tárolt információ.

A rendszer az egyes elemek között időbeli és egyéb logikai kapcsolatot föltételez és ezt a vezérlő gráfban tükrözi, pl. vezérlésátadás történhet /választható/ nemcsak készlet —>raktári készlet, illetve készlet —> fatelepi készlet relációban, hanem aktuális raktári készlet —> aktuális fatelepi készlet relációban is.

A TELEDATA rendszer hardware/software alapelemeit és az alkalmazások bizonyos elemeit az SzKI elsőként a '82 BNV-n mutatta be.

A szóbanforgó anyaggazdálkodási modult a felhasználónak 1983. juniusban az SzKI R15/R16 számitógép rendszerén átadtuk.

TELEDATA szolgáltatásra épülő alkalmazói rendszerek erőforrás igényük alapján, különböző kapacitásu számitógépekre telepithetők. Erre épitve az SzKI elkészitette személyi számitógépén müködő TELEDATA rendszerét. A rendszer megoldását tekintve olyan, hogy ezáltal a személyi számitógép akár önálló-, akár részfeladatok megoldására alkalmas /raktárkészlet nyilvántartás, vagy képszerkesztés... stb./.

Minden egybevetve, egy feladat eldönti, hogy megoldása milyen keretek között legcélszerübb. A már korábban deklarált célunk az volt, hogy bemutassuk, bizonyos alkalmazásoknál a TELEDATA tipusu megoldás a korábbiaknál célravezetőbb, s az erőforrások széles körében /ESzR, IBM, személyi számitógépek.../ van mód ezen lehetőség kihasználására.

Lovász Attila PSZTI

Real-time adatbáziskezelő rendszerek teljesitményvizsgálata és tervezése modellezéssel

Az ismertetésre kerülő software-teljesitmény vizsgálatok a real-time tranzakciós rendszerek középpontjában álló alapsoftware, az adatbáziskezelő rendszer teljesitményparamétereinek /átbocsátóképesség, válaszidők eloszlása, stb./ elsősorban kisérleti uton történő meghatározását jelentik és a benchmarking fogalomkörébe sorolhatók. Az adatbáziskezelő rendszerek nagy volumenü alkalmazások bevezetése előtt elvégzendő teljesitményvizsgálatát indokolttá teszi az a tény, hogy a dokumentációk általában nem tartalmazzák - az alkalmazás-specifikusság miatt nem is tartalmazhatják - a rendszer valamenynyi ismérvét, amely alapján optimalizált rendszerek tervezhetők.

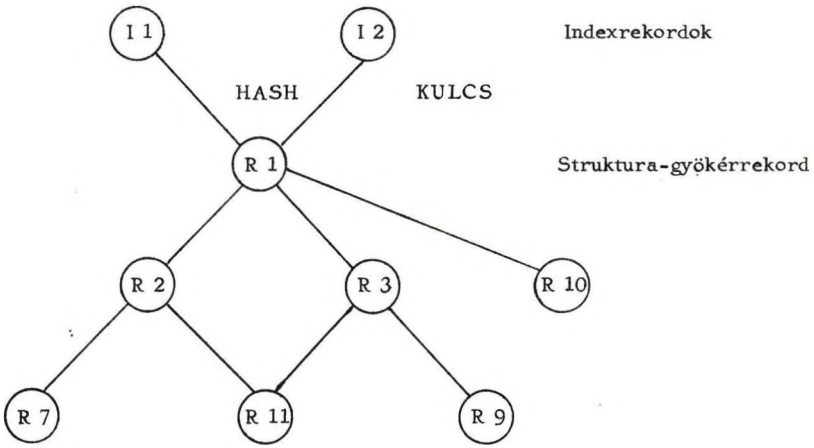
A PSZTI és a KFKI által a TRACCS-11 kommunikációs hálózati tranzakciókezelő rendszer adatbáziskezelő alrendszerében, a DSMS-ben végzett vizsgálatokat kettős cél motiválta.

1. Nyujtsanak támpontot az Állami Biztositóban tervbevett országos hálózati tranzakciós rendszer tervezéséhez és méretezéséhez; adjanak becsléseket egy megvalósitható rendszer várható átbocsátóképességéről, szük keresztmetszeteiről.

2. Ujonnan forgalmazott és vásárolt software-termékről lévén szó, a vizsgálatok foglalják magukban a software-feltárás munkáit is. Ennek keretében állapitsák meg a DSMS-re általánosan jellemző teljesitménykorlátokat és azokat a lehetőségeket, amelyekkel a hatékonyság lényegesen befolyásolható.

A vizsgálatok egy részének módszeréül azt a megoldást választottuk, hogy a Biztositó által rendelkezésre bocsátott információk alapján jóval kisebb, laboratóriumi méretekben létrehozunk egy modell-adatbázist, amely strukturá-

jában, a rekordkapcsolatok eloszlásában megfelelt az alkalmazás során előálló adatbázisra várható értékekkel, azonban a tipuskapcsolatok leírásában jóval egyszerübb volt.



ÁBRA: A modelladatbázis tipuskapcsolatai

Ezt azzal értük el, hogy az eredeti adatbázisban több, a strukturában hasonló helyzetben lévő rekordtipust, a kapcsolataik eloszlásával együtt összevontunk. Ezt az tette lehetővé, hogy vizsgálatainkban elvonatkoztattunk a programfunkcióktól: a felhasználói programok overhead-jét nem vettük

számitásba, csak a szigoru értelemben vett, a software által megvalósitott adatbázismüveletek müveletigényét és sebességét vizsgáltuk. Ebben a megközelitésben az adatbázis csupán egy "számrendszert" definiált, amely által meghatározott pályákon a tranzakciók "közlekedhetnek". Az adatzsugoritás során merült fel az a technikai probléma, hogy a kétirányu pointerekkel adminisztrált rekordkapcsolatoknak az a konzisztenciája, amelyet a rekordok adatbázisba kerülésének "természetes módja" automatikusan megőriz, ne sérüljön meg a generálás során sem.

Kidolgoztunk egy algoritmust és egy programcsomagot, amely a rekordkapcsolatok megadott "független" eloszlásaira, a rekordtipusok közötti kapcsolatok bármely strukturájára megoldja a problémát.

Az igy előállitott laboratóriumi adatbázist azután kétféleképpen hasznositottuk mérések elvégzésére:

- A tervbevett adatbázisfunkcióknak megfelelő "tranzakciókeveréket" definiáltunk, amelyekkel részben a várható batch-futtatások, részben az online hálózati futás várható teljesitménykarakterisztikáit vizsgáltuk. Mértük a szekvenciális, direkt elérés, módositás, törlés, beszurás tranzakciók végrehajtási idejét.

- Modellkisérleteket végeztünk, amelyekkel az egyes izolált adatbázismüveletek müveletvégzési sebességét és a különböző indexszervezési módszerek /hash-kulcs/ hatékonyságát vizsgáltuk; ehhez a modell-adatbázist egy átlagot képviselő mérési környezetként használtuk.

A vizsgálatok eredményeként a rendszerben több meglepő kritikus pontra, potenciális szük keresztmetszetre bukkantunk, és az adatbázis és program-

tervezés számára megfogalmazhattuk kivédésük lehetséges módszereit. Igy pl. a hash-index módszer által kinált elvi hatékonyság csak a kulcs megfelelő ábrázolási módjának megválasztásával használható ki, másképpen alkalmazása nagy kihasználatlan file-területet és többszintü indexstrukturákat eredményez. Az olyan alkalmazásoknál pedig, ahol az adatbázis olyan rekordjainak törlését kell végrehajtani, amelyek egy másik rekordhoz több szintre hasadt indexhierarchián keresztül kapcsolódnának és a törlés az indexhierarchia inverz végigjárását is igényelné, a törlés a batch feldolgozás szempontjából is reménytelen, rekordonkénti több perces nagyságrendnyi ideig tartana.

A vizsgálat eredménye volt a rekordok közötti kapcsolatok többszintü indexhierarchiáját felépitő és kezelő algoritmus lényeges elemeinek lokalizálása. Ismeretében rögzitettük a rekordok kontroll-területe, a rekordkapcsolatokat adminisztráló pointermezők méretei tervezésének azon módszerét, amellyel a tárgazdálkodás és adatelérés szempontjából kiegyensulyozott, hatékony adatbázisstruktura alakitható ki.

Az empirikus vizsgálatokról nyert tapasztalataink birtokában általában elmondható, hogy a teljesitménymeghatározásnak ez a szintje a rendszertervezés számára szükséges információk megszerzésének gazdaságos módszere. A kapott eredmények kiindulási paramétereit képezhetik egy teljes hálózati rendszer /- amelynek középpontjában az adatbáziskezelő áll -/ más uton, szimulációs vagy analitikus eszközökkel történő tanulmányozásának.

Bertalan József
Honvédelmi Minisztérium
Országos Müszaki Fejlesztési Bizottság
KFIIR Kidolgozó Csoport

A Kutatási Fejlesztési Irányitási Információs Rendszerről

A Magyarországon folyó K+F tevékenységek országos szintü áttekintésének igénye éppugy mint az egyes K+F helyek relativ "információs elszigeteltsége" sok olyan problémát vet fel, amely indokolttá teszi egy - ezzel a területtel foglalkozó - információs rendszer felállitását. Ahogy azt a Tudománypolitikai Bizottság számára készitett egyik jelentés 1979-ben megállapitotta:

- a meglévő nyilvántartási rendszer bonyolult, megalapozott információszolgáltatásra nem alkalmas, összetevői /a Kutatási és Fejlesztési Statisztika, az Országos Kutatásnyilvántartás/ az érvényben lévő rendeleteknek ugyan lényegében megfelelnek, de sem egymással, sem a valósággal nincsenek összhangban;

- az adatok feldolgozása decentralizáltan és részben manuálisan, részben számitógépen történik, ezért a kutatás és fejlesztés irányitásához és tervezéséhez korlátozott lehetőséget nyujt;

- a K+F tevékenység költségkihatására, az eredmények gyakorlati hasznositásának felmérésére és ellenőrzésére, a gazdasági hatékonyság megitélésére a nyilvántartási rendszer nem ad lehetőséget;

- nem megfelelő az egyes kutatóhelyek, szervezetek adatszolgáltatási felelősségének és kötelezettségének, továbbá adatszolgáltatási területeinek meghatározása és igy az információs szervezetek egymástól többé-kevésbé függetlenül - a gyakorlatban megvalósult módszerek szerint - önállóan tevékenykednek /többszörös adatközlési redundancia mellett tájékozatlanok maradnak/;

- nincs biztositva a K+F tevékenység kivánatos mértékü átfogó áttekintése és ellenőrzése, a felesleges párhuzamos-

ságok kiküszöbölése, illetve a témák kidolgozásával kapcsolatos tárgyi és határidőbeli önkényes eltérések feltárása.

A felsorolt problémák legalább részbeni felszámolása érdekében - több korábbi, hazai és külföldi kisérlet tapasztalatainak felhasználásával - 1979-ben egy uj program indult az ebben az előadásban ismertetendő KFIIR rendszer felállitására.

A KFIIR rendszer számára célként volt megfogalmazható, hogy javitani kell az irányitás, a kutató-fejlesztő munka koordinálásának szinvonalát. Összehangoltan, az egyes részterületek között kiegyensulyozottan szükséges gyüjteni a kutatásra, fejlesztésre és irányitásra stb. vonatkozó információkat. A lehetséges és megfelelő hatékonyság érdekében megbizható adatok szükségesek a kutatási és fejlesztési tevékenységekről, a fejlesztésre forditott anyagi eszközökről, ezek megoszlásáról, az eredmények hasznositásáról, a kutató-fejlesztő helyek összetételéről, szerkezetéről.

Ennek érdekében a kutatási-fejlesztési tevékenységek információinak feldolgozására olyan rendszert fejlesztünk ki, amely:

- biztositja a kutatási-fejlesztési feladatok nyilvántartását és megengedi a bővitő értelmü ujraszervezést;
- megfelelő aggregáltságu információkat szolgáltat a K+F helyeknek és a tudománypolitika irányitóinak;
- biztositja a kutatási-fejlesztési tevékenységek pénzügyi-anyagi ráforditásainak áttekintését;
- elősegiti a kutatási-fejlesztési eredmények hasznositását;
- az eddiginél pontosabban nyilvántartja a kutató-fejlesztő helyek adatait.

A kitüzött célokat - a feldolgozandó információk nagy mennyisége és a közöttük fennáló bonyolult kapcsolatok miatt - egy számitógépes adatfeldolgozó rendszerrel támogatott információs rendszer keretében lehet elérni.

A célokból az információs rendszerre levezethető követelmények két csoportra voltak oszthatók./1/ Az első csoportba a K+F szféra információs, vagy pontosabban adatmodelljére, mig a másikba az adatmodell alapján elégségesen müködni képes adatfeldolgozó rendszerre vonatkozók tartoztak.

/1/ A követelményeknek most az informatikai vonatkozásairól van szó. Természetesen nem az államigazgatási, szervezeti, jogi vonatkozások "rovására", és nem az ezekkel szembeni elsőbbséget feltételezve vizsgáljuk tárgyunkat.

Az adatmodell tisztázása érdekében számos K+F intézményben és irányitó hatóságnál folytattunk konzultációt. A konzultációk eredményeként nyilvántartásunk objektum tipusaiként:

1. A hazai K+F tevékenységek bejelentései,
2. A nemzetközi együttmüködésben végzett K+F tevékenységek bejelentései,
3. A licence, know-how vásárlások és eladások,
4. A K+F célu ki- és beutazások,
5. A kutatási dokumentációk,
6. A K+F intézmények,
7. A kutatással-fejlesztéssel foglalkozó személyek

tárgykörei voltak rögzithetők. Javaslatként, illetve további tisztázás után bővitésként jöhetnek még számba a:

8. A nagyértékü és/vagy egyedi müszerek,
9. A K+F tevékenységek hasznositásai,
10. A K+F témák kapcsán keletkezett publikációk

tárgykörei.

Az 1-7 tárgykör csoportra szoritkozva az 1-5 pontok az általánosabb értelemben vett K+F tevékenységeket, mig a 6-7 pontok az un. K+F bázist jelentik. Az egyes tárgykör tipusok - de különösek a K+F tevékenységek csoportba tartozók - elemzése közben az derült ki, hogy:

- a tárgykörök jellemzésére használható egyes adatok nem formalizálhatók, csak szöveges leirásuk adható meg és mint ilyenek, gyakorlatilag végtelen értékkészlettel rendelkeznek; /2/
- nagyon sok az egyazon tárgykörön belüli más objektum előfordulásra, és a más tárgykörök előfordulásaira vonatkozó utalás, azaz nagyon sürü a kapcsolattipusok hálója;
- az egyes tárgykörök előfordulásai /például egy konkrét hazai K+F téma/ adataikban időfüggőek. Időfüggőkegyrészt abban az értelemben, hogy felbukkanásukkor "tervezet" jellegüek és később "tény" jellegüvé alakulnak, másrészt abban az értelemben, hogy vannak az objektum felbukkanásakor nem

/2/ Ugyancsak a szöveges adatelemek irányába tolta el a rendszert az a tény, hogy a rendszer adatszolgáltatói /és igénylői is/ nem számitástechnikusok. Részükre formalizált előirást vagy kódkönyveket kiadni nemcsak lehetetlen de értelmetlen is. Az ilyen szervezés a K+F szféra nem eléggé előrelátható változásai és rendkivüli tematikus szélessége miatt elvileg nem keresztülvihető.

értelmezhető vagy nem ismert adatok /például a K+F témák dokumentációira utalók/, amelyek az objektum jellemzőinek csak fokozatos, mozaikszerü felgyüjtését teszik lehetővé.

A felsorolt adattulajdonságok véleményünk szerint eltérőek a hagyományosan megszokottakról, érvényesitésük az adatmodell uj vonásaihoz vezetett. Természetesen vannak a felsorolt tulajdonsággal nem rendelkező adatok is, ezek kezelésére a CODASYL modell elvei jól érvényesithetők. Mivel ez utóbbiak tartalmuk szerint az objektumok állapotára vonatkozó információt hordoznak/jól formalizálahtó alakban/tényadatoknak/faktografikus adatoknak neveztük. Szembeállitva ezzel az első csoport adatai általános tulajdonságuk szerint a szöveges adatok nevet kapták.

Az eszköz vonatkozásában választásunk az IDMS-re esett, de ennek egyenlőre csak az elveit alkalmaztuk. A fokozatos rendszerépitésre vonatkozó elképzelésünk szerint az IDMS környezetébe továbbvihető, de hagyományos adatkezelési technikával is müködtethető adatmodellt és szervezést valósitottunk meg. Ebben érvényesitettük a problematikus uj tulajdonságokat:

- az adatmodell bővitéssel való továbbépithetőségének eszméjét, /a fizikai adatbázis ujraszervezésének kényszere nélkül/;
- az automatikus szövegnormalizálás eredményeként a generált adattipusok definiálásának lehetőségét;
- az ideiglenes kapcsolattipusok kezelhetővé tételét.

Az adatmodellünk témavázlatát Bachmann-diagramm formájában az 1. sz. ábra adja.

Az adatmodellről elmondottak /és a szervezeti követelmények/ az adatfeldolgozó rendszer kialakitását eléggé szigoruan determinálták.

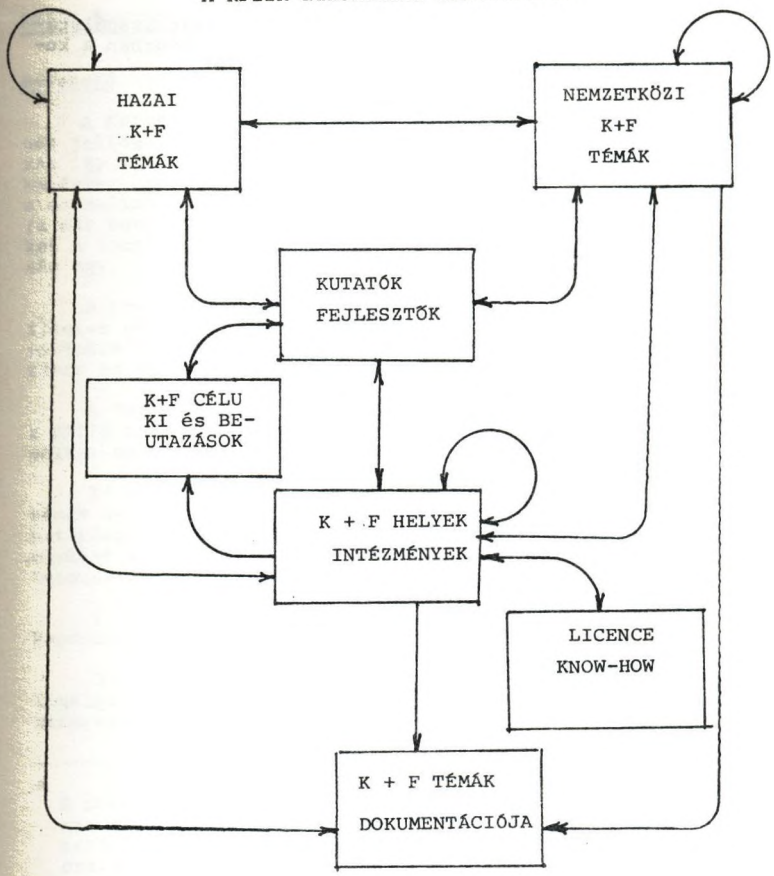
A fokozatos kifejleszthetőség érdekében jelenleg csak az egyes tárgykörök elkülönült feldolgozására alkalmas funkciókat épitettük ki. Ezen funkciók azonban a tárgykörön belüli kapcsolatokat kezelik. Elsősorban arra törekedtünk, hogy a szöveg és adatnormalizáláshoz szükséges belső kapcsolatok éljenek. A funkciók IDMS környezetbe való továbbvihetőségének érdekében az adatkezelést egy interface-en keresztül bonyolitjuk, amely cserélhető.

Az adatszervezésben lehetséges változásokhoz, az adathiányos rekordelőfordulásokhoz, a szöveges adatok terjedelembeli dinamizmusához egy,a KGST országok által elfogadott NME2 rekord szervezési szabvány segitségével alkalmazkodunk.

Az adatnormalizálás céljára - ahol lehetséges- egy olyan ellenőrző tábla kezelő rendszert alkalmazunk, amely az ellenőrző konstansokat logikai kapcsolataik alapján hálós szerkezetben képes kezelni.

1. sz. ábra

A KFIIR adatmodell sémavázlata



Végezetül néhány adatot szeretnénk ismertetni a KFIIR rendszer jelenlegi állapotáról:

- a rendszer nyilvántartó funkciói már müködnek,
- végrehajtottuk 25ooo hazai K+F téma bejelentésének feldolgozását,
- a témák /bejelentések/ automatikus hasonlóságvizsgálatának végrehajtásával segitséget nyujtunk elsősorban a kutató helyek számára.

Lőrincz István
Honvédelmi Minisztérium

KFIIR fogalmi szótár kezelő és karbantartó rendszer

Bevezető

A KFIIR rendszer adatfeldolgozó rendszere elsősorban szöveges jellegü, un. dokumentum tárgyköröket kezel, az adatfeldolgozás igy alapvetően szövegkezelést jelent. A szövegkezelés a rendszer input oldalán normalizáló lépések sorozatára bomlik; a normalizálás célja a dokumentumok olyan átalakitása, amely /a már bevitt dokumentumokat figyelembevéve/ az ujonnan bekerülőket a többivel összehasonlithatóvá teszi. Az összehasonlithatóság egy, a dokumentumtérben bevezetett kvázimetrikán nyugszik.*

A normalizáló lépések nyelvfüggő lexikai /szövegelemfeltáró/, illetve részben nyelvfüggetlen szemantikus /fogalomfeltáró/ csoportokra oszthatók. Mind a két csoport támaszkodik a fogalmi szótárra és annak kezelő rendszerére.

A fogalomfeltáró szótárral /FFSZ/ kapcsolatos követelmények a KFIIR rendszer egyéb részeinek függvényében a következőképp voltak megfogalmazhatók:

1/ Az FFSZ tegye lehetővé több nyelv szó- és fogalom készletének kezelését. A többnyelvüség ellenére az FFSZ a lehetőségek határáig támaszkodjon közös /adat-/szervezési elvekre, igy minimumként tartsa be az ISO és a KGST többnyelvü információkezelő tezauruszokra vonatkozó előirásait;

2/ engedje meg néhány a lexikai elemzéshez szükséges szókapcsolat ábrázolását /nyelvfüggő módon/;

3/ engedje meg néhány a szemantikai elemzéshez szükséges fogalmi kapcsolat ábrázolását /nyelvfüggetlen módon tezaurusz-struktura formájában/;

---

* A dokumentumtér metrizálása nemcsak egy szabványos matematikai és dokumentumábrázolási /tárolási/ formalizmust tesz lehetővé, hanem segitségével a visszakeresési módszerek egész családja vált kipróbálhatóvá és egymással összehasonlithatóvá. Igy a hagyományos kulcsszavas módszerek is.

4/ automatizáltan biztositsa a tezaurusz-struktura ellentmondás- és redundanciamentes /tovább/építésének lehetőségét;

1. Fogalmak és osztályozásuk - a fogalmi szótár

A KFIIR szöveges adatkezelő rendszerében szótári bejegyzés-nek nevezünk minden olyan /1/ a rendszer által ismert nyelvü /2/ szövegrészletet, amely /3/ alkalmas egy vagy több dokumentum jellemzésére. /Számunkra az egyik legkitüntetettebb osztály a fogalom megjelölésre használható névszói kifejezéseké, ezért a szótár neve fogalomfeltáró szótár/.

A fogalomfeltáró szótár bejegyzések strukturált halmaza, amelyben a strukturát fogalmi és formai kapcsolatok biztositják.

A szótári bejegyzések értelmezését egészitsük még ki a következő jegyekkel: /4/ a szótári bejegyzések tetszőleges hozzusága karakter stringek, /5/ amelyeknek kódja van, és /6/ aspektusa/i/ van/nak/. Aspektuson olyan önkényesen kiválasztott osztályozó jelzést értünk, amely a szótárbejegyzés különböző értelmezéseinek megkülönböztetésére szolgál. Az aspektust fogalmilag is, formailag is a bejegyzés részének tekintjük.

A szótári bejgyzések több egymástól független szempont szerint osztályozhatók: a fogalmi szótárbeli létezésük módja, nyelvük, bonyolultságuk és indexelési értékük szerint.

Létezésük módja szerint vannak aktiv, a szótáradminisztrátor által szövegelemzésre, indexelésre kijelölt bejegyzések, és inaktiv a szövegelemzésből, indexelésből kizárt, azaz logikailag törölt bejegyzések.

A nyelvek szerinti osztályozás nem szorul magyarázatra. Az internacionális alakokat azonban az alkalmazó nyelvek szótáraiban meg kell ismételni. A különböző nyelvü, de azonos értelmü szavak /kifejezések/ szótári bejegyzéseinek kódja szigoruan azonos. Ez a kikötés biztositja a fogalmi kapcsolatrenszer nyelvüggetlen ábrázolhatóságát.

Bonyolultságuk szerint megkülönböztetünk egyszerü /általában egyszavas/ és összetett /általában többszavas/ bejegyzéseket. Az egyszerü alakok elsősorban a lexikai elemzés céljait szolgálják és a szavak tőjellegü alakjait adják meg, mig az összetettek lényeges támpontjai a szemantikai elemzésnek. A lexikai egységek tehát alacsonabb fogalmi szintet képviselnek és biztosan egyszerüek, mig a szemantikus egységek egyszerüek és összetettek egyaránt lehetnek.

Az indexelési érték szerinti osztályozás a szemantikus elemzés eszköze. Megkülönböztetünk deszkriptorokat, amelyek a szöveg tartalmi osztályozásához közvetlenül felhasználhatók, non-deszkriptorokat amelyek közvetve használhatók fel, és stoplista elemeket, amelyek a rendszer szempontjából indexelési értékkel /emberi döntés alapján/ nem birhatnak.

2./ A fogalomfeltáró szótár kapcsolatrendszere

A fogalmi szótárt kapcsolatai teszik strukturált halmazzá. A bejegyzések között feszülő kapcsolatok tipizálhatók, és két főtipusuk különböztethető meg. A felosztás azonban nem diszjunkt, vannak mind a két főtipusba beletartozó kapcsolatok. Az egyik főtipus a lexikai elemzéshez használható tipusokat foglalja magába, a másik a szemantikus elemzéshez használhatókat.

A fogalmi szótár bejegyzései közötti kapcsolatokat bináris relációknak tekintjük és azt mondjuk, hogy az A és B szótárbejegyzések R relációban vannak, ha A,B $\in$ FFSZ és a szótár biztositja az A $\xrightarrow{R}$ B átmenetet. /jelölve: A.R.B /.

A KFIIR fogalomfeltáró szótárának megengedett kapcsolatait az 1. sz. táblázat definiálja és foglalja össze.

3./ A fogalmi szótárbeli kapcsolatok tulajdonságai

A fogalmi szótár épitése közben egy uj kapcsolat felállitása szabályokhoz van kötve. A szabályok logikájának megértéséhez érdemes a kapcsolatok tulajdonságait külön megvizsgálni.

Reciprok kapcsolatpárok: a kapcsolatok között vannak olyanok, amelyeknek bevezetése automatikusan maga után vonja ellentétes párjuk /inverzük/ bevezetését is. Például, a BT-bővités NT-szükités kapcsolatokra az A.BT.B$\in$FFSZ állitás helyességéből következik a B.NT.A FSZ állitás helyessége. Az ilyen inverz párok teljes listája a következő.

2. sz. táblázat

Direkt kapcsolat	Inverz kapcsolat
BT	NT
USE	UF
SEE	SF
SNR	SNX
RT	RT
CPT	CT

Szimmetrikus kapcsolatok: azokat a kapcsolatokat, amelyekre az A.R.B $\in$FFSZ állitás helyességéből következik a B.R.A$\in$FFSZ állitás helyessége, szimmetrikusoknak nevezzük.
A fogalmi szótárban értelmezett kapcsolattipusok között csak a rokonitó /RT/ kapcsolat ilyen.

Tranzitiv kapcsolatok: Tegyük fel, hogy az A.R1.B$\in$FSZ és B.R2.C $\in$ FFSZ kapcsolatok fennállnak. Akkor az A és C szótári bejegyzések egymással is kapcsolatban állnak. Ezt az ujabb kapcsolatot nevezzük az R1 és R2 kapcsolatok szorzatának. A szorzatot R-rel jelölve R=R1.R2 irható. Ha A.R.C kapcsolat 1. sz. táblázatunk

1. sz. táblázat

A K A P C S O L A T

Csoportja	Neve	Tip. Lex.	Tip. Szem.	Jele	Alternativ jele	Értelmezése
Hierarchikus kapcsolatok	Bővebb fogalom		x	BT=BROADER TERM	B /bővitése/	A.BT.B: A bővitése B /B értelmezése tartalmazza A értelmezését/
	Szükebb fogalom		x	NT=NARROWER TERM	S /szükitése/	A.NT.B= A szükitése B /B értelmezése része A értelmezésének/
Utaló kapcsolatok	Használt mint	x	x	USE= REFERENTIAL USE	H /helyes alakja/	A.USE.B= az A helyes alakja B
		x	x	UF=USED FOR	IH /hibás alakja/	A.UF.B= az A hibás alakja B
	Lásd még	x	x	SEE=REFERENTIAL SEE	R /röviditès/	A.SEE.B=A röviditése B
		x	x	SF=SEEN FROM	IR /röviditettje/	A.SF.B=A röviditettje B-nek
Affinitiv kapcsolat	Rokonitás		x	RT=RELATED TERM	K /kapcsolható/	A.RT.B= A kapcsolható B-hez és viszont
Aspektus kapcsolatok	Aspektus hozzárendelés		x	SNR=SCOPE NOTE	A /aspektusa/	A.SNR.B= A aspektusa B
			x	SNX=SCOPE NOTE	IA /aspektus	A.SNX.B=A aspektus B-hez
Összetett-szókapcsolatok	Összetevője	x		CT=COMPONENT TERM	IO /összetevője/	A.CT.B=A összetevője a B-nek
	Összetevője	x		CPT=COMPOSITE TERM	O /összetettje/	A.CT.B=A összetettje a B-nek

szerint értelmes, akkor R-t aktiv-nak tekintjük, mig ha A.R.C önmagában nem értelmes, akkor R-t fiktiv kapcsolatnak nevezzük. A fiktiv kapcsolatok csak formálisan léteznek, értelmük nincs. Ezzel ellentétben aktiv szorzatok a kapcsolatokat értelmesen ugyanazon a reláción belül továbbviszik; ezt a tulajdonságukat tranzitivitásnak nevezzük.

A FFSZ aktiv szorzatai a következők:

3. sz. táblázat

BT.BT=BT	CPT.CPT=CPT
NT.NT=NT	CT.CT=CT
USE.BT=USE	NT.UF=UF

Az aktiv szorzatok felhasználásával tranzitiv kapcsolatláncok épithetők, amelyek eredménye feltétlenül helyes és értelmes kapcsolat marad.

Ellentmondásos kapcsolatok: szótárbejegyzések szorzatkapcsolaton keresztül értelmetlen relációkba állithatók. De nemcsak az ilyen kapcsolatok kerülendők,külön figyelmet kell forditani az 1.a ábra szerinti helyzetek elkerülésére is. Itt az A.BT.B.BT.C és a C.BT.A kapcsolatok nem lehetnek egyszerre igazak.

1. ábra

a/ Ellentmondásos kapcsolatlánc

b/ Redundáns kapcsolatlánc

Redundáns kapcsolatok: uj kapcsolatok bevitele nemcsak ellentmondásos, hanem felesleges is lehet. Ez akkor áll elő, ha az ujonnan bevinni kivánt kapcsolat mint szorzat aktiv. Az 1.b ábra példáján: az A.BT.B.BT.C kapcsolat aktiv volta miatt az FFSZ-nek a A.BT.C kapcsolattal való kiegészitése felesleges.

Ha a szótár egy állapota mentes az ellentmondásos és redundáns kapcsolatoktól akkor ezt a szótárállapotot konzisztensnek mondjuk.

A fogalmi szótár ellentmondás- és redundanciamentes épitésének szabályai

A szótárépitéshez szükséges szabályok megadásakor induljunk ki abból, hogy uj szótári bejegyzések bevezetése vagy a meglévők attributumainak módositása a szótár konzisztens voltán nem változtat. Hibás állapot igy csak helytelen kapcsolatépités követ-

kezménye lehet.

1/ Mivel egyszerre eredményez ellentmondást és redundanciát, tilos a szótárba az A.R.A. tipusu, önmagára utaló, kapcsolatok bevitele.

2/ Mivel két fogalom között ugyanazon kapcsolat bevitelének csak egyszer van értelme, az A.R.B és FFSZ tipusu kapcsolatok ismételt bevitele is helytelen. /A müködő FFSZ ezt megengedheti, de a kapcsolatrendszerre hatástalan. Az ismételt végrehajtás a kapcsolat megerősitéseként adminisztrálódik./

3/ Két különböző szótári bejegyzés között egyszerre csak egy aktiv kapcsolat létezhet. Kivétel: az SNR-aspektus reláció, amely második kapcsolatként is megengedhető.

4/ Hierarchikus és affinitiv kapcsolatot stoplista elemhez nem szabad definiálni.

5/ Utaló kapcsolatot csak deszkriptorok és nondeszkriptorok között szabad értelmezni.

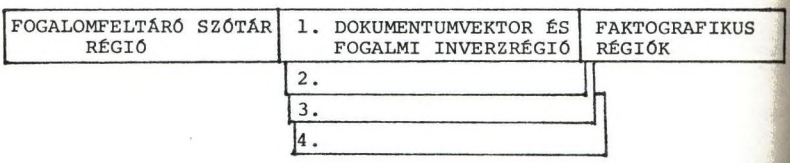
6/ Az uj kapcsolat bevezetése nem eredményezhet ellentmondásos vagy redundáns kapcsolatláncot. Mindkét esetben kapcsolat törléssel kell megelőzni a hibahelyzet kialakulását.

A fogalomfeltáró szótár és a dokumentációs adatkezelés viszonya

A fogalmi szótár IDMS eszközökkel kerül megvalósitásra. Az ismertetés megkönnyitésére nevezzük régiónak az azonos sémában leirt, fogalmilag összetartozó areá-k csoportját../A különböző régiókhoz tartozó areákat fizikai adatszervezési okokból nem célszerü egysiteni/.

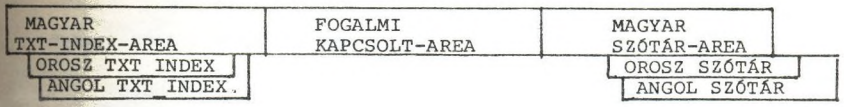
A KFIIR adatfeldolgozó rendszere olyan sémára támaszkodik, amelynek régiói a szöveges illetve a faktografikus adatfeldolgozó rendszer régióiból állnak. A fogalmi szótár ezen belül a szöveges tipusba ágyazódik. Ezt szemlélteti a 2. sz. ábra:

2. sz. ábra

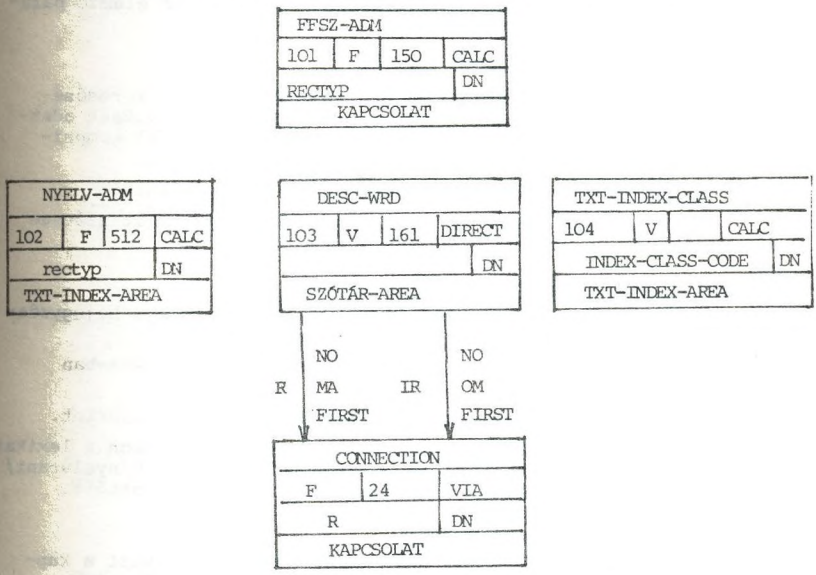


A KFIIR dokumentációs adatbázisában feltételezzük, hogy a fogalomfeltáró szótár olyan közös eszköz, amely egyszerre több dokumentációs adatbázis kiszolgálója. Erre utal a több rétegü dokumentumvektor és fogalmi inverzrégió.
A fogalomfeltáró szótár régiójának egy fokkal mélyebb szerkezetét a 3. sz. ábra mutatja.

3. sz. ábra



A fogalmi kapcsolatokat hordozó area a nemzetközi tezauruszajánlásoknak megfelelően közös az egyes nyelvi szótárak számára.
A fogalomfeltáró szótár IDMS-diagrammja, mint a szerkezet kifejtésének következő héja, a 4. sz. ábrán látható.



Az egyes rekordtipusok szerepe és legfontosabb tulajdonságai a következők.

FFSZ-ADM: fogalmi szótár adminisztrációs rekord:

- leirja a fogalomfeltáró szótár egy verziójának általános adatait, /egyetlen példálya létezik minden verzióban/;
- nyelvüggetlen, ezért a KPACSOLAT-AREA-ban található,
- lokalizációs módja CALC.

Általános szótárleiró adatoknak tekintjük a szótár nevét, verzió-számát, a verzió dátumát, az utolsó módositás dátumát, a módositó menetek számát, a laponkénti bejegyzésszámot, a szótár aktuális kapacitáshatárát, a következő kioszthatő adatbázis kulcsot, az aktuális fogalom kódszámlálót, a szöveges indexosztályok számát, a szótár tulajdonosának és felelősének nevét.

NYELV-ADM: nyelv adminisztrációs rekord:

- leirja a szótár egy konkrét nyelvi változatának nyelvészeti tulajdonságait;
- nyelvfüggő, ezért a TXT-INDEX-AREA-ben található,
- lokalizációs módja CALC;
- nyelvenként egy előfordulása van.

Nyelvészeti tulajdonságként tartjuk számon a nyelv nevét, a nyelvhez tartozó lexikai-elemző program/csomag/ nevét, az elemző által használt táblázatok könyvtárbeli neveit, az elemző paramétereinek defaul értékeit.

TXT-INDEX-CLASS: szöveg index rekord

- a szótárbejegyzések szövegük alapján történő kereséséhez szükséges index táblázat. Az egyes bejegyzések adatbázis kulcsát /kódját/ és szövegét sorolja fel komprimált formában értékpár táblázat szervezéssel.
- nyelvfüggő, ezért a TXT-INDEX-AREA-ban található;
- lokalizációs módja CALC, a bejegyzés komprimált szövege alapján.

DESC-WRD: fogalmi szótár bejgyzés rekord

- a szótár egy bejegyzésének hordozója, leirja a bejegyzésbeli szó/deszkriptor tulajdonságait.
- mivel nyelvfüggő, a nyelvhez tartozó SZÓTÁR-AREA-ban található.
- lokalizációs módja DIRECT a bejegyzés KÓD-ja szerint.

A szótár bejegyzések tulajdonságaként tartjuk számon a lexikai elemzéshez szükséges jellemzőket a bejegyzésbeli szótő/nyelvtani/ tipizálás kódjait, a szemantikai elemzés szabályazonositóit.

CONNECTION kapcsolatleiró rekord

- egy konkrét kapcsolat hordozója, amely megnevezi a kapcsolat tipusát /BT,NT...../ és a kapcsolat értelmezését segitő aspektust;
- A KAPCSOLAT areában található;
- lokalizációs módja VIA az előremutató R-set alapján.

6/ Jellmező adatok a szótárról

A hozzáférés gyorsasága becslés szerint az IS elérésmódhoz hasonló. A szótár helyigénye 2oo ooo bejegyzést és 6oo ooo kapcsolatot, a magyar nyelvet, 48 Byte-os átlagos bejegyzés hosszat és IBM-2314 diák tárolót feltételezve 14O cilinder, 19.8 MB.

Irodalomjegyzék

- MCNTI Pravila razrabotki mnogojazücsnüh informacionno-pois kovüh Tezaurusov, Moszkva, 1976
- 174-75 KGST szabvány: Egy nyelvü információkereső tezaurusz. "Tartalmi, formai, ábrázolási szabályok."

- Guidelines for establisment and development of multilingual scientific and technical thesauri for information refriveal. UNESCO SC/WS/50. Paris 1971

- A.Chepkasov: INIS: Thesaurus Maintenance System International Atomic Energy Agency, Vienna, 1977

- Varga Dénes: Információs tezauruszok készitésének módszertana OMKDK, Budapest, 1969.

- Horváth Tibor - Varga Dénes: Információs tezauruszok, Budapest, 1978.

Lőrincz István
Honvédelmi Minisztérium

Lőrincz János
OMFB KFIIR Kidolgozó Csoport

Több lépéses duális clusterezési algoritmus alkalmazása információs rendszerekben

Bevezető

A szöveges információs rendszerek szempontjából alapvető fontosságu a dokumentumok és a fogalmak clusterezése. A clusterezések végrehajthatók manuálisan vagy automatikusan. A manuálisan végzett clusterezés nagyon munkaigényes és ezért nehezen tudja követni a dokumentumok és fogalmak jellegében mutatkozó változásokat. Az automatikus clusterezés teljeskörü végrehajtása még számitógéppel is jelentős ráforditásokat igényel. Ugyanakkor a visszakeresés teljessége és pontossága is csak akkor lesz elfogadható ha a clusterezést a kérdésből kiindulva hajtjuk végre.

A clusterezési eljárással szemben támasztott követelmények a következők:

- legyen korrektül deffiniált, vagyis minden dokumentum vagy fogalom halmazon adjon eredményt;
- az eredmény legyen független az input adatok sorrendjétől;
- az eljárás legyen stabil, vagyis a kiinduló adatok jelentéktelen változása csak jelentéktelen mértékben módositsa az eredményt;
- az eljárás legyen független a léptéktől, vagyis az objektumok koordinátáinak konstans értékkel történő szorzása nem változtathatja meg az eredményt;
- nagyon hasonló tulajdonságu dokumentumok nem kerülhetnek különböző clusterekbe.

Az általunk választott clusterező eljárás a kérdésből indul ki

-/ez biztositja a rugalmasságot/ -

a teljes dokumentum állomány statisztikai tulajdonságai alapján

-/ez biztositja a stabilságot/ -

két osztályba a releváns és nem releváns dokumentumok osztályába

sorolja a dokumentumokat. Az algoritmus többlépéses végrehajtása kényelmes eszközt biztosit a válasz pontosságának és teljességének hangolására.

Dokumentum - Dokumentum tér

Az előbbiekben vázolt problémák vizsgálatához vezessük be a következő fogalmakat - definiciókat.

A KFIIR adatfeldolgozó rendszerében dokumentumnak számit a nyilvántartás meghatározott tárgyköreinek kijelölt szöveges kódolásu adataiból/rovataiból álló minden rekord. A dokumentumok jele: d.

A dokumentumok nyilvántartott halmazát dokumentum térnek nevezzük és D-vel jelöljük.

A dokumentumokra és a dokumentum térre igazak a következő jellemzők:

D1. A dokumentumoknak azonos nyelvü rovatok előfordulásából kell állnia.

D2. A dokumentumok minden egyes rovatának az eredeti rekordtipusa által meghatározott és a rendszer adatmodelljében rögzitett sulyt tulajdonitunk.

D3. A dokumentumok egymás között eltérő nyelvüek lehetnek, ezért a nyelv a dokumentum minősitője.

D4. A dokumentum térben az egyes dokumentumok egyértelmü /numerikus/ azonositóval rendelkeznek.

Tételezzük fel, hogy képesek vagyunk a dokumentumok rovatbontásán tul a dokumentumok fogalmainak meghatározására is. A dokumentum térben előforduló összes fogalmak halmazát fogalomtérnek nevezzük.

T1. A fogalomtérben az egyes fogalmak egyértelmü azonositóval rendelkeznek.

T2. A fogalomtér nyelvfüggetlen. A fogalmak tényleges előfordulásának nyelve nem a fogalmak minősitője, hanem az adott dokumentumoké.

Tehát a dokumentumtér alatt az egyes dokumentumokban előforduló fogalmak azonositóiból képzett vektorok összességét értjük.

$D=\{\bar{D}_j\}$ ahol $\bar{D}_j=(f_{j1}, f_{j2} \ldots\ldots f_{jt_F})$ és

$$f_{ji}=\begin{cases} 0 & \text{ha az } f_i \text{ azonositóju fogalom nem fordul elő a j-edik dokumentumban} \\ s_{ji} & \text{az } f_i \text{ fogalom sulya a j-edik dokumentumban} \end{cases}$$

Hasonló módon fogalomtér alatt az egyes fogalmak dokumentumtérbeli előfordulásainak azonositóiból képzett vektorok összességét értjük.

$F=\{\bar{F}_i\}$ ahol $\bar{F}_i=(d_{i1}, d_{i2}, \ldots\ldots d_{it_D})$ és

$$d_{ij}=\begin{cases} 0 & \text{ha az } f_i \text{ azonositóju fogalom nem fordul elő a j-edik dokumentumban} \\ s_{ji} & \text{az } f_i \text{ fogalom } D_j \text{ azonositóju dokumentumban való előfordulásának sulya.} \end{cases}$$

Mindkét téren értelmezünk egy távolságfüggvényt

$$R(D_k, D_j)=\frac{\sum_{i=1}^{t_F}(s_{ki}\cdot s_{ji})}{\sqrt{\sum_{i=1}^{t_F}(s_{ki})^2\cdot\sum_{i=1}^{t_F}(s_{ji})^2}}$$

valamint

$$R(F_i, F_i)=\frac{\sum_{j=1}^{t_D}(s_{ji}\cdot s_{ji})}{\sqrt{\sum_{j=1}^{t_D}(s_{ji})^2\cdot\sum_{j=1}^{t_D}(s_{ji})^2}}$$

Ezeket relevancia együtthatóknak nevezzük, segitségükkel bevezethetők mindkét téren a "környezet" a "pont sürüségi" fogalmak.

Tovább menve a keresőkérdéseket is tekintsük dokumentumnak, amelyeket - normalizáljuk és ideiglenesen szintén a dokumentum térbe illesztünk. Egy ilyen kérdés dokumentumhoz /alkalmas hasonlósági kritérium alapján/ meghatározhatjuk a hozzá legközelebb eső dokumentumok halmazát, amit a hasonló dokumentumok "nulladik cluster"-ének nevezhetünk. A 0-clusterbeli dokumentumok azonositói, mint a fogalomtérbeli koordinátaértékek egy pontot határoznak meg a fogalomtérben. A távolságfüggvény /és alkalmas fogalmi rokonsági

kritérium/ segitségével meghatározhatjuk a fogalmak egy másik, fogalomtérbeli clusterét,amelyek ehhez a ponthoz "legközelebb" helyezkednek el. Ezen fogalmak azonositói, mint a dokumentumtér koordinátaértékei egy pontot definiálnak a dokumentumtérben. Azt a pontot a "pontositott" kérdésnek nevezhetjük, az eddigi tevékenységet pedig a kérdés "értelmezésének" vagy pontositásának.

A távolságfüggvény segitségével meghatározhatjuk a pontositott kérdéshez legközelebb eső dokumentumok halmazát.

Ezt a halmazt nevezhetjük a dokumentumok "első cluster"-ének. A pontositási lépés addig ismételhető amig az előirt tulajdonságokkal rendelkező clustert előállitjuk.

Az ismertetésre kerülő algoritmust természetes nyelvü dokumentumok tartalmi hasonlóságának elemzésére használjuk a KFIIR-ben. A kifejlesztett programcsomag ESZER számitógépen OS/VS1 operációs rendszer alatt hatékonyan és megfelelő pontossággal működik.

Az ismertetett eljárást assembler nyelvü keretprogram hajtja végre. A minden lényeges funkciót külön rutin valósit meg. A fejlesztés során a problémás rutinokból több verziót készitettünk el. Ezeket cserélgetve kisérlet sorozatok során hangoltuk rá a rendszerünket a dokumentumterünkben tárolt rendkivül szerteágazó tematikáju szövegekből történő visszakeresésre. Ilyen problémás rutinok:

- clusterek határainak meghatározása,
- a sulyok képzése,
- a pontositás konvergenciája,
- a zaj leválasztása. Ha van jó válasz könnyebb. Ha nincs jó válasz nagyon nehéz.

A KFIIR hazai kutatási+fejlesztési témák leirásait feldolgozó alrendszerében jelenleg 25.ooo dokumentumot tárolunk. Ezek mindegyike 15o-2oo fogalmat tartalmaz. Az összes előfordult szótövek száma kb. 12o ooo ezeket képeztük le a kb. 2o ooo fogalmat tartalmazó fogalomtérre. Egy kérdés egyszeri pontositással történő megválaszolása 35-4O másodperc CPU időt használ fel. Minden további pontositás 2o-25 másodpercet igényel.

A rendszer fejlesztését a leképzés tökéletesitésével, valamint a problémás rutinok további verzióinak kipróbálásával folytatjuk.

Irodalom

1. Gerard Salton: Dynamic Information and Library Processing

Lőrincz István
Honvédelmi Minisztérium

Tóth Lajos
OMFB KFIIR Kidolgozó Csoport

ETKR: egy általános gráfkezelő rendszer

I. Ha egy számitógépes rendszer ugynevezett emberi tulajdonságait próbálnánk mérni, intelligenciájának kétségtelenül alkotó része lenne a rendszer alkalmazkodóképessége a valós környezethez, alakithatósága a megváltozott viszonyokhoz. Az is nyilvánvaló, hogy változó ismereteket nem lehet és nem is szabad beleépiteni a rendszer egyetlen programjába se. Ilyen leválasztott tudás pl. modern forditó-értelmező rendszerekben a szótár, illetve szótőtár, amelyhez egy csatolórutin biztositja a hozzáférést. Az ilyen cserélhető szótárak lehetővé teszik pl. egy rendszer speciális szakterületre leszükitett, viszont ezen területen szinte hibamentes alkalmazását. A fent vázolt probléma felmerült a KFIIR rendszer kifejlesztése során is. Itt is több helyen szükség volt olyan tudás reprezentálására, amely a rendszertől függetlenül létrehozható illetve aktualizálható. Ezt a tudást csatolórutinok teszik hozzáférhetővé a rendszer tetszőleges elemei számára.

Ezeknek az igényeknek a kielégitésére született meg az az általános gráfkezelő rendszer, amely most előadásunk tárgya. Mivel a rendszer elsősorban bizonyos triviális visszavezetések és ellenőrzések céljára létrehozott ugynevezett ellenőrző táblák kezelésére jött létre, illetve erre lett elsődlegesen alkalmazva, a neve Ellenőrző Tábla Kezelő Rendszer, röviden ETKR.

II.1. Az ETKR tervezésekor különböző igények merültek fel, mind formai, mind funkcionális szempontból. Ezek közül nézzünk néhányat.

1/ Többféle tartalmu, funkcióju ellenőrző táblákat /gráfokat/ létre tudjon hozni, ezeket tárolja egy erre a célra létesitett könyvtárban és innek vissza tudja tölteni a memóriába;

2/ Alapvető igény a gyorsaság szempontjából, hogy használatához a táblának teljes mértékben a memóriában kell lennie.

3/ A táblák elemei tetszőleges felépitésüek lehessenek, amelyet a felhasználó által megadható rutin kezel. Alapértelmezésben a táblaelemeket a rendszer karaktersorozatként kezeli.

4/ Az elemek között többféle kapcsolatot lehessen definiálni, irányitva vagy irányitás nélkül, és ezeket a kapcsolatokat mnemonikkal is lehessen jelölni. Ezek lesznek a gráf "szinezett" élei.

5/ A gyors elemeléréshez meg lehessen adni egy hash-kód képző függvényt /szintén opcionálisan/, amely az elemek osztálybasorolását végzi.

6/ A tábla memóriabeli megszoritásoktól eltekintve tetszőlegesen bővithető legyen a memóriában mind elemekkel, mind köztük definiált kapcsolatokkal.

III.2. Nyilvánvaló, hogy a tábla nem lesz egységes szerkezetü, hiszen optimális helykihasználást fix hosszuságu elemrekeszek nem tesznek lehetővé, másrészt a hashelés miatt egy ugynevezett hash-táblát mindenképpen külön kell választani a tábla más részeitől. Ugyanakkor az is természetes, hogy minden táblának tartalmazni kell speciális, csak rá jellemző információkat, pl. a hash-osztályok száma, a táblában definiált kapcsolatmnemonikok, a felhasználó speciális rutinjainak nevei, stb. Még azt vegyük figyelembe, hogy az elemek között elvileg korlátlanul definiálható kapcsolatok miatt a gráfok csucspontjait és éleit el kell különiteni egymástól.

A kezelő rendszerre vonatkozóan alapvetően azt kell szem előtt tartani, hogy bár a táblákat más rendszer fogja alkalmazni, mint amelyik létrehozza és aktualizálja, a müködés alapelemei, az elemi gráfkezelő funkciókat ellátó primitivek azonosak kell, hogy legyenek.

III.3. Végül az alkalmazói és gépi korlátok adta megszoritások által szabott mederben a rendszer a következő formát öltötte:

1. Táblák; könyvtárbeli formájuk: Minden táblának alapvetően két formája létezik. Egy memóriabeli ábrázolás és egy könyvtárbeli ábrázolás. Azért lehet és kell ezt a két formát szétválasztani, mert a két tárolási formának teljesen más a funkciója. A memó-

riában a gyors hozzáférést kell biztositani minden elemhez, ugyanakkor a tábláknak rugalmasan bővithetőknek kell lenniük. A könyvtárbeli táblánál az a legfontosabb szempont, hogy a háttértárolóra való kivitel és az onnan való beolvasás között a tábla egyes részeire nem történik külön hivatkozás. Tehát a belső ábrázolástól teljesen eltérő formát alakithatunk ki.

Alapvetően háromféle rekordkép létezik a könyvtárbeli tárolásnál. Az első fajta rekord a tábla fejrészének információit, a második fajta rekord az egyes elemeket és a hozzájuk tartozó kapcsolatláncot, mig a harmadik a hashtáblát tárolja. A könyvtárban természetesen abszolut cimek helyett mindenhol relativ cimek szerepelnek.

2. A táblák tárbeli felépitése: Minden tábla legfeljebb négy, a memóriában függetlenül elhelyezkedő mezőből áll, melyeket az első mező, a táblafej alapján lehet elérni. Itt tárolódnak a táblát jellemző adatok, az egyes mezők cimei, hosszuk, a definiált kapcsolatok mnemonikjai, kódjai, stb.

A másik mező amelyet mindenképpen tartalmaznia kell a táblának, az elem-mező. Ez az egyetlen olyan mező, amelynél nincsen fix szerkezet, eltekintve persze a szükséges mutatóktól. Az itt levő elemek tetszőleges szerkezettel rendelkezhetnek.

Az ugynevezett pointer-mező az, amely az elemek közötti kapcsolatokat tartalmazza, azaz a gráf éleit reprezentálja, mutatók formájában. Minden élnek egy bejegyzés felel meg, ha kétirányu a kapcsolat, akkor kettő.

Ha nincs megadva a táblához pointer-mező, akkor egy egyszerü hash-elérésü elemtárat kaptunk.

Az utolsó mező a hash-mező, amely szintén opcionális. Hiánya esetén az elemek listába lesznek felfüzve, ugyanugy, mintha hashelő függvényünk egy osztályba sorolná az elemeket. Ez a mező annyi elemü, ahány hash-osztályba soroljuk az elemeket, és egyes elemei vagy üres mutatót /null-pointert/ vagy az adott lánc első elemének cimét tartalmazzák.

3. A rendszer felépitése.
A táblák létrehozására és aktualizálására /update/ alapvetően egy keretrendszer szolgál, melv közbenső csatoló rutinokon keresztül, a rendszer primitivjeivel végzi mű-

ködését. A keretrendszer képes táblák létrehozására, bennük kapcsolatfajta definiálásra, törlésre, illetve elemek beépitésére, illetve törlésére. Ezenkivül az elemek mint csucspontok között élek /konkrét kapcsolat/ létrehozására illetve megszüntetésére.

A rendszer alapelemei, a primitivek, alapvető gráfkezelő funkciókat valósitanak meg, mint például egy csucspont vagy él definiálása, illetve megszüntetése, egy csucspontból kiindulva egy-egy lépés megtétele a gráfban adott szinü él mentén stb.

A rendszerben használt primitivek természetesen tetszőleges programból aktivizálhatók, a megfelelő környezet kialakitása révén: ez a környezet egy szabvány felépitésü paramétertábla, amelyen keresztül a rendszerelemek messzemenően képesek egymással kommunikálni. Igy az egyetlen paraméter a paramétertábla cime, amelyet a rendszer komponensei átadnak egymásnak.

4. A felhasználó által irt rutinok is ezt az egyetlen paramétert használják. Egy táblához három ilyen, ugynevezett "user-rutin" adható meg, melyeket a primitivek képesek aktivizálni.

Az első rutin melyet kötelező megadni, két tetszőleges táblabeli elem azonosságát kelll, hogy eldöntse, természetesen ismerve az elemszerkezetet.

A második rutin végzi a hash kód képzését.

A harmadik felhasználói rutin segitségével megadható az elem ellenőrzése, konvertálása, illetve válogatni lehet az elemek közül /hasonlóan a SORT eljárások rekordmódositó felhasználói eljárásaihoz/. Igy pl. ez a rutin végezheti az elem konvertálását a külső /nyomtatási/ illetve belső /memóriabeli/ alak között, illetve a táblakönyvtárba való kivitel előtt, illetve visszahozatal után.

II/4. Ezek után nyilvánvaló, hogy rendszerünk ellenőrző táblákon kivül, sokkal szélesebb körben alkalmazható. Vegyük például a néhány éve nagyon divatos szemantikai hálók problémáját. Az ilyen ugynevezett mélyszerkezeti strukturák számitógépes megvalósitása nagyban hozzájárul gyakorlatban történő alkalmazásukhoz és ezzel az elmélet fejlődéséhez. Az ETKR tulajdonképpen egy kész eszköz ilyen jellegü próbálkozások felé, hiszen csupán az elemszerkezetet ismerő felhasználói rutinokat kell

elkésziteni az elmélet létrejötte után és már gyakorlatban lehet a további lépéseket megtenni.

Könnyüszerrel átültethetők az ETKR nyelvére és segitségével kezelhetők bizonyos irányitó-vezérlő tevékenységhez tervezett parancs-paraméter táblák is.

Mindössze a rendszer szempontjából szükséges felhasználói rutinokat és csatolórutinokat kell elkésziteni a primitivek használatát megkönnyitő különböző alkalmazásoknál. Ez utóbbihoz makrókészlet áll a felhasználó rendelkezésére.

III. Az ETKR rendszer használata /alkalmazásai/

A jelenleg megvalósitott rendszer a KFIIR információs rendszerünk szerves része, mely PLIOPT és 370-es ASSEMBLERben iródott meg, ESZR gépeken OS/VS1 operációs rendszerben. Mivel a kezelt ellenőrző táblákat a memóriában tárolja, nagymértékben növeli a feldolgozás sebességét.

A felhasználás során egyidejüleg öt táblát használunk, melyeknek betöltése másfél- két perc, helyfoglalásuk pedig több száz Kbyte. Ezek a táblák egyrészt fastrukturájuak, egyszerü hashelést tartalmaznak, másrészt többféle relációt tartalmazó un. szegmensellenőrző táblák.
Ez utóbbiakban lehetőség van több rovat tartalmának együttes ellenőrzésére, adatok korrekciójára, illetve pótlására hiányos vagy hibás bejelentés esetén. Az ETKR rendszer helyfoglalása attól függően, hogy milyen mértékben használjuk a rendszerelemeket 50-200 Kbyte között van. Jelenleg kb 10 primitivvel és kb ugyanannyi csatoló rutinnal rendelkezik.

Zvara Flórián
Honvédelmi Minisztérium
Budapest
1052. Pf. 117.

DINAMIKUS LISTÁZÓ

A KFIIR rendszerben, mint szöveges információs rendszerben különleges listakészitési gondok merültek fel. A gondok a szöveges adat és formátum elemek nagy számából, az adatelemek terjedelembeli változékonyságából, a feldolgozandó rekordok adatelem állományának egy elvi maximum alatti rendkivüli mobilitásából, több forrásból való összeszerkeszthetőség igényéből, valamint a sokszorositható /kézbe adható/ listatermék követelményéből származtak.
Célunk volt egy olyan listázó nyelv és listázó program kifejlesztése, amellyel könnyen áttekinthető lista készithető ugy, hogy közben minél nagyobb szabadsági fokkal lehessen azt megtervezni és leirni.
Listázási nyelvnek azt a makro-assembler nyelven irt eszközt tekintjük, ami a listakép kódolására szolgál.
A listatervezés és a listaleirás elkészitése a teljes listázási folyamatnak egy önálló szakaszát jelenti, és teljesen elválasztható a tényleges listázástól /nyomtatástól/.
Célja a listaleirás létrehozása.
A listaleirás tartalmazza mindazokat az információkat és adatokat, amik a listakimenet külső formáját irják le. Egy listaleirás csupán egyetlen listakimenet formátum létrehozására szolgál. Ezáltal annyi listaleirást kell készitenі, ahány féle formáju listát elő szeretnénk állitani.

A listaleirásoknak forrás és load modul formájuk létezik, amelyek a szokásos könyvtárkezelő eszközökkel tarthatók karban. A könyvtárazás egyuttal kényelmes lehetőséget nyujt egy "lista-menü" kialakitására, illetve listasorozatokkal operáló összetettebb szolgáltatások szervezésére.
Az alkalmazott listaleirás lényeges vonása, hogy elsősorban az eredményt irja le, az input rekordokra vonatkozóan elvileg semmilyen megszoritást nem tesz. Az inputtal való kapcsolat leirására kizárólag egyetlen eszköz, a nyomtatott adatok neveinek megadása szolgál.
Az implementált listázó változat a KFIIR rekordszervezési szabványának megfelelően az ISO-2709/1971 /KGST NME-2/ előirást használja. De mivel a rekord-szerkezet jellemzőit csupán egy adatkezelő modul tartalmazza, és azt is kizárólag az adatnevek segitségével vezérli, a modul kicserélésével egyszerüen áttérhetünk más rekordszerkezetre.
A listázó további jellemzői:

- egyszerre több rekordból történhet a listázás, s az egyes rekordok adatait párosithatjuk egymással; csoportosithatjuk őket.
  Kiiratásukat feltételekhez köthetjük. A feltételek megadásával az alábbiakat valósithatjuk meg:
  . a csoport elemeit csak akkor nyomtatjuk ki, ha az első elem nem üres /azaz tartalmaz kinyomtatandó adatot/.
  . ha a csoportnak akár egyetlen eleme üres, akkor egyik elemet sem listázzuk ki,
  . a csoport első elemét csak akkor listázzuk, ha a többi eleme között van nem üres.
- a listára kinyomtathatók előre megadott konkrét szövegek is /amiket nem tartalmaznak a rekordok/.
  Ezek a szövegek - listázási szempontból - egyenértékűek a rekordok adatelemeivel. Használhatjuk őket a rekordbeli adatelemek feliratozására, megyarázatára stb.

- automatikus függőleges irányu tömörités.
  A hiányzó adatok helyét nem hagyjuk üresen a listán, hanem az alákerülő adatokat felfelé toljuk.
- kérhető átmozgathatóság is.
  Ekkor a program kis mértékben megváltoztatja a megtervezett lista képét a nyomtatásra váró adatok számának és méretének megfelelően, ugy, hogy az tömörebb legyen.
- lapszerkesztési lehetőségek:
  Lapdobást irhatunk elő ill. tilthatunk meg.
  A/4-es nagyságu listatervek esetén kérhetjük a lapok egymás mellé ill. egymás alá történő nyomtatását.
  Soremelést adhatunk meg az adatelemek /ill. azok csoportjai/ előtt.
- a sorokra történő tördelés és a szavak soron belüli elhelyezkedése paraméterezhető /jobbra-balra sorkizárás/.
- lehetőség van kiemelésre többszörös ráütéssel. **
- lehetőség van speciális rutinokkal egyedi problémák megoldására, azaz olyan listakép kialakitására, amely a rendelkezésre álló eszközökkel nem érhető el másként.
- külső fejléc rutin csatlakoztatható a listázóhoz, amellyel a mindenkori igénynek megfelelő fejléc alakitható ki.

A listázó Assembler nyelven irődott. Gyorsasága elsősorban a listaleirás nagyságától függ. A/3-as méretü listatervek esetén egy-egy lap 1 sec-nél kevesebb CPU idő alatt készül el.

** Kiemelhetnek egy adatelemet, vagy pedig [illegible]

dr. Szőke Péter
OMFB KFIIR Kidolgozó Csoport

Baranyai István
Honvédelmi Minisztérium

A KFIIR /általános szöveges-faktografikus információ kezelő rendszerének/ általános programeszközei

A Kutatás-Fejlesztési Irányitási Információs Rendszer fejlesztése során az [1] -ben ismertetett okok miatt törekednünk kellett arra, hogy programeszközeink a lehető legáltalánosabbak legyenek, a különféle követelmény-változásokkal szemben stabilitást mutassanak.

Igyekeztünk ennek megfelelően rendszerünket olyan strukturában megvalósitani, mely módot ad a többcélu alkalmazásra, egy alkalmazáson belül a követelmények változásának követésére, csatlakozásra a nemzetközi információcseréhez, biztositja a hardware hibákkal szembeni kellő megbizhatóságot.

Ezeket a célokat az előadásban ismertetett szervezési módszerekkel és koncepciókkal valósitottuk meg.

1. Módszerek

1.1. Általános, specializálható, de gyors folyamatok szervezése

Az egész rendszert folyamatok összességeként fogjuk fel, melyek lehetnek ciklikusak is. Az egyes folyamatok nagyobb, logikailag összetartozó feldolgozási lépéseket valósitanak meg.

Az egyes folyamatokat jobok sorozata alkotja. Az egyes jobok közötti kapcsolat

- . file-okon keresztül
- . visszatérési kódban
- . üzemeltetői ellenőrzés/beavatkozás

formájában valósul meg.

Az, hogy a folyamatokat sok job alkotja, lehetőséget nyujt a számitógép hardware bizonytalanságainak a leküzdésére, mert a hibásan végrehajtott jobokat meg lehet ismételni.

Az egyes jobok /programok/ szerkezete olyan, hogy a lényegi, minden alkalmazásra vonatkozó, és a külső követelmények változásától nem függő funkciókat a programokba beépitettük, mig az adott alkalmazástól függő speciális ismereteket és feladatokat jól elkülönitett modulok tartalmazzák ill. oldják meg. Ennek megfelelően az általános programok önmagukban csak minimális tevékenységet

/mint pl. üres tevékenység vagy file-másolás/ valósitanak meg, és biztositják potenciálisan korlátlan számu speciális /valódi feldolgozó/ tevékenység csatlakoztatását.

1.2. Nemzetközileg elfogadott adatábrázolási előirás

A KFIIR rendszerben tárolt adatokat elsődlegesen a [2] -ben definiált NME2 formátumban tároljuk, mely a nemzetközi műszaki információcsere alapvető formája. Ennek a Müszaki Előirásnak az ISO-megfelelője a [3]. Müszaki információcserében széles körben használják a [2] -n alapuló MEKOF-2 ill. MEKOF-1 Normativ-Müszaki előirást, mely definiálja az egyezményes megengedett adatelemeket, és azok belső azonositóit /"hivójeleit"/ is. E két előirásnak a mi ábrázolásunk nem tesz eleget, elsősorban, mert az általunk tárolt információk csak részben esnek egybe azokkal az adatokkal, melyre a MEKOF előirásokat létrehozták. Nem okoz azonban nehézséget a MEKOF előirásokra történő konvertálás vagy ilyen anyagok átalakitása a KFIIR szellemében.

Lényeges különbség, hogy a KFIIR rendszerben az egyes hivójelek nincsenek örökre definiálva, hanem minden egyes alkalmazásban, vagy akár az idők során egy alkalmazáson belül egy adatelem-azonositó jelentése változtatható, mint ahogy változtatható az adatelemmel kapcsolatban elvégzendő ellenőrző-feldolgozó tevékenység is.

1.3. Adatfüggetlenség biztositása adatkezelő modulok segitségével

Az elemi adatok kezelésére külön adatkezelő programmodulokat alakitottunk ki, az adatokhoz történő hozzáférés csak ezek segitségével történik. Igaz ez az NME2 formátumu, az invertált, szótár- és segédfilejainkra, ellenőrző táblázatainkra egyaránt.

Ez megfelel a modern "információ-elrejtési" elvnek, és lehetőséget nyujt arra, hogy szükség esetén az elemi adatábrázolást megváltoztathassuk anélkül, hogy a feldolgozó programokban változtatni kellene. Ehhez csak azt kell biztositani, hogy az uj szerkezet szerinti adatkezelő eljárások felhasználói képe a régivel azonos legyen.

1.4. Speciális információk és funkciók elválasztása, azok algoritmikus megadása

Már az I. pontban utaltunk arra az alapelvre, amit mi "a keretprogramok elvének" nevezünk, éspedig, hogy a folyamatok vezérprogramjai csak olyan általános funkciókat valósitanak meg, melyek adott információhalmaz sajátosságaitól nem függenek.

Egy keretprogram önmagában még nem kész eszköze az információs rendszernek, hanem azt ki kell egésziteni az adott információhalmaz sajátosságait tükröző részekkel. Ezek megadják a halmaz lehetséges adatelemeit, az azokkal kapcsolatos néhány általános tulajdonságot, azok ellenőrzési szempontjait, származtatott adatok esetében a származtatás módját stb.

Lényeges tulajdonsága a rendszernek, hogy nem szab korlátot a lehetséges adatelem-fajták mennyiségének, és a köztük fennálló kapcsolatoknak, illetőleg a lehetséges ellenőrzések bonyolultságának sem.

Ilyen nagy adaptálhatóság célul történő kitüzésekor viszont felmerült az a probléma, hogy nehéz lenne olyan speciális adatle-

iró nyelvet kialakitani, mely tükrözné mindazokat a tulajdonságokat, lehetséges feldolgozási módokat, amelyek az adatokkal kapcsolatban egyáltalán előfordulhatnak. Egy ilyen nyelvvel kapcsolatban két fő probléma lépne fel:

- Nem lenne teljes, idővel óhatatlanul felmerülnének ujabb, korábban figyelembe nem vett követelmények;
- Nagy munkát jelentene a nyelv implementálása, melynek egy jelentős része kárbaveszne: sok olyan tulajdonságot kellene általános formában megvalósitani, melyeket konkrétan nem, vagy csak kis részben hasznositanánk. Ez kihatna az egész rendszer hatékonyságára is, mert az általános algoritmusok általában kevésbé hatékonyak, mint a konkrét esetre alkalmazottak. Láthatunk erre elegendő példát az univerzális programozási nyelvek implementációi körében is.

Ezek alapján ugy döntöttünk, hogy a konkrét /KFIIR rendszerre jellemző/ sajátosságokat nagyobbrészt algoritmikusan, univerzális programozási nyelvek eszközeivel fejezzük ki. Ennek megfelelően a rendszerben tárolt minden egyes konkrét vagy potenciális, inputból eredő vagy származtatott adatelemmel kapcsolatba hoztunk egy eljárást, mely az adatelemmel kapcsolatos ellenőrző/generáló funkciókat valósitja meg. Megengedjük az adatelemek ismétlődését, ebben az esetben azonos eljárást asszociálunk az összes ismétlődéssel. Ha több adatelemnek együttesen kell valamely kritériumnak eleget tennie, akkor ezeket együtt egy potenciális, aktuálisan meg nem jelenő adatelemnek tekintjük, /adatszegmens/, és egy csoportos ellenőrzést/generálást megvalósitó eljárást asszociálunk vele.

A keretprogramok számára azt kellett még biztositani, hogy össze lehessen őket kapcsolni tetszőleges számu eljárással, meg lehessen adni nekik tetszőleges számu adatelemet. Ez többféle technikával is megvalósitható. Minden esetben a végleges programot egy hármas alkotja: Keretprogram + adatleirás + eljárásgyüjtemény.

a/ Preprocesszoros technika.

A leirást a PLIOPT preprocesszor értelmezi; a keretprogramból és a megadott eljárásgyüjteményből egy specializált PL/I nyelvü forrásprogramot kompilál. Ugyanez a módszer alkalmazható PL/I és Preprocesszor helyett Assembly és Makrogenerátorral is.

b/ Rutinkapcsolásos technika

A leirás egy táblázat, melyet külön erre szolgáló eszközzel generálunk /Assembly makrók/. A futó keretprogram a leirást értelmezi /triviális értelmezés/, és a linkage editor által a táblázathoz csatlakoztatott eljárásokat aktivizálja, majd azok vége után értelmezi az egyezményes visszatérési kódokat. A specializált program csak bináris kód alakjában áll elő, a linkage editor hozza létre.

1.5. Programok moduláris felépitése, strukturált programozás

A modularitásnak és strukturált programozásnak esetünkben igen nagy szerepe van, mivel a környezetben beállott változásokat általában a speciális tulajdonságokat tükröző eljárásgyüjtemény változtatásával /programmódositással/ kell követni. Ezért eljárásaink kicsik, a különböző eljárások által használt tevékenységek /adatkezelés, hibaüzenetek, szövegkezelés stb/ közös modulok belépési pontjaként vannak implementálva, a névkonvenciókat könyvtári forrásprogram-szeletek /INCLUDE-tagok/ használata könnyiti meg, a forrásprogramok strukturáltan, ember számára könnyen olvasható alakban készülnek.

2. A KFIIR rendszer folyamatai

Az alább ismertetendő folyamatok mindegyikát a fenti "általános program - speciális tulajdonságok algoritmikus megadása - specializálás - késztermék-használat" koncepció alapján hoztuk létre. Megközelitésünk természetesen nem előzmény nélküli, igy pl. az IBM Sort/Merge utility-je is variálható adatbeviteli és kiiratási felhasználói eljárások csatlakoztatásával, és hasonló lehetőségeket találhatunk egyes adatbázis-kezelő rendszerekben is. Nálunk azonban ez a koncepció lényegét képezi: minden speciális ismeretet az eszközspecializálás során adunk meg, a legáltalánosabb /adatelemenként legfeljebb néhány bittel ábrázolható/ tulajdonságokat kivéve mindent algoritmusban, és az igy csatlakoztatott felhasználói eljárások a folyamatok legtöbbjében tetszőleges számuak lehetnek. /Természetesen egyszerübb folyamatoknál erre nincs mindig szükség/.

A KFIIR rendszer az alábbi főbb folyamatokból áll:

- adatfelvitel; karbantartás szövegnormalizálással; rendezések; szótárkezelés; invertált file kezelés; hálós kapcsolatok gráfreprezentációjának kezelése; szöveges- ill. tényadat-visszakeresés; dinamikus szövegszerkesztés és listázás; leválogatások faktografikus kritériumok szerint; szöveges hasonlóság-vizsgálat.

3. Komponens-osztályok

. a folyamatok keretprogramjai
. speciális ismereteket megtestesitő un. "felhasználói eljárások"
. a keretprogramok és az eljárások kapcsolatát biztositó tábla; az ezzel kapcsolatos apró, de fontos assembly eszközök;
. közös adat- ill. szövegkezelő modulok.

3.1. A felhasználói eljárások

Két kategóriába sorolhatók:

- egy keretprogram általános, adatelemektől független bővitményei. Ilyenek lehetnek pl. a cimkefeldolgozó ill. rekordválogató eljárások, parancselemzés,hash algoritmus, tábla-

elemek ekvivalencia-definiciója, rendezések input- és output eljárásai stb.

- az egyes adatelemekkel vagy logikailag összefüggő adatelem-csoportokkal kapcsolatos eljárások. Ugyanaz az eljárás különböző adatelemekkel kapcsolatban is megadható, továbbá az egyik elem eljárása hivható másik elem eljárásából. Adatelem-csoportnak lehetnek közös adatelemei.

3.2. Kapcsolattábla /adatleirás, funkcióspecifikálás/

Hasonlóan a felhasználói eljárások két csoportra osztásához, egy állandó és egy változó részből áll. Az állandó részben adhatók meg

- pufferek cimei, méretei,
- flagek a felhasználói eljárások és a keretprogram közti vezérlő információk átadására,
- az egyes adatelemektől független felhasználói eljárások belépési pontjai

A változó rész annyi "bejegyzésből" áll, amennyi az elemi adatok, szintetizált adatok és a csoportos ellenőrzések összmennyisége. A bejegyzésekben megadásra kerülő általános tulajdonságok pl: adatelem azonositója, külső megnevezése, egyes programokban az adatelem maximális hossza, egyedi/csoportos ellenőrzést ir-e elő, inputból származó vagy generált adatelem, részt vesz-e invertált /általában: redundáns/ információ képzésében vagy nem, ismétlődéses-e az adatelem, stb. Ezen felül csak az esetleges ellenőrző/generáló eljárás belépési pontját, ill. az annak hiányát jelző flag-et találjuk a leirásban.

3.3. A kapcsolattábla assembly eszközei

A rutinkapcsolásos technika esetében assembly makrók segitségével állitjuk elő a kapcsolattáblát. Assemblálás után ez programmodul formájában található, a felhasználói eljárásokra feloldatlan külső hivatkozásokat tartalmaz. A keretprogramot, a táblát és a felhasználói eljárásokat a linkage editor szerkeszti egységes programmá. Ettől kezdve a keretprogram hol adatmezőnek, hol programkomponensnek tekinti a táblázatot. Pontosabban, a benne szereplő belépési pontokat programcimekként kell értelmeznie, amit a PL/I nyelvben nem lehetne adatelemként kezelni. Igy az összeszerkesztett program előre /a forrásprogramban/ definiálatlan számu szubrutint akar hivni. Ezt egy apró assembly rutin felhivásával oldjuk meg /készitette: Bondy István/. A rutin paraméterként megkapja a kapcsolattábla egy bejegyzésének a cimét és a felhasználói eljárásnak szánt paramétereket. Ez a bejegyzésben /adatként/ megadott eljáráscimet felismeri, és a megfelelő eljárást a PL/I konvencióknak megfelelően aktiválja.

4. Tapasztalatok:

A rendszer fejlesztésekor megtakaritottuk egy bonyolult, de kellőképpen általánossá ugy sem tehető adatleiró nyelv kidolgozását és implementálálást. Az egyes ellenőrzések eljárásai többnyire néhány sorosak, de lehetőség van több száz soros, több modulból definiált feldolgozások /leképzések/ felhasználói eljárásokként való megadására /pl. szövegek leképezése lényeges fogalomkódok sorozatára/. Kiváló eszköz módszerünk a dekompozicióra, funkcók külön cserélhetőségének biztositására.

A karbantartás keretprogramját a KFIIR rendszeren belül két különböző eljáráscsomaggal használjuk. A preprocesszoros technikával egyetlen keretprogramból nagyon sok specializált tényadat-visszakereső és táblázó eljárás készült. A gráfreprezentációju ellenőrző-táblázatokon az ekvivalencia-reláció megadása mindig egyedi, a hashelés algoritmusa az adatok függvényében változik. A dinamikus szövegszerkesztővel igen különböző formáju és tartalmu, sok szöveges leirást tartalmazó, áttekintő táblát definiáltunk és állitottunk elő.

Az egész rendszert a KFIIR-től teljesen független információrendszer előállitására is alkalmazzuk az államigazgatás területén. Lehetőség van a KFIIR-rel kapcsolatos vállalati ill. ágazati információs rendszerek speciális igényeinek olyan kielégitésére, melyek lehetővé teszik az ágazatnál /vállalatnál/ a központi és a helyi adatok együttes kezelését, azokból egyszerü eszközökkel a KFIIR számára szükséges adatok mágneses hordozón történő átadását, mégpedig ellenőrzött adatokkal.

A rendszer jelenleg R35 gépen, OS VS1 operációs rendszer alatt müködik, a KFIIR rendszer több tizezer K+F bejelentés adatait tartalmazza.

A tervezett szolgáltatások körét tovább bővitjük.

Irodalom

[1] Bertalan József: A Kutatási-Fejlesztési Irányitási Információs Rendszerről NJSZT Kongresszus, Székesfehérvár, 1983

[2] Коммуникативный формат записи данных на магнитной ленте НТП МЦНТИ, 1-84, 1974, Moszkva

[3] Magnetic tape labelling and file structure for information interchange, ISO/R loo, 1969

Demjén Ferenc
OMFB KFIIR Kidolgozó Csoport

Teljes dokumentumok normalizálása természetes nyelvü információs rendszerben

Természetes nyelvü információs rendszerekben a beérkező szöveges információkat valamilyen módon normalizálni kell, azaz alkalmassá kell tenni gépi feldolgozásra. Ennek legelterjedtebb módja a géprevitel előtti kódolás. Hosszabb, természetes nyelvü szövegek kódolása csak gép segitségével történhet hatékonyan.

Rendszerünkben a gépi szövegnormalizálás főbb lépései a következők:

- szórabontás
- kódolás /szótőmeghatározás, szótári kód hozzárendelés/
- a kódolt szavak fogalmi visszavezetése főszókódokra /fontos szavak kódjaira/

I. A szórabontás

Egy dokumentum több rovatból áll. Ezek közül csak néhány, a rendszer szempontjából lényeges rovatot kell normalizálni. Kiválogatjuk a szórabontandó rovatokat. Minden rovat külön rekordba kerül. A szórabontás eredményeként keletkező szavak is külön-külön rekordba kerülnek. Minden egyes rekord tartalmazza az eredetre vonatkozó információkat, s így a normalizálás végén ismét összeállnak az egy dokumentumhoz tartozó információk.

A rovatok szórabontásához három segédtáblát használunk:

- nagyon gyakran előforduló lényegtelen szavak táblázata
- kötőjellel kapcsolható tagok táblázata
- pontot tartalmazó rövidítések táblázata

A teljes jelkészletet négy diszjunkt részre osztjuk fel:

- numerikus jelek /0/
- alfabetikus jelek /A/
- [illegible]
- [illegible]

A szórabontás elvei a következők voltak:

- Szónak tekintünk egy terminátorjelekkel határolt karaktersorozatot, ha tartalmaz A-beli elemet, de K-beli nem, feltéve, hogy nincs benne a lényegtelen szavak táblázatában.
- Nem szó az /NUK/-beli elemekből álló karaktersorozat.
- A K-beli elemet tartalmazó alfanumerikus karaktersorozatokra környezetfüggő vizsgálatot végzünk. Esetünkben K elemei: a kötőjel, a minuszjel és a pont.
- Ha egy karaktersorozat csak alfanumerikus jeleket és pontokat tartalmaz, akkor ellenőrzendő, hogy benne van-e a rövidítések táblázatában. Ha igen, akkor egy szónak tekintjük. Ellenkező esetben vesszük a sorozat kezdetétől az első pontig terjedő karaktersorozatot. Ezt szónak tekintjük, ha nincs benne a lényegtelen szavak táblázatában. A csonkolt karaktersorozatra ezt az eljárást addig ismételjük mig a karaktersorozat el nem fogy.
- Ha a karaktersorozat kötőjelet /vagy minuszjelet/ is tartalmaz, akkor az utolsó kötőjelet követő karaktersorozatot megvizsgáljuk, vajon benne van-e a ragok táblázatában. Ha igen, akkor a ragot levágjuk. Ezután kihagyjuk az összes kötőjelet és tömöritjük a karaktersorozatot. Ha az igy keletkező sorozat pontot tartalmaz, akkor ugyanugy járunk el mint az előbbi pontban. Ha nem tartalmaz pontot, nem lényegtelen szó és tartalmaz alfabetikus jelet, akkor az előállitott karaktersorozatot szónak tekintjük.

II. Szótárazás, kódolás

A szavak standard alakjának meghatározásához létrehoztunk egy szótárt. Egy szótári bejegyzés - egy szótár rekord - tartalmaz egy szót, a szót egyértelműen azonositó kódot, a szó hosszát, a szó bontástipusát és a szó összetevőinek kódjait.

A szó bontástipusának lehetséges értékei F, N, S, H, R, O, aszerint, hogy fontos szó /főszó/, nem lényeges szó, szinonimája valamely szónak, hibás alakja valamely szónak, rövidítése valamely szónak illetve összetett szó. A szó összetevőinek kódjai S, H, R tipusnál a visszavezetett alak kódja, O tipusnál a szóösszetevők kódjai.

A szótár jelenleg 110000 szót tartalmaz és folyamatosan bővül a beérkező új dokumentumok szónyersa alapján.

II. Szótárazás, kódolás

Szótárazás, kódolás alatt a következőket értjük:

- megnézzük, hogy a szó benne van-e a szótárban az adott alakban. Ha igen, akkor készen vagyunk, mert máris sikerült szótárazni és megvan a szótári kódja is.
- Ha a szót nem találtuk meg a szótárban, akkor elképzelhető, hogy ez csak azért történt, mert valamely szótárbeli szó ragozott vagy igekötős alakjáról van szó. Ezért csonkitjuk a szót egy karakterrel és ellenőrizzük, hogy a csonkitott szó benne van-e a szótárban, valamint azt, hogy a csonkolt karakter lehet-e rag vagy végződés, amelyet le szabad vágnunk.
- Ez a csonkolási sorozat addig tart, mig a szó két karakternyire fogy vagy valamely közbülső állapotct megtalálunk a szótárban.

Azok a szavak, amelyeket nem sikerült szótárazni, egy listán jelennek meg. Ezt a listát át kell nézni, a szükséges szótárbővitést el kell végezni és a szótárazást ujból lehet inditani a sikertelenül szótárazott szavakra.

A szótárazási folyamat igen sok szótárhoz fordulást követel és ez elég hosszu gépidőt vesz igénybe. Ezért a szótárhoz fordulások számának csökkentésére kidolgoztunk egy módszert, mely jelentősen csökkenti ezt a számot és igy a gépidő igényt is. Lényege a következő:

Az ábécé szerint rendezett szavakból egy aránylag nagy adaçot /200 Kbyte-nyit/beolvasunk a memóriába. Az azonos karaktersorozatokból csak egyet olvasunk be. A beolvasással párhuzamosan épitünk egy cimtáblázatot valamint egy munkatábla épitése is megtörténik. A cimtáblázat minden eleme 4 byte hosszu: az első byte a szó hossza, az azt követő három byte a szó realtiv cime az első beolvasott szó ciméhez képest. A munkatábla felépitése azonos a cimtábláéval. A munkatáblában a szavak ábécé szerinti növekvő sorrendjében találjuk a szavakra mutató cimtáblabeli bejegyzések cimeit.

A szótárazó program megállapitja a táblában levő szavak maximális szóhosszuságát és először a maximális szóhosszuságu szavak szótárazását végzi. A sikeresen szótárazott szavakat megjelöli, ezekkel tovább nem foglalkozik. Ezután csökkenti a maximális szóhosszuságot eggyel. Ez az aktuális szóhossz, és most már csak az aktuális szóhosszuságu vagy annál hosszabb, még nem szótárazott szavakkal foglalkozik. Minden szótárhoz fordulás előtt ellenőrizzük, hogy az adott szó aktuális hosszu kezdete megegyezik-e az előzőleg szótárazott szó kezdetével. Azonosság esetén a szótárhoz fordu-

lás elmarad. Csupán a csonkolt darabról kell még eldönteni, hogy az lehet-e toldalék. Ezt egy olyan táblázatos ellenőrzéssel végezzük, amelyben a keresés hash-kód technikával történik, tehát igen gyors. Miután végeztünk az összes aktuális hosszuságu szóval, ismét csökkentjük eggyel az aktuális hosszatt és ismételjük az előbb leirtakat.
Az algoritmus eredményét a két táblázat tartalmazza. A munkatáblában nullától eltérő hosszusággal rendelkező szavak szótárazása minden tőhossznál sikertelen volt. Ellenkező esetben a szótárban találtunk lehetséges szótövet. Ennek kódját és a szótő hosszát tartalmazza a munkatáblához tartozó címtáblabeli bejegyzés, feltéve, hogy a toldalék is helyes volt. A keletkező output ugyanolyan felépitésü mint az input. Csupán a szótőkód mező és egy jelzőbit változik, ha sikerült a szó kódolása. Igy szótár-update után ezt az outputot inputként használva csak a nem-szótárazott szavak szótárazását kell elvégezni.

Két lista készül. Az egyik a sikeresen szótárazott szavakat tartalmazza kódjaikkal együtt, a másik azokat a szavakat, melyeket nem sikerült szótárazni.

Ezzel a módszerrel 200 000 byte-nyi szó /20000-40000 szó/ 50-70 perc CPU idő alatt szótárazható. Ez elég gyors, ha figyelembevesszük a szótár méretét.

III. Főszókódra /fogalomkódra/ való visszavezetés

Ez a lépés a normalizálás utolsó fázisa, melynek eredményeként előálló rekord tartalmazza

- a feldolgozás első fázisából adódó input adatokat, karakteres formában.
- A szövegfeldolgozásra szánt rovatok szótárazása után nyert alapszókódvektorokat. Minden rovathoz egy alapszókódvektor tartozik. Az alapszókódvektor tartalmazza a rovatban előfordult szavak szótőkódjainak a sorozatát.
- A főszókódra való visszavezetés eredményeként előállitható főszókód vektorokat /minden szövegfeldolgozásra szánt rovathoz tartozik egy főszókód vektor is/.
- A főszókód vektorokból előállitható dokumentum vektort.

A dokumentumvektor tartalmazza egyszeres multiplicitással a szövegfeldolgozásra szánt rovatokból előállt főszókódokat.

A szótárazás után előállitunk egy olyan állományt, melynek rekordjaiba összegyüjtjük az egy dokumentumhoz tartozó rovatok szótőkódsorozatait, az alapszókódvektorokat. A főszókódra való visszavezetés inputját lényegében az alapszókód vektorok adják.

Az alapszókódvektor elemei szótári szótőkódok. Egy szótőkódhoz tartozó szótári rekord több tipusu lehet /F, N; S, H, R, O/. Ha egy szó bontástipusa nem F és nem N, azaz a szó visszavezethető egy vagy több szóra, akkor ezt a 'fogalmi' visszavezetést megtesszük. Mindaddig végezzük ezt a visszavezetést, mig minden egyes szótőkódtól eljutunk olyan szótőkód/ok/ig, amelye/ket/ már nem lehet visszavezetni ujabb szóra, azaz amig főszóhoz /F jelü/ vagy nem fontos szóhoz /N-jelü/ értünk. A multiplicitásokat, ciklusokat már menetközben kiszürjük.

A főszókódvektorból egy összefésüléssel és a multiplicitások kiszürésével nyerjük a dokumentum vektort. Ezzel a dokumentum normalizálása befejeződött.

Önmagában a normalizált dokumentumokat tartalmazó file-on lassu és nehézkes visszakereséseket végezni hasonlóság, illetve ujdonságvizsgálatot produkálni. E tevékenységek elvégzéséhez szükségünk van két, egymás duális párját képező invertált file létezésére. Az egyik az un. dokumentumtér, amely az előbbiekben leirt dokumentumvektorokból épül fel. A duális társ, az un. fogalomtér, inverze, azaz a fogalomtérben megtaláljuk az egyes fogalmak dokuemtumtérbeli előfordulásainak helyét.

A KFIIR rendszer fentebb leirt komponensei már elkészültek, sőt elfogadható gépidő és periféria igények mellett üzemeltetjük is. Eddig kb. 24000 dokumentum feldolgozása történt meg, ezeken sikeresen folytattunk hasonlóság-vizsgálatot.

Fogarassy Károly, LOGIC GM

LOGAR ADATBÁZISKEZELŐ RENDSZER

I. ÁLTALÁNOS ÁTTEKINTÉS

A LOGAR egy mikrogépre épülő, interaktiv adatbáziskezelő rendszer.

- mikrogépes: Z80 assembler nyelven iródott, igy - különféle feltételek megléte esetén - bármelyik Z80 alapu mikrogépre adaptálható

- interaktiv: felhasználói szintü párbeszédes üzemmódban müködik

- adatbáziskezelő: tehát egy adatbáziskezelővel szemben támasztott általános követelményrendszert céloz meg. Ezt a fogalomkört nézzük meg közelebbről. Az adatbáziskezelőnek az alábbi főbb követelményeket kell kielégitenie:

  - összetett logikai adatszerkezet kezelése: a LOGAR egy hálós /összetett hierarchikus/ felépitésü adatállomány kezelését biztositja
  - minimális redundancia: a többszörös adattárolás elkerülését a LOGAR többféle eszközzel biztositja /pl. az adatok közötti kapcsolatok pointerezéssel valósulnak meg/
  - konkurrens felhasználói hozzáférés: a LOGAR biztositja, hogy a tárolt és módositás alatt álló adatokhoz egyidejüleg más felhasználó ne férhessen hozzá; ezen kivül egyidejüleg több terminál kiszolgálását biztositja azzal, hogy a felhasználó által hivott rutinok reentrant-ak /ujra belépőek/,
  - többféle adat-hozzáférési módszert támogat: a LOGAR által kezelt adatok elérhetők kulcs szerint és kapcsolataikon keresztül is
  - adat-program függetlenség: a LOGAR egy adatbázis-leiró segitségével biztositja, hogy a felhasználó egy logikai adatstrukturát lásson, igy az adatbázis felépitésének változtatása nem jár feltétlenül programmódositással is
  - adatbiztonság: a LOGAR rendszer adatállomány másolással, illetve automatikus naplózással, helyreállitással biztositja az adatok védelmét az esetlegesen fellépő hardware hibák ellen.

Az eddig ismertetett általános szempontok után tekintsük át a LOGAR sajátosságait:

- hardware környezet: a fejlesztés és a tesztelés a VIDEOTON VT20/A rendszerén folyt, de a rendszer - megfelelő alapsoftware megléte esetén - bármilyen Z80 alapu mikrogépre adaptálható. Az adatbázis floppy-n, fix-

vagy mozgófejes diszken, illetve diszkeken helyezkedik el. A software támogatja a többterminálos adatelérést, 4 terminálig elfogadható hatásfokkal. Követelmény még a 64 kbyte-os memória megléte. A rendszer a mikrogépeken szokásos további perifériaválasztékot /printerek, lyukszalag/ is kiszolgálja.

. software környezet: a VT20/A rendszerre a LOGIC kifejlesztett egy LOGOS nevü, time-sharing operációs rendszert, mely támogatja a LOGAR futtatását. A LOGOS leirása külön dokumentációban szerepel. Más rendszereken is futtatható a LOGAR, ha az ott futó operációs rendszer az alábbi fő szolgáltatásokat nyujtja:

- dinamikus memóriakezelés
- szektorosan cimző diszk-handler

/Az adaptálásban a LOGIC szakemberei támogatást nyujtanak./

. a kezelt adatbázis mérete, korlátai:

a file-ok maximális száma	128
a file maximális rekordszáma	$2^{16}-1$
egy rekordban lévő mezők maximális száma	128
az adatterület maximális mérete	kb. 16 Mbyte

A LOGAR területigénye: 14 kbyte.

II. A LOGAR ISMERTETÉSE

II.1. Az adattárolás, elérés elve

Logikai felépités

Az adatbázis logikai elemei - a hagyományos terminológiával - file, rekord, mező.
/A felhasználó szempontjából: adatállomány, tétel, jellemző./

Egy file tehát rekordokból épül fel, a rekordok pedig mezőket tartalmaznak. A hagyományos file-kezelőktől eltérően a LOGAR a file rekordjai közötti kapcsolatok létesitését is lehetővé teszi.
Egy adott rekord elérése kétféle uton valósitható meg:

1. Vagy ismerem a keresett rekord egyedi jellemzőjét /vagyis azt az azonositót, ami a file-on belül csak ehhez a rekordhoz tartozik; ez különbözteti meg a rekordot a többitől/, pl. a gépkocsi tipus file-ban a TRABANT - ez esetben a kért rekordot a rendszer szolgáltatja. Tehát a rekord elérése a file név /gépkocsi tipus/ és rekord név /TRABANT/ alapján történik.

2. Vagy a kezemben van egy olyan rekord, melyhez a keresett rekordot rokoni kapcsolat füzi. Ez esetben a rendszer a kapcsolatokon keresztül éri el a keresett rekordot /illetve rekordokat/.

A LOGAR-ban a különféle rekord-mezőket tartalmuk alapján csoportba soroljuk /rekordok által tartalmazott mezőket tipizáljuk - vagyis csoportokra osztjuk/. A csoportositást a fizikai tárolás és a kezelés különbözősége indokolja. A csoportositás szempontjait és az igy keletkező 9 különböző tipusu mezőt leolvashatjuk a következő táblázatból:

Fizikai tárolás szerint	Kezelési szempont szerint	Mezőtipusok
a mezőt az adatrekord a direktben tárolja	szöveg	SZÓ
	szám	NUMERIKUS
		DCB
	egyéb	DÁTUM
a mező nem a rekordban található, a rekordban csak egy hivatkozás van	a rekord egyedi jellemzője	KULCS
	a jellemző csak meghatározott értéket vehet fel	LISTA
	kapcsolat	REFA
		REFF
	szöveg	TEXT

Fizikai felépités

A kezelés megkönnyitése és a tárolóhely jobb kihasználása érdekében a LOGAR - egy file-on belül - fix hosszuságu rekordokkal dolgozik. A változó hosszuságu jellemzőket ezért a rendszer a rekordon kivül helyezi el, az ugynevezett szótárakban.
Az egyedi azonositóval rendelkező adatrekordok eléréséhez a LOGAR az un. hash-coding eljárást használja. Az eljáráshoz szükséges, hogy az egyedi azonositókat egy külön adatállományban - az un. táblában - helyezzük el. Ugyanilyen okból, ugyanigy táblákban tároljuk a LISTA tipusu jellemzők által felvehető egyedi értékeket.

Összefoglalva: a LOGAR három különböző tipusu adatállományt kezel, e három file-tipus az

adatfile
a tábla és
a szótár.

Adatfile

A felhasználó definiálja, az általa meghatározott adatokat tartalmazza vagy direktben, vagy indirekt módon. Ez utóbbi esetben egy pointert tartalmaz, ami a keresett információra mutat. A rekordokat felépitő mezők - mint már előbb láttuk - 9 tipusba sorolhatók. Nézzük meg közelebbről a különféle tipusu mezők felépitését:

- szó tipusu: szöveges, előre definiált hosszuságu információ. Közvetlenül a mezőben helyezkedik el.
- numerikus tipus: max. $2^{16}-1$, 2 byte-on ábrázolt bináris egész, szintén a rekordban helyezkedik el.
- DCB tipus: előjeles, előre definiált hosszuságu tizedes szám. A mező direktben tartalmazza.
- dátum tipusu: a századfordulóhoz /1900/ relativ év, hónap, nap. Belső kódolása egy 2 byte-os bináris információ, mely a rekordban helyezkedik el.
- kulcs tipus: a kulcs az adott rekord egyedi értéke. Ez az érték egy táblában /az un. elérési táblában tárolódik. Az adatrekord nem tartalmazza a kulcsot, csak egy pointert az elérési tábla megfelelő rekordjára. A mező igy mindig 2 byte hosszu.
- lista tipus: szintén táblában /az un. érték táblában/ tároljuk, igy a megfelelő mező csak az érték tábla rekordindexét tartalmazza 2 byte-on. Ezt a jellemző-tipust akkor célszerü használni, ha az adott jellemző csak előre meghatározott értéket vehet fel, és ezek az értékek egy táblába előre összegyüjthetők.
- kapcsolat tipus: /REFA, REFF/
  - REFA egy fiu rekordról az apára mutató pointer.
  - REFF egy apa rekordról lemutat a legfiatalabb /időben utoljára keletkezett/ fiura. A mező ezen kivül tartalmazza a fiuláng hosszát is, 2 byte-on.
- text tipus: tetszőleges hosszuságu szöveges információ, mely egy szótárban kerül elhelyezésre, ugy a mezőben csak egy 3 byte-os pointer áll.

tartalmaz egy rekordpointert /ez mutat a kulcshoz tartozó adatrekordra/ és egy szótárpointert. Ez ad lehetőséget arra, hogy a rekordok egyedi kulcsához egy tetszőleges hosszu szöveges információ is tartozzék. Ezt a szöveget teszi el a rendszer egy szótárba és a szótárrekord pointere kerül az elérési tábla mezejébe.
Pl.: a gépkocsi tipus file kulcs tipusu jellemzői közül a Ø1-es a TRABANT. Igy a kulcs=Ø1, a szöveges megnevezés pedig a "TRABANT" string, ami egy szótárba kerül.

az érték tábla: ez is egyedi értékeket tartalmaz, de ez az egyedi érték nem tartozik egy file egy rekordjához, hanem az egész rendszerre vonatkozik. Ezért a táblaelemből hiányzik az adatrekordra mutató pointer.

A különböző tipusu tábla-file kezelését a rendszer teljesen automatikusan végzi, a felhasználó számára szükségtelen ismerni azok felépitését és kezelési módját.

Szótár file

A rendszer a változó hosszuságu szöveges információk tárolására a szótár file-okat használja. A felhasználói rendszer létrehozásakor meg kell adni a szükségesnek tartott szótarak számát és méretét.

Ezután, ha el kell tárolni egy adott hosszuságu szöveget, a rendszer kikeresi azt a szótárt, amelynek elemhossza megfelel az eltárolandó szöveg hosszának, és ide teszi el a szöveget.

Dr. Dajka Miklós
SZÁMALK

Az információrendszerek felhasználói szintű leírása adatbáziskezelő rendszerek alkalmazása esetén

A számítástechnika hazai terjedése következtében mind gyakrabban tapasztalhatjuk, hogy a vállalati információrendszerek egyes részeit az adatbázis-technika felhasználásával számítógépesitik. Jól segiti ezt a folyamatot az interaktiv feldolgozásokat is lehetővé tevő számitógépek vállalati szintü tömeges terjedése, ami kedvező feltételeket teremt a korábban szinte egyeduralkodó batch rendszerek napi operativitású on-line rendszerekkel való felváltására és ezen keresztül lehetőség nyilik az adatbázis-technika által nyujtott előnyök realizálására.

A DB-technika hazai alkalmazását illetően azonban egy ellentmondásos folyamatnak lehetünk tanui: egyfelől örvendetesen növekszik a hazánkban hozzáférhető adatbázis-kezelő rendszerek és az interaktiv számitógépek száma, ugyanakkor pedig ezek alkalmazás-technológiája területén alig tapasztalható előrelépés. Erre engednek következtetni az olyan jelenségek, hogy a kidolgozott adatbázis-rendszereket esetenként nem vesznek üzemszerü alkalmazásba, az elkészült rendszereket gyakran többször is átdolgozzák, és hogy a felhasználók ritkán nyilatkoznak kedvezően a számukra kidolgozott DB-rendszerekről.

Az adatbázis-kezelő rendszerek alkalmazásával kapcsolatos problémák többsége - véleményem szerint - a számitógépesités alapját képező felhasználói feladatmeghatározás hiányosságaira vezethető vissza. Kézenfekvő ugyanis, hogy a felhasználói feladat jelenti azt a kiindulási pontot, amiből elindulva az adatbázis szerkezete és müködése megtervezhető, és a kialakitott rendszerterv ezen a ponton még a felhasználóval is egyeztethető. Nyilvánvaló tehát, hogy amennyiben a hardware és software feltételek megfelelőek - amit készen kapott DB software esetén eleve feltételezhetünk -, akkor a sikertelenség okát az alkalmazástechnika területén, és ezen belül is a felhasználói feladatmeghatározás pontatlanságaiban kell keresnünk. A következőkben ezért részletesen megvizsgáljuk a felhasználói feladatmeghatározás szerepét és annak változását a számitógép-alkalmazás egyes korszakaiban, elemezzük a feladatmeghatározás tartalmát befolyásoló tényezőket, majd megvizsgáljuk, hogy jelenleg mennyire teszünk eleget a feladatmeghatározással szemben támasztott követelményeknek.

A felhasználói feladatmeghatározás szerepe

A vállalati számítógépalkalmazás hazai gyakorlatában a felhasználói feladatmeghatározás nagyon változó szerepet töltött be. Általában nem tekintették a számítógépalkalmazás önálló munkaszakaszának, habár jelentőségét korszakonként, ezen belül pedig vállalatonként, sőt projektenként is eltérően értékelték. Ennek megfelelően a feladatmeghatározás tartalma és funkciója is tág határok között változott.

A számítógépesítés kezdetén a felhasználói feladatmeghatározás szerepét a részletes rendszerterv töltötte be, mivel a kettő még nem vált szét egymástól. A számítógéptechnika /hardware/ akkori fejlettségi szintjét tekintve a rendszerterv és így szükségképpen a feladatmeghatározás is jórészt csak az input/output adatelemek rekordonkénti felsorolásából és a karakterek számának megadásából állt, amit a programozandó algoritmusok szöveges leirása tett teljessé. A főként mágnesszalagos konfigurációra épülő akkori rendszerek tervezéséhez és programozásához ezek az adatok, kiegészitve néhány volumen-adattal /pl. rekordok száma az egyes file-okban/, elegendőek is voltak, mivel a számítógépi eszközök választéke komolyabb tervezési optimalizálásra nem adott lehetőséget.

A lemezes tárolók megjelenésével és a különböző file-szervezési módszerek terjedésével a felhasználói feladatmeghatározás mind nagyobb szerepet játszott az alkalmazási rendszerek létrehozásában. Az új eszközök ugyanis egyrészt módot adtak arra, hogy az alkalmazás programrendszerét a megoldandó feladat viszonyaihoz igazitsuk, másrészt az erőforrások magas költségei és viszonylagos szükössége /pl. a tárolóterületé/ egy bizonyos határon túl rá is kényszeritette a tervezőket az optimalizálásra, mivel ellenkező esetben a feladatot esetleg nem is tudták volna megoldani.

Ilyen körülmények között a feladatmeghatározás tartalma szükségképpen kiszélesedett. Az adattipusok és a rekordszerkezetek megadása mellett szükségessé vált a lehetséges adatértékek részletes viszgálata is. Ennek keretében a maximális adatértékek előfordulási gyakoriságát, a többszörös adatértékek előfordulását, a karakterek tipusát, valamint a volumenadatokat kellett részletesebben feltárni ahhoz, hogy a programrendszer működését hatékonyabbá tudjuk tenni.

Az adatbáziskezelő rendszerek használatba vétele tovább növelte a felhasználói feladatmeghatározásnak a számítógépi feladatmeghatározásnak a számítógépi alkalmazási rendszerek kidolgozásában játszott szerepét. A DB-kezelő rendszerek ugyanis minden esetben megkivánják az adattipusok /egyed/, a lehetséges adatértékek /tulajdonság/, illetőleg az ezek közötti belső

kapcsolatok szisztematikus feltárását és rendezését, amit csak a felhasználói viszonyok részletekbe menő elemzése útján lehet elvégezni.

Az adatbázisok alkalmazásával a fejlődés tehát végül is oda vezetett, hogy a felhasználói feladatmeghatározás a számítógépalkalmazás önálló munkaszakaszává vált, olyan értelemben, hogy alapos és részletekbe menő feladatmeghatározás nélkül ma már egyetlen DB-projekt sem számithat automatikusan sikerre.

A felhasználói feladatmeghatározás tartalma

Nézzük meg ezek után, hogy konkrétan milyen tartalommal kell a feladatmeghatározást elkésziteni ahhoz, hogy abból hatékony programrendszer vagy DB-alkalmazás születhessen. Ehhez először is azt kell meghatároznunk, hogy a feladatmeghatározás hol és milyen helyet foglal el mint munkaszakasz az alkalmazási rendszerek fejlesztésében.

Azzal kell kezdenünk, hogy az előbb vázolt fejlődési folyamat a számitógép-alkalmazás folyamatában és módszerében is jelentős változásokat hozott. Korábban a koncepciókészités, a nagyvonalu és részletes rendszertervezés képezték a számitógépi alkalmazási rendszerek létrehozásának egyes munkaszakaszait, és ezek a szakaszok a felhasználói, illetőleg a számitógépi oldalon végzendő munkálatokat egyaránt felölelték. Az egyes fázisok munkavolumene, valamint a szükséges ismeretek mélysége és a specializálódási igény szükségessé tette a munkaszakaszok tartalmának átstrukturálását, és magát a folyamat lebonyolitását is megváltoztatta.

Napjainkban a számitógépalkalmazás leginkább a kétszintü tervezés módszerével történik, amikor is az egyes szinteken zajló munkálatok igen nagymértékben átfedik egymást. Az egyik szintet a felhasználói szint képezi, amelyben a szervező módszeresen összegyüjti mindazokat az ismereteket, amelyek a rendszerre vonatkoznak, majd ezek alapján kidolgozza a felhasználói feladat részletes megoldási tervét. A másik szintet a számitógépi megoldás szintje képezi, amelyben a programrendszer tervezője részletesen megtervezi a felhasználói rendszertervben leirt feladat optimális megoldását. Ezeket a munkálatokat azonban ugy kell elvégezni, hogy a két szinten végzett munkálatok valamilyen módon állandóan kölcsönös kapcsolatban legyenek egymással, mivel csak igy lehet biztositani az egyes szintek munkájához szükséges információkat.

Felhasználói feladatmeghatározáson /vagy felhasználói rendszerterven/ ezek után egy olyan részletes dokumentációt értünk, amely a megoldandó feladatra nézve mindazon ismereteket tartalmazza, amelyekre a feladat számitógépi megoldását végző programrendszer /vagy DB-rendszer/ megtervezéséhez és kivitelezéséhez egyáltalán szükség lehet.

Az ilyen céllal készült felhasználói feladatmeghatározás legáltalánosabban kétféle tipusú információkat tartalmaz; olyanokat, amelyek közvetlenül belekerülnek a programrendszerbe és annak részét képezik, valamint olyanokat, amelyek közvetlenül nem kerülnek bele a programrendszerbe, hanem csak annak felépitését, szerkezetét és müködését bevolyásolják.

Az első csoportot kézenfekvően maguk az input/output adatok, valamint az adatokkal végzendő müveletek /algoritmusok leirása képezik. Ez utóbbiak adatellenőrzési, számitási, csoportositási és rendszési stb. algoritmusok lehetnek, amelyeket más szempontból adatelőállitási és file-előállitási algoritmusokra bonthatunk. A vállalati rendszerszervezésben szerzett gyakorlati tapasztalatok arra engednek következtetni, hogy az adatok és algoritmusok leirását a felhasználók érdemben akkor tudják támogatni és segiteni, ha az számukra érthető módon, vagyis felhasználói szemléletben készül. Idegenkedést kelt az olyan dokumentáció, amelyben a felhasználó index-szekvenciális file-ok, karakter-stringek leirását találja meg. Felhasználói szempontból azt a dokumentációt tekinthetjük megfelelőnek, amely az adatok és algoritmusok leirását funkcionális bontásban adja meg, és az egész rendszert olyan felépitésben mutatja be, ahogyan azt a felhasználó az általa ellátott tevékenységekbe ágyazottan érzékeli.

A második csoport tulajdonképpen a rendszer müködését jellemző technikai paramétereket tartalmazza, amelyek leggyakrabban volumen- és gyakorisági adatok, feldolgozási és válaszadási időkorlátok stb. formájában jelennek meg. Részben ide tartozónak tekinthetjük továbbá az adattipusok, illetőleg az adatértékek közötti kapcsolatokat is, mivel a programrendszer felépitését ez is jelentős mértékben befolyásolja. Maguk a paraméterek azáltal fejtik ki befolyásoló hatásukat, hogy a feladat megoldására számitásba vehető hardware/software elemek közül egyeseket kizárnak, másokat illetően pedig azok alkalmazása mellett szólnak. Ezt a hatást természetesen a programrendszer tervező érvényesiti, aki a technikai paraméterek befolyásoló hatását a felhasználó által megadott szempontok szerint értékeli és enne eredményeként dönt a hardware/software elemek végső összetételéről.

Lényeges mozzanat, hogy habár a technikai paraméterek mindig a felhasználói feladat jellemzőiként jelennek meg, azok körét mindenkor a számitógépi megoldáshoz igénybe vett hardware/software elemek sajátosságai határozzák meg. Ebből következően a felhasználói feladatmeghatározás mindig eszközfüggő, vagyis az egyik adatbázis-rendszer alkalmazásához készült feladatmeghatározás egy másik rendszerhez egyáltalán nem vagy csak korlátozottan használható fel.

A feladatmeghatározás kidolgozása

Az előzőekből érzékelhető, hogy a felhasználói feladatmeghatározás a programrendszer tervezésseIk azonos nagyságrendü feladatot jelent. Ebből kifolyólag röviden azt is célszerü áttekinteni, hogy egy ekkora feladatot hogyan lehet hatékonyan megszervezni és koordinálni. A kidolgozás megszervezésének szempontjai közül véleményem szerint a következő három csoportot kell kiemelni:

- szakmai és tartalmi szempontok
- project-management szempontok
- dokumentálási szempontok.

A szakmai szempontokat tulajdonképpen már kifejtettük akkor, amikor elmondtuk a feladatmeghatározás tartalmát. Ennek kapcsán még egyszer hangsúlyozzuk, hogy a feladatmeghatározás képezi az egyetlen és egyben az utolsó olyan munkaszakaszt, ahol a felhasználó a számára értelmezhető módon, részleteiben látja a készülő rendszer elemeit és müködési vázlatát, és itt még olyan fázisban van módja elképzeléseit érvényesiteni /avagy megváltoztatni/, amikor a rendszer kivitelezése még nem kezdődött meg.

A project-management szempontok a feladatmeghatérozás kidolgozásával kapcsolatos vezetői elveket tartalmazzák. Nyilvánvaló, hogy a feladatmeghatározás kidolgozása nagyvolumenü feladatot jelent, amelynek során több munkatárs munkáját kell összehangolni és kkordinálni. Ennek kapcsán gondoskodni kell a munkafeladatok kiadásáról és számonkéréséről, a kiadott feladatok teljesitésének nyomonkövetéséről, ugyanakkor a feladatok tervezhetőségét is biztositani kell, egyrészt a határidők, másrészt a ráforditandó munakvolumen szempontjából is. Vezetési szempont továbbá az is, hogy a munkavégzésben állandó, standard munkarészeket különitsünk el, amelyek a munkatársaknak lehetőséget adnak a specializációra, ami növeli a munkavégzés hatékonyságát.

A dokumentálással kapcsolatos szempontok azt rögzitik, hogy a feladatmeghatározás kidolgozása csak egy közbenső munkafázist jelent, amelynek eredményeit a soronkövetkező munkafázisokat elvégző munkatársak használják majd fel. Ebből következően a feladatmeghatározást irásban dokumentálni kell, amihez megfelelő dokumentálási rendszert kell kidolgozni. A dokumentálási rendszernek egyrészt igazodnia kell a feladatmeghatározás elkészitésével összefüggő tevékenységekhez és ugyanakkor a dokumentáció felhasználása /vagyis a programrendszer tervezés/ során elvégzendő tevékenységekre is tekintettel kell lennie.

Mindezeknek a követelményeknek egy olyan szabványos ürlapokból felépülő dokumentálási rendszerrel lehet eleget tenni, amely előre rögziti az elvégzendő /és dokumentálandó/ munkafeladatokat, azok elvárt eredményeit, ezeket megfelelően strukturálja, és ugyanakkor kellőképpen rugalmas is ahhoz, hogy alkalmazkodni tudjon a feladatok és eszközök változásához. Az ilyen dokumentálási rendszer is csak akkor tudja betölteni project-management funkcióit, ha a dokumentálás a rendszer kidolgozásával egyidőben és nem utólag történik. Az ürlapos megoldás erre igen kedvező lehetőségeket nyujt.

A feladatmeghatározás alkalmazási problémái

Az eddigiekben megkiséreltük bemutatni a felhasználói feladatmeghatározásnak az alkalmazási rendszerek szervezésében játszott szerepét és a vele szemben támasztott követelményeket. A gyakorlati rendszerszervezési problémákat és a felhasználói szint jelentőségét tekintve ugy tünhet, hogy a feladatmeghatározás önálló alkalmazási munkaszakaszként történő felfogása teljesen indokolt, és ennek megvalósitása csupán csak idő kérdése. Szakmai körökben azonban közismert, hogy ez koránt sincs mindig igy, és a feladatmeghatározás esetenkénti elkészitése vagy elkészittetése időnként viszonylag komoly nehézségekbe ütközik.

Az első ilyen gátló tényezőt a kétszintes szervezési tevékenység szükségességének fel nem ismerése képezi. Egyes szervezők tagadják, hogy felhasználói tervezésre egyeltalán szükség van, Azzal érvelnek, hogy megfelelő szakmai felkészültséggel a felhasználói tervezés kihagyható, és a rendszerterv a felmérési eredmények alapján közvetlenül elkészithető. Ehhez azonban olyan hatalmas méretü ismeretanyagra lenne szükség /hardware, software, operációs rendszer, DB-rendszer stb. ismeretek/, amelyek megszerzése tömeges méretekben egyetlen szervezőtől vagy szervezői team-tól nem várható el reálisan.

A másik gátló tényezőt az egyes hardware/software rendszerek alkalmazása által megkivánt technikai paraméterek körének kidolgozatlansága jelenti. Egyszerüen nem állnak rendelkezésre olyan tájékoztató leirások arról, hogy egy adott eszköz használatához a felhasználói feladatnak milyen paramétereit kell a hatékony alkalmazás érdekében felmérni és a tervező rendelkezésére bocsátani. Ez jobbik esetben a felmérések kiegészitését eredményezi, de rosszabb esetben a szervezés sikerét kockáztatja, ami az egész projektet megkérdőjelezheti.

A harmadik gátló tényezőt a megfelelő dokumentálási rendszer hiányában jelölhetjük meg. A sokirányu kisérletezések ellenére ma sem rendelkezünk olyan dokumentálási rendszerrel, amelynek segitségével egy adott DB-rendszer felhasználói feladatmeghatározása teljeskörüen elvégezhető és dokumentálható lenne. E problémák leküzdése ma az egész szakmai társadalom érdekét szolgálná.

Locsmándi Miklós

Állami Fejlesztési Bank

Matematikai statisztikai módszerek alkalmazása adatbázisok kiértékelésénél

Az elmult évtizedben hazánkban dinamikus fejlődés ment végbe a számitástechnika, ezen belül is leginkább a gépi adatfeldolgozás területén. E folyamat mintegy melléktermékeként a nagyobb számitóközpontokban nagy volumenü, gazdag információtartalmu gépi adatbázisok jöttek létre. Ezek az adatbázisok, melyek esetenként 10-15 éves átfogásu idősorokat is tartalmaznak, szinte kinálják a lehetőséget az igényesebb matematikai statisztikai módszereket is igénybe vevő kiértékelő feldolgozások, kutatások számára.
Az Állami Fejlesztési Bank számitóközpontjában is létrejött már az az adatbank, mely 1971 óta tartalmaz adatokat a teljes beruházási pénzügyi folyamattal kapcsolatban. Az utóbbi néhány évben megszaporodtak az olyan megbizásaink, melyek e beruházási adatbank adatainak matematikai statisztikai feldolgozására irányultak, de még intenzivebb felhasználásáról számolhatunk be a PSZTI számitóközpontjában kiépitett vállalati gazdálkodási adatbázissal kapcsolatban.
A matematikai statisztika fontosabb módszereit felölelő programcsomagok természetes munkaeszközei számitóközpontjaiknak némelyik olyan programcsomag, mint pl. a BMD valóságos nemzetközi hirnévnek is örvend. A közelebbi megismerkedés velük azonban gyakran komoly csalódásokat okoz. E csalódások okát röviden igy jelölhetjük meg: a felhasználás komfortosságának teljes vagy részbeni hiánya. A komfortosságot általában is a számitástechnika alkalmazásának egyik jelenlegi legfontosabb problémájának tekinthetjük. Ma már természetesnek tekintjük, hogy a software-gyártó cégek által rendelkezésre bocsátott programok megoldhatósága megfelelő, s hogy a matematikai statisztikai programcsomag [illegible] pl. faktoranalizist számitó programot [illegible] természetesnek tekintjük, hogy a számitásokat bármilyen bonyolult [illegible] is [illegible], a program végrehajtja. [illegible]

[illegible]

togrammokat tud rajzolni, de kivülről nem lehet megadni az osztályközöket, illetve az automatikus osztályköz számitás csak egyenlőközü osztályokra vezethet.

- A kiindulási adatok közlésének lehetősége merev, korlátozott. Az input formai kötöttségei általában a felhasználható programcsomagok forrásnyelvére vezethetők vissza. A matematikai statisztikai programok szinte kivétel nélkül FORTRAN nyelven iródtak. A FORTRAN nyelv ilyenirányu korlátozottsága közismert. Különösen zavaróak azonban a komfort hiányosságai az adatbázisok kiértékelése esetén. Az adatbázisok legtöbbször valamilyen specifikus konvenciórendszer szerint épülnek fel, az adatvisszanyerés módszerei ennek megfelelően nem férnek el a FORTRAN nyelv keretei között. Az adatbázisok és a kiértékelő statisztikai programok közötti kapcsolatmegteremtése emiatt sokszor nyügös és pusztán technikai jellegü programozási munkát igényel.

- A paraméterezés korlátai. Természetesen a programok mindig paraméterezhetők, de gyakran olyan paramétereket is kivánnak, amelyek a gépben könnyebben előállithatók, mint kézzel. Pl. ha a kiértékelés egy adatbázis valamilyen részhalmazása irányul, melyet a megfigyelési egységek kiválasztó feltételeivel határozunk meg, a feldolgozásba bevont megfigyelési egységek számát csak a lekérdezés után tudhatjuk meg, ha kiiratjuk a gép memóriájából, s csak ezután tudjuk paraméterként megadni a feldolgozó programnak. Tipikus esete ez az olyan feldolgozási folyamatnak, amelyben két gépi feldolgozási lépés között az ember képezi a kommunikációs felületet.

- A szöveges információk hiánya. Hosszu időn át tartó, sok feldolgozási lépésből álló számitási folyamatoknái /ilyenek általában a tudományos igényu elemzések, melyeknél gyakran a már elért eredmények hozzák meg az igazi átvágvat/ fontos, hogy az egyes számitási eredmények, dokumentumok magukon viseljék mindazon azonositó jegyeket, melyek esetleges későbbi ujraelővételnél a felhasználót megfelelően tájékoztatják /melv megfigyelési mi-tán milyen változókkal mikor stb. futott a program/. A standard statisztikai programok szolgáltatásai e téren rendkivül szegényesek, a változókat csak sorszammal jelölik, a minta azonositására csak minimális, vagy semmilyen lehetőséget nem tartogatnak stb.

- A feladatvariációk ciklikus kezelésének lehetősége hiányzik. A gyakorlatban nagyon gyakori, hogy egy-egy számitási eljárást valamilyen paraméter változtatásával /pl. a minta részhalmazaira vonatkozóan/ többször is le kell futtatni. Ilyen esetben a kivánatos komfortot az jelentené, ha a programot egyszeri inditással többszöri futásra lehetne késztetni. A standard programok jó része erre nem alkalmas, vagy ha alkalmas is, a fentiekben már emlitett valamely hiányosság miatt a programot ujra és ujra inditani kell.
Anélkül, hogy az e tárgyban még tapasztalható hiányosságokat tovább részleteznénk, összefoglalásul egy példával szeretnénk illusztrálni eddigi mondanivalónkat. Szórásanalizis céljára rendelkezésünkre állt egy standard program, melynek az adatbázisunk adataihoz való kapcsolását megoldottnak tekinthettük /bár ez sem volt eleve adott/.
A program ciklikusan müködtethető ugyan, de csak egy változóval tud számolni. A számitásokat el kellett végezni a teljes mintára, azon belül ágazatonként, népgazdasági áganként, a kiemelt vállal atok körére, a vállalatok különböző nagyság kategóriáira, összesen 49 változóval, 5 év adataival. Az adatbázis adataihoz való kapcsolást/a lekérdezést/ is figyelembe véve könnyü megbecsülni, hogy a feladat megoldása ezernél több egyedi programfutást igényelt volna. Emiatt a meglevő program használatától el kellett tekintenünk, s helyette más megoldást kellett keresnünk.
Anélkül, hogy lebecsülnénk a standard csomagokban közreadott programok óriási jelentőségét, meg kell állapitanunk, hogy a fenti problémák alaposan rontják ugyanezen programok felhasználásának esélyeit, különösen, akkor, ha a program valódi kezelője nem kimondottan számitástechnikai szakember.

Számitóközpontunkban adottságként fogadhatjuk el, hogy vannak adatbankjaink, melyek felépitéséhez és müködtetéséhez egyrészt a Siemens cégtől származó SESAM adatbank kezelő rendszert, másrészt számitóközpontunkban kifejlesztett SEDI99 lekérdező rendszert használjuk, ez utóbbit 100 %-os kizárólagossággal. Ebből adódik, hogy a problémáink megoldását is ebben a software környezetben keressük.
A fejlesztés alatt álló rendszerünk szorosan kapcsolódik a már meglévő lekérdező - adatfeldolgozó rendszerhez. Igy maguk a statisztikai eljárások teljes mértékben mentesülnek az adatbázis kezelés, az adatok előzetes feldolgozása, transzformációja, tömöritése stb. kapcsán adódó problémáktól.
A számitások végrehajtásánál jelentkező variációs lehetőségeket messzemenően figyelembe vesszük, ugy, hogy programkomponenseink tudják mindazt, amit az általunk ismert standard programok tudnak, illetve, ami tapasztalataink szerint

ezen kivül szükséges lehet. Az adatbankjaink szervezési sajátosságainak megfelelően rendelkezésre állnak a lekérdezett ill. számitott változók-, adatok szöveges megnevezései, a tipikus részminta /ágazati, területi, felügyeleti stb./ megnevezések, melyek a számitások megfelelő pontjainál külön eziranyu intézkedés nélkül is automatikusan megjelennek.
A feladatvariációk ciklikus kezelésére több egyenértékü lehetőség is rendelkezésre áll, gyakorlatilag korlátlan számu lekérdező, kiértékelő, számitó programlépés füzhető fel egy programinditáson belül többé vagy kevésbé automatizált cikluskezeléssel. Az igy előkészitett feladat-csomagok az adatbankban tárolhatók, s később változatlanul, vagy ujabb variációkká módositva lefuttathatók. A fejlesztés első ütemeként a leiró statisztika módszereit egyesitő alrendszer készült el. Tapasztalataink igazolják, hogy a korszerü, komfortos software termék előállitásánál a munkaigény aránya erősen eltolódik a komfort elemek felé, s egyre kisebb sulya van a tényleges számitási feladatot ellátó matematikai programnak. Esetünkben a matematikai program mag és a körülötte dolgozó komfort "felhő" közötti arány /programsorok száma, munkaigény, futási időigény, stb./ kb. 1:20.
A következő ütemekben rendszerünkbe kivánjuk integrálni a matematikai statisztika minden fontosabb, gyakran használt módszerét. Arra természetesen nem törekedhetünk, hogy rendszerünk "mindent tudjon", amit ezen a téren tudni lehet. De - tapasztalataink szerint - a számitási eljárások használtsági foka ill. matematikai igényessége bonyolultsága forditott arányban állnak egymással, s a jövőben is arra számithatunk, hogy az egyszerübb módszerek alkalmazására kell a nagyobb sulyt helyeznünk. Azon eljárások számára pedig, melyeket rendszerünk nem tartalmaz /s esetleg nem is fog tartalmazni/ most is fennáll a külső programhoz való csatlakozás lehetősége.

Locsmándi Miklós

Állami Fejlesztési Bank

Építkező programgenerátor SESAM tipusu adatbankok kiértékeléséhez

A gépi adatfeldolgozás fő módszerei már évtizedekkel ezelőtt kialakultak, s máig annyira kikristályosodtak, hogy a legkülönbözőbb szintü számitástechnikai oktatási intézményekben alaptantárgyként oktatják. A mindennapos gyakorlat ezzel szemben azt mutatja, hogy még mindig nem születtek meg azok az általános megoldások, melyek programozóinkat megkimélnék a különböző tipikus feldolgozási lépések ujra és ujra megprogramozásától. Igaz, régebben is és ma is rendszeresen napvilágot látnak programrendszerek, melyek magukat univerzális, végleges, teljesen általános megoldásként árulják. Ezek sikere azonban általában csak átmeneti és korlátozott.

Egy általános igényü programrendszer megalkotásának receptje lényegében ismert. Az adott területen már megoldott problémák elemzésével meg kell keresni a közös vagy többé kevésbé stabil logikai elemeket, s segitségükkel meg kell konstruálni azokat a programkomponenseket, melyek a felmerülő problémák elegendően nagy hányadát megoldják. A programkomponensek lehetnek pl. assembler makrok, melyek egy assembler nyelven irott programba ágyazva a feladat fő terhét viselik, de lehetnek kész modulok, melyeket egy főprogramból szubrutinként hivhatunk. Az általános igényü programrendszerek egy más része futtatható, kész programfázisokból áll, de elképzelhetünk más számitástechnikai realizációkat is.
A makrok ill. szubrutinok adott határok között nagyon jól használható módszert jelentenek, de végülis nem küszöbölik ki a programirást, tesztelést, s ráadásul a rendszer bonyolultságának növekedésével párhuzamosan kezelésük egyre nehézkedsebb.
A kész programfázisok rendkivül praktikusak az azonnali használhatóságuk, futtathatóságuk miatt, de a feladat fokozott bonyolultsága egyre igényesebb paraméterezési eljárást kiván, ami viszont egy határon tul már a programirás nehézkességét vonja maga után.
Végül is nem lehet kikerülni a tényt, hogy egy adott bonyolultsági szinten tul bármilyen számitástechnikai technológiát használunk, a programrendszer használata nem lehet egyszerü. Az alábbiakban egy olyan adatbázis kiértékelő rendszerről lesz szó, mely a hagyományos adatfeldolgozás területéhez tartozó feladat tipusok igen széles körében nyujt gyors, komfortos megoldási lehetőséget. Rendszerünket, a SEDI99 programrendszert, mely a SIEMENS cég SESAM nevü adatbank rendszeréve létrehozott adatbankok kiértékelésére alkalmas, számitóközpontunkban évek óta 100 %-os kizárólagossággal használjuk feladataink megoldására, de felhasználóinak köre egyre szélesedik belföldön és külföldön is.

A rendszer koncepciójának teljeskörü ismertetésére itt nincs lehetőség, a legfontosabb sajátosságokat csak a tájékoztatás kedvéért soroljuk fel:

- a rendszer futtatásra kész programfázisokból áll,
- a felhasználó és a rendszer közötti kapcsolat négy önálló, egymástól különböző kommunikációs sikon lehetséges /feladatközlő nyelv, dialog vezérlő parancsok, üzemi paraméterek, batch vezérlő paraméterek/,
- a feladat megfogalmazása a tipikus tevékenységek specifikációja utján történik,
- a feldolgozó ill. outputáló tevékenységek messzemenően függetlenek egymástól,
- az adatok megnevezhetőek /a háttérben a SESAM rendszer adatkatalógusával/,
- a feladat megoldás minden része technológiai egységet képez, a programfutást semmilyen technikai ok miatt nem kell megszakitani,
- batch, enter és dialog üzemmódokban történhet a futás, ezen üzemmódok más és más feldolgozási módokat, más és más kezelési lehetőségekkel támogatnak,
- a végleges feladatmegoldó program a feladat specifikációjának fogadásával párhuzamosan alakul ki, s a rendszer futása során tetszőleges számu alkalommal tudja magát ujra "generálni",
- az egymás után futó feladatok, u.n. "lépések" egymáshoz láncolhatók, egymásnak adatot adhatnak át a munkatárolóban, az adatbankban, vagy külső adathordozón.

Témánk szempontjából a fenti elemek közül kettőnek van különös jelentősége, a tipikus tevékenységek elvének ill. a rendszer öngeneráló, önépitő jellegének.
Rendszerünk tipikus tevékenységei ismertek a hagyományos adatfeldolgozás eszköztárából, s ezen a téren semmilyen elvi ujdonságról nem beszélhetünk. Olyan tevékenységekről van szó, mint primér és szekundér adatvisszanyerés, szelekció, agregálás, tételes vagy összesitett adatokkal végzett számitások, rendezés, a számitott adatok tárolása stb. A tipikus tevékenységekből gyakorlatilag minden olyan feladat megoldása összeállitható, mely adatállományok konvencionális feldolgozását célozza. A tipikus tevékenységek programtechnikai szempontból nézve többé kevésbé vázlatosan megirt programelemek, melyek végleges alakjukat a feladat teljes meghatározása után nyerik el, s melyek végrehajtása azonnal meg is indul. Az egyes tipikus tevékenységek között lényeges különbségek

vannak specifikálhatóságuk, ill. a megvalósitásukat végző programelemek előregyárthatósága szempontjából. Az agregálásnál pl. csak két ponton jelentkezik variációs lehetőség, az agregálandó adatok számát, ill. hosszát illetően. Ennek megfelelően az agregálás specifikációja az adatok neveinek egyszerü felsorolásából állhat, végrehajtó programelemként pedig elképzelhető egy szinte teljesen véglegesen megirt programrutin, mely az agregálandó adatok cimét és hosszát, valamint egy ciklusszámlálót kivéve kész és változatlan. Hasonlóan egyszerü a rendezés kezelése, ahol a szintaktika megelégedhet a rendező kulcsként szolgáló adatok neveinek felsorolásával, a végrehajtó rutin pedig nem áll másból, mint a rendező alprogram felhivó paramétereinek aktualizálását végző néhány utasitásból.

Más a helyzet pl. a feltételes számitásokat részletező specifikációk kezelésével. Itt összetett feltételek és bonyolult képletek értelmezésére kell felkészülni, a végrehajtó programelemek pedig előre még vázlatosan sem irhatók meg. Az eddig emlitett két véglet között az átmeneti szituációk sokféleségével találkozhatunk. Igy eljutunk az önépitő programrendszer fogalmához. Lényege, hogy a végrehajtó program azon elemeit, melyek a feladat specifikációjának fogadása előtt előre nem láthatóak, a rendszer egyáltalán nem tartalmazza, hanem azokat a specifikáció fogadásával párhuzamosan hozza létre. Ebben a pillanatban a rendszer mint valamely programnyelv fordítóprogramja viselkedik. A forditás eredményeként létrejövő programelemeknek futáskor együtt kell működniök a standard, kész, vagy menet közben aktualizálódó programelemekkel. Az együttmüködés biztositása a szerkesztőprogram funkcióját igényli. Igy az önépitő program egyesiti magában a fordító és szerkesztő program funkcióit is /esetünkben ehhez a "könyvtár" kezelése is járul, de ennek nincs köze az önépitő funkciókhoz/.

A feladat megoldása időben két szakaszra oszlik, az értelmezés és a végrehajtás szakaszára.

Az értelmezés időszakában a rendszer egymás után olvassa a tipikus tevékenységeket specifikáló paramétereket, ezek alapján felépiti a megfelelő tevékenységek végleges programjait. Az utolsó paraméter értelmezése után készen áll a végrehajtó program, indulhat a feldolgozás. A végrehajtás szakaszában az értelmező funkciókat ellátó programelemekre nincsen szükség, sőt jelenlétük esetén szükségtelenül a többszörösére nőne a tárolóigény. Ez a körülmény indokolja az overlay technika széleskörü alkalmazását.

Mindezek után az önépitő rendszerünk működését az alábbiakban foglalhatjuk össze.

- a munkatároló három régióra oszlik:
  a rezidens modulok régiójára,
  a szintaktikai elemző modulok régiójára, és
  az adott feladat lépés számára specifikusan generált programelemek régiójára.

- a rezidens régió mindegyik tipikus tevékenység számára egy modult tartalmaz, mely a végrehajtás idején szükséges, vázlatosan megirt programrutinokat hordozza.

- egy adott paraméter értelmezésekor véglegesitődik a rezidens modul, s a harmadik régióban megjelennek a generált specifikus programelemek.

- az utolsó paraméter értelmezése után a feladat végrehajtásában résztvevő tevékenységek mindegyike véglegesitett rezidens modullal és a harmadik régióban kész programelemekkel rendelkezik, ugyanekkor felszabadul a második régió. A második régiót a végrehajtás szakaszában az output eljárás programmodulja foglalja el.

Láthatjuk, hogy a rendszer inditásakor rendelkezésre álló programrészek, melyeket az előbbiekben vázlatosan megirtnak neveztünk, valójában a feladatot megoldó programnak mintegy csontvázát képezik, melyeket a feladat értelmezésekor tölt meg a rendszer hussalvérrel. A feladat lefutása után a hus elillan, s ismét üres csontváz várja a következő feladatot.

Dr.Szelezsán János
KSH, Államigazgatási Számítógépes
Szolgálat

Hálós adatbázisok egy matematikai modellje

Legyen adva tulajdonságtipusok /adattételtipusok/ egy A halmaza. A hálós adatbázisban a tulajdonságtipusokat rekordtipusokba szervezzük és összefüggéseket kapcsolatokat irunk fel az egyes rekordtipusok között, megadva ezzel a tulajdonságtipusok közötti kapcsolatokat is.

A tulajdonságtipusok értékei /előfordulásai, adattételek/ adják meg az adatbázis összes lehetséges kimenetét /nevezzük ezt az adatbázis outputjának/. A tulajdonságtipusok értékeiből állnak össze a rekordtipusok előfordulásai, a rekordtipusok közötti kapcsolatok meghatározzák a tulajdonságtipusok előfordulásainak kapcsolatait is.

Legyen R a hálós adatbázis r_i rekordtipusainak halmaza. A rekordtipusok kapcsolatait /és ezen keresztül a tulajdonságtipusok kapcsolatait is/ egy G irányitott gráffal adjuk meg, amelynek csúcspontjai az R halmaz elemei. A G gráfot az adatbázis gráfjának hivjuk.

Két szomszédos csúcspontot összekötő él az adatmodellben set-tipust alkot.

A gráfnak azokat a csúcsait, amelyekbe nem vezet él, navigációs gyökérelemeknek nevezzük, azokat, amelyekből nem megy ki él navigációs végpontoknak hivjuk. Egy navigációs gyökérelemtől egy navigációs végpontig vezető út a navigációs út /lásd IDMS/. A navigációs utaknak közös részük is lehet. Minden navigációs út a rekordtipusok közötti kapcsolatot /kapcsolat-sorozatot/ fejezi ki.

A navigációs úton a rekordtipusoknak sorrendje van; a gyökérelem kivételével minden rekordtipusnak van előző és a navigációs végpont kivételével követő rekordtipusa. Az előző rekordtipust owner-nek, a követőt member-nek nevezzük. Ha egy p rekordtipus bármely navigációs úton megelőzi q-t, illetve q követi p-t /nemcsak közvetlenül/ akkor azt p < q-val jelöljük.

Az adatbázis egy $\mathcal{F}$ leképezésnek tekinthető, amely a /lehetséges/ kérdések halmazát a válaszhalmazba képezi le. Minden válaszhoz /a tulajdonságtipusok előfordulásaihoz/ egy navigációs úton jutunk hozzá.

Az alábbiakban [2] -re támaszkodva egy matematikai modellt adunk a G gráffal leirt adatbázissal reprezentált $\mathcal{F}$ leképezésre.

Legyen $Kq = \{ p : p < q , \forall p \in R \}$

$Mq = \{ p : q < p , \forall p \in R \}$

Legyen $U_k(r_0)$ az r_0 rekordtipust megelőző rekordtipusok által meghatározott k -adik út, azaz legyen

$U_k(r_0) = r_{m_k}^k \dots r_1^k r_0^k$

és legyen $M_k \subset M r_{m_k}$

Vezessük be az $r U_k(r_0)$ utak ahol $r \in M_k$ az $M_k U_k(r_0)$ jelölést.

Vegyük az r_0 -t követő utak $K_{r_0}^k$ halmazát. Legyen ennek M_k egy részhalmaza. Az $M_k U_k(r_0) r$ alaku utak halmazát, ahol $r \in K_{r_0}^k$ az r_0 -ra támaszkodó navigációs útsorozatnak nevezzük, és $N(r_0)$-lal jelöljük. $N_k(r_0) \in N(r_0)$ egy navigációs útvonal, $r_{m_k}^k$ a navigációs útvonal gyökere.

Egy N_k útvonal $\mu(N_k)$ hosszán az úton lévő csomópontok eggyel csökkentett számát értjük. Jellemezzük az adatbázis bonyolultságát a leghosszabb navigációs úttal.

Az adatbázishoz tartozó összes navigációs útvonalak halmazát jelöljük N-el. Az adatbázis a (G,N) párral irjuk le.

Az adatbázisnak a G gráffal illetve az N -nel a navigációs utak halmazával történő leirása alkalmas arra, hogy egy adott $\mathcal{F}$ leképezést megvalósitó (G,N) párhoz megszerkesszünk egy ugyancsak az $\mathcal{F}$ leképezést leiró olyan (G',N') párt, amely valamilyen szempont szerint kedvezőbb.

Legyenek adva az $N_k(r_0) \in N(r_0), (k = 1,2,\dots,\ell)$ utak, $(r_0 \in R)$ és jelöljük ezek közül $N_k^{(i)}$ -val azokat $(k = 1,2,\dots,\ell_1)$ amelyekre $\mu(N_k^{(i)}) = 1$

Definiáljuk az $N_0(r)$utat a következő módon:

$U_0(r_0^0) = r_0^0 = r_0 , \quad M_0 = M_{r_0} \setminus \bigcup_{i=1}^{\ell_1} M_i , \quad K_0 = K_{r_0}$

Legyen $N = (N_0, N)$ és

$M_k = \bigcup_{j=1}^{\mathcal{J}} M_j^k \ (k = 1,\dots,\ell)$ahol ha $M_j^{k_1} \cap M_i^{k_2} \neq 0$ akkor $M_j^{k_1} = M_i^{k_2}$

Bontsuk fel az $\{M_j^k\}$ halmazt $(j = 1,2,\dots,\mathcal{J}, \ k = 1,2,\dots,\ell)$ $C_1', C_2', \dots, C_{\mathcal{K}}'$ csoportokra úgy, hogy

$M_{j_1}^{k_1} \in C_i$ ha $M_{j_1}^{k_1} = M_i^k$, $M_i^k \in C_i$

$M_{j_1}^{k_1} \notin C_i'$ ha $M_{j_1}^{k_1} \neq M_i^k$; $M_i^k \in C_i'$

és jelöljük a c_j-vel a C_i elemeit.

Az R helyett vegyük az $R_1 = R - \{r\} \bigcup_{i=1}^{w} r_i^*$ halmazt, ahol $r_i^* \equiv r_0$, $i = 1,2,\dots,w$. Az R_1 -hez szerkesszük meg a G_1 gráfot a következő módon:

$$K_{r_i^*}^{(1)} = \begin{cases} K_j & \text{ha } r_i \subset M_j,\ r_0 \notin K_j,\ j \leq w \\ (K_j \setminus r_0) \cup r_k^* & \text{ha } c_i \subset M_j,\ r_0 \in K_j,\ j \leq w \end{cases}$$

ahol feltesszük, hogy $r_0 \in c_k$. Az R többi elemére legyen:

$$K_r^{(1)} = \begin{cases} K_r & \text{ha } r_0 \notin K_r \\ (K_r \setminus r_0) \cup r_i^* & \text{ha } r_0 \in K_r \text{ és } r \in r_i \end{cases}$$

Szerkesszük meg az előbbieknek megfelelően N-ből N_1 -et.

Legyen $s > \ell$ esetén és ha $v_{\ell_i+1} \neq r_0$ akkor $s \leq \ell$-re is

$$U^{(1)}(v_0^i) = v_{-\ell_i}^i \cdots v_{-1}^i v_0^i \quad \text{ahol} \quad v_s = \begin{cases} r_s & \text{ha } r_s \neq r_0^* \\ r_j^* & \text{ha } r_s = r_0, r_{s-1} \in c_j \end{cases}$$

$s \leq \ell$ esetén pedig ha $v_{-\ell_i+1} = r_0$ akkor $v_{-\ell_i+1} = r_k^*$ ha $r_0 \in c_i$. Az $i \leq \ell$ értékekre legyen $M_i^{(1)} = \bigcup r_k^*$ feltéve, hogy $\bigcup c_k = M_i$

Amennyiben $i > \ell$ legyen

$$M_i^{(1)} = \begin{cases} M_i & \text{ha } r_0 \notin M_i \\ (M_i \setminus r_0) \cup (\bigcup r_k^*), & c_k = M_j^p,\ r_{-\ell_i}^i \in K_p,\ p \leq \ell_1 \end{cases}$$

és $i > \ell_1$ esetén

$$K_i^{(1)} = \begin{cases} K_i & \text{ha } r_0 \notin K_i \\ (K_i \setminus r_0) \cup r_k^* & \text{ha } r_0 \in K_i, \end{cases}$$

A fenti eljárással megszerkesztettük a G_1 gráf $N_1 = \{N_1^{\ell_1+1}, N_1^{\ell_1+2}, \dots, N_1^{\ell}\}$ halmazát, ahol $N_1^{\ell} = \{M_i^{(1)} \cup (v_0^i), K_i^{(1)}\}$

Most megmutatjuk, hogy megadható olyan T transzformáció, amely G_1 utjait G utjaiba viszi át.

Legyen $Tr = \begin{cases} r & \text{ha } r \neq r_k^* \\ r_0 & \text{ha } r = r_k^* \end{cases}$ $k = 1,2,\dots,w$

Legyen $U^{(1)}$ a G_1 gráf $r_1' r_2' \dots r_m'$ útja és legyen $TU^{(1)} = \{Tr_1, Tr_2, \dots, Tr_m\}$. Vegyük a $TU^{(1)} = r_1 r_2 \dots r_m$ utat és jelöljük ezt U -val. Megmutatjuk, hogy U a G gráf útja. Legyen $r_i = r_i'$ $(i = 1,2,\dots,m-1)$ Nyilván $r_{i+1}' \in K_{r_i'}^{(1)}$, mivel $U^{(1)}$ a G_1 gráf útja. Ha $r_{i+1}' \neq r_k^*$ akkor $r_{i+1}' = r_{i+1}$ és $r_{i+1}' \in K_{r_i'}$ Világos, hogy $r_i r_{i+1}$ a G gráf éle.

Ha $r'_{i+1} = r_k$ is teljesül akkor $r_{i+1} = r_0$, tehát $r_{i+1} = r_0 \in K_{r_i}$ azaz $r_i r_{i+1}$ ebben az esetben éle a G gráfnak.

Ha $r_i \neq r'_i$ akkor $r'_i = r^*_k$ és $r_i = r_0$

Mivel $r'_i r'_{i+1}$ a G_1 gráf éle ezért nyilván $r'_{i+1} \in K^{(1)} r^*_k$ és amennyiben $r'_{i+1} = r_{i+1}$ akkor $r_{i+1} \in K r^*_k$ Ez azt jelenti, hogy $r_{i+1} \in K_k \subset K_{r_0}$ azaz $r_i r_{i+1}$ a G gráf éle.

Ha $r'_{i+1} \neq r_{i+1}$ akkor $r_{i+1} = r^*_\ell$, $r_0 \in K_k$ vagyis $r_0 \in K_{r_0}$.

Azt kaptuk tehát, hogy $r_i r_{i+1}$ a G gráf éle.

Beláttuk tehát, hogy a T transzformáció a G útjait G_1 utjaiba viszi át, azaz $T U^{(1)} = U$

A hálós adatmodellnek az ismertetett módszer segitségével történő leirása lehetővé teszi egy adatbázisnak kedvezőbb tulajdonságu adatbázisba való átalakitását.

Irodalom

1. D.C Tsichritzis, F.H. Lochovsky: Data Base Management Systems. /Academic Press New York, 1977/

2. Ю.И. Смирнов: О преобразовании операторной схемы. (СБ „Проблемы кибернетики" Вып. 3, 1960)

Pölöskei Pál
SZÁMALK

Néhány adatbázis alkalmazás tapasztalatai /Az IDMS használata alapján

Az IDMS adatbázis-kezelő rendszer nem egészen 3 éve került Magyarországra. Ezalatt az idő alatt igen széles körben elterjedt, ma már több mint 50 számítóközpontban használható. Természetesen az IDMS alapu alkalmazások kidolgozása időt vesz igénybe és ezért még nagytömegü felhasználói tapasztalatról nem beszélhetünk. A rendszer alkalmazását a SZÁMALK az elsők között kezdte meg és ma már üzemelő rendszereket dolgozott ki. Emelett több más intézmény is sikerrel használja az IDMS-t /MÁV, ÉGSZI, SZTAKI, stb./. Az előadás három, már üzemelő alkamazást tekint át, kitér néhány felhasználói tapasztalatra és kiegészitő program fejlesztési javaslatra.

A SZÁMALK IDMS alkalmazás-fejlesztési tevékenységében az ÁTM /volt OTÁF/ takarmány-nyilvántartási és ellenőrzési rendszere, a csepeli Fémmü AFG DB/DC rendszere, valamint a BAGE MEVIR rendszere tekinthetők az első siekres kisérleteknek. Ezek az alkalmazások már müködőképesek, noha további fejlesztésük, kiegészitésük folyamatban van.

Az ÁTMI takarmány-nyilvántartási és ellenőrzési rendszere kb. egy éve müködik. Az ÁTMI egy országos laboratóriumi hálózaton keresztül ellenőrzi a felhasználásra kerülő takarmányokat. A laboratóriumi vizsgálatok eredményeit központilag értékelik ki un. "etalon" értékkel való összehasonlitás alapján. A kiértékelés alapján intézkedésekre /pl. a felhasználás megtiltása/ is sor kerül. Ezen felül statisztikai elemzések készülnek a takarmány felhasználás hatékonyságáról. Az évi kb. 50 ezer takarmányminta vizsgálata alapján gyors intézkedésekre és további elemzésekre van szükség. Ennek érdekében a számitógépes rendszer előállitja

- egy adott időszakban vett takarmányminták osztályozását,
- adott gyártóknál vett minták értékelését,
- az előállitott takarmány mennyiségét megyénként,
- a mérések értékeinek átlagait és
- egyéb statisztikákat.

Ebben az évben viszonylag kis ráforditással egészitettük a rendszert a vizsgálatok költségeinek ki-

mutatásával és számlázásával.

Az adatbázis 12 rekord tipusból és 12 set tipusból áll. Az adatbázis mérete 100 ezer rekord, 70 MB. A rendszer üzemeltetése során figyelembe véve a méreteket és a szolgáltatásokat a futási időkkel kedvező tapasztalataink vannak. Jelenleg folyik a rendszernek on-line funkcókkal való kiegészitése. A csepeli Fémmü részére a Csepel ISZI közremüködésével egy on-line anyag- és félkésztermék gazdálkodási rendszer készült. Jelenleg a rendszer próbaüzeme /párhuzamos feldolgozás/ folyik.

A rendszer kezeli a szükséges anyagok és fogyóeszközök adatait. Az adatbázis szerkezetét 17 rekord tipus és 13 set tipus adja. 25 ezer törzs és 45 ezer készlet rekordot tart nyilván. Az adatbázis mérete kb. 20 MB. A rendszer havi kb. 10 ezer mozgást kezel főként kötegelt üzemmódban, de megengedi az on-line karbantartást és lekérdezést is. A feldolgozás 10 naponként történik, a listák az adatbázis valamennyi változását rögzitik. További lekérdezések és változások terminálon keresztül lehetségesek. A felhasználó 7 update listát, valamint 15 elszámolási táblát kap az anyagokról. Az on-line rész elfogadható válaszidőkkel müködik, mig a batch feldolgozás futási idői nem haladják meg egy hagyományos file szervezésü rendszerét.

Az IDMS-nek a SZÁMALK-nál folyó legösszetettebb alkalmazása a BAGE által kezdeményezett mezőgazdasági vállalatirányitási rendszer /MEVIR/ kidolgozása.
A rendszer az első fázisban /1985-ig/

- a készletgazdálkodás,
- az eszközgazdálkodás,
- a munkaerőnyilvántartás és bérelszámolás,
- termelés- és üzemelszámolás

modulokat tartalmazza.

Ebben az esetben különösen lényeges az adatbázis szemlélet, hiszen itt ugyanazokat az adatokat többféle feldolgozás céljára használjuk. A rendszer ujdonságai közé tartozik, hogy egy erőforrásokkal kapcsolatos müvelet valamennyi rendszerbeli következménye - a termelés részletes bontásu szükitett önköltségének kimutatásáig - külső beavatkozás nélkül végrehajtódik.

Először a készlet modul készült el, év eleje óta 3 gazdaság éles feldolgozása üzemel. A modul kielégiti a könyvelés, az elemzés és a statisztika igényeit,

bizonyos gazdálkodási elemeket is tartalmaz. kb 20 táblát állit elő.

Az adatbázis 32 rekord és 23 settipusból épül fel. Jelenleg mintegy 60 ezer rekord előfordulást tartalmaz, mérete kb 25 MB. Mindez 3 gazdaság adatait tartalmazza, amiből a közös rész kb. 40 ezer /törzs/ rekord, 8 MB. Az üzemeltetési tapasztalatok kedvezők. 10-15 ezer tétel feldolgozása legfeljebb egy R55 müszakot igényel.

Az eszköz modul szintén elkészült. Egy gazdaságra a próbaüzem jelenleg befejezés alatt áll. A modul gondoskodik az eszközök nyilvántartásáról, a gépkartonok aktualizálásáról, az értékcsökkenés elszámolásáról, a táblatörzskönyvek vezetéséről, az eszközök teljesitményének nyilvántartásáról. A modul adatbázisa 14 rekord és 7 set-tipusból áll. Egy átlagos gazdaságra 15-20 ezer rekordelőfordulást tartalmaz, mérete mindössze kb. 3 MB. A próbaüzem során azt tapasztaltuk, hogy a feldolgozási költségek nem lesznek számottevők. A további modulok fejlesztés alatt állnak.

Megemlitem még, hogy a SZÁMALK eddigi legnagyobb vállalkozása, a VIDEOTON belső információs és irányitási rendszere is az IDMS felhasználásával készül.

Az IDMS eddigi használata megengedi annak kijelentését, hogy az adatbáziskezelés elég kényelmes és hatékony eszköz az alkalmazási rendszerek fejlesztésére. Az eddigi tapasztalatok azt is sejtetik, hogy a rendszerek üzemeltetése sem különösebben költséges.

Az alkalmazások kidolgozása során kitünt, hogy az IDD használata támogatja a tervezést, mert gyakorlatilag az első rendszerre vonatkozó információktól kezdve sokmindent nyilvántarthatunk vele, igy mindenki számára viszonylag egyértelmü és ugyanaz az információ /mégpedig a legutolsó változat/ áll rendelkezésre. Ezt a lehetőséget teszik kényelmesebbé a SZÁMALK uj, kiegészitő fejlesztései /SAID, IFRA/.

A CULPRIT használata igen hatékonynak bizonyult, mert igen gyorsan nagymennyiségü tábla készitését teszi lehetővé. Az OKQ 1.2 változata azonban - bár igen egyszerü lekérdező nyelve van - lényegében csak a fejlesztőknek nyujt hasznos segitséget. Az ESZR gépeken nem vált be a gyorstöltő segédprogram.

Néhány egyéb tapasztalat:

- a sémában kisebb változtatásokra a program tesztek során is sor kerülhet,
- érdemes az adatszótárból és az adatbázisból legalább egy másolatot is tartani a teszt időszakban, mert a teszt futások gyakran "elronthatják" a szótárt és az adatbázist;
- a programteszteket először csak kis méretü teszt adatbázison érdemes végezni, de szükség van viszonylag nagyméretü tesztadatbázisra is, mert a programok performanciája nehezen becsülhető;
- a sorted láncok használatát nagyon alaposan meg kell gondolni, mert könnyen ronthatják a rendszer müködési hatékonyságát,
- betöltés előtt célszerü az adatokat előrendezni, mert a lemezek gyakori mozgatása lassitja a müveletet.

Végül megemlitem, hogy a SZÁMALK az adatbázis alapu rendszerek kidolgozásának technológiáját jeleneg fejleszti. Ennek keretében támogató szoftver csomagokat dolgoz ki. Ide tartozik a már emlitett SAID továbbá több kisebb segédprogram jellegü szoftver. A technológia középpontjában az IDD áll.

Pogány András
COMPORGAN rendszerház

Egy fájl elérési és mozgatási stratégia lokális elosztott rendszerekre

1. Bevezetés

A fájlok elhelyezése és mozgatása minden elosztott rendszernek központi problémája. A stratégiák költség minimalizálásra törekszenek figyelembe véve, hogy mind az elérésnek mind a tárolásnak bizonyos költségei vannak. Átrendezésre akkor kerül sor, ha az elérési költségek tartósan megnövekszenek és található a fájl számára egy olyan uj csomópont ahol az elérési költségek lényegesen csökkenthetőek. Tehát a fájlok átrendezéséhez elég sok statisztikai adat összegyüjtése és kiértékelése szükséges.

Az ismertetésre kerülő algoritmus a fájlok átrendezését mindezen bonyolult adatgyüjtő és kiértékelő algoritmusok nélkül végzi, de természetesen csak akkor hatásosan, ha a környezet bizonyos feltételeknek eleget tesz. Feltételeink,főleg a mikrogépek világában, a gyakorlatban rendszerint teljesülnek.

Feltételezzük, a felhasználók tevékenysége olyan, hogy egy-egy fájl halmazon hosszabb ideig dolgoznak s egy fájl halmaz elemei időben csak lassan változnak, azaz a fájl hivatkozásokban létezik bizonyos csomósodás. A fájlok igénybevétele lekérdezés jellegü /pontosabban, intenziv használat esetén ritka a felujitás/. A fájlok nem tulságosan nagy méretüek az egyes csomópontokban lévő tárolókapacitáshoz mérten.

2. Az algoritmus müködése

Az osztott fájl rendszerben /ami nem szükségszerüen adatbázis, de lehet az is, mintahogy hálózati szintü katalógus, vagy katalógizált fájl rendszer/ nyugalmi helyzetben /amikor mindenfajta tevékenység szünetel/ a fájlok elhelyezése osztott nem duplikált. Ilyenkor a rendszerben lévő összes fájlnak csak egyetlen pél-

dánya létezik, ezt elsődleges példánynak hivjuk.

Ha egy felhasználó, egy nem a saját csomópontjában lévő fájlra hivatkozik lekérdezés jelleggel, akkor az adott fájlról egy másolat keletkezik a felhasználó csomópontjában egy munkaterületen /ezentul raktár/. A raktár mérete olyan, hogy egyidejüleg több fájl befogadására is alkalmas. A raktár fájljai, lekérdezésre, lokális fájlként viselkednek. A raktár megtelése esetén a selejtezést például LRU algoritmus szerint lehet végrehajtani /a raktár megtelése és ürítése analóg a lapozott memória kezeléssel -a memória a raktár, a lapok a fájlok- és igy a lapozásnál használt algoritmusok itt is alkalmazhatóak. Az analógia azonban még messzebb menő: ahogy a lapozás feltételezi, hogy egy program memória hivetkozásai csomósodnak, ugyanugy mi is feltételezzük, hogy egy felhasználói tevékenység fájl hivatkozásai csomósodnak./.

Elvünk tehát az, hogy egy fájl /első közelitésben legalább olvasásra/ legyen ott, ahol dolgoznak vele. Ez azt jelenti, hogy a fájlok azon csomópontokban duplikálódnak, ahol szükség van rájuk. Ez a duplikáció azonban idéglenes, s egy tevékenység befejeztével, akár automatikusan /selejtezés utján, vagy a felhasználó kijelentkezésével/, akár kifejezett utasitásra megszünik. Azaz alapvetően nem változtatja meg sem az adatbázis elosztott nem duplikált jellegét, sem a fájlok csomópontokhoz rendeltségét.

Felujitás jellegü hozzáférés esetén nem másolat készül, hanem első lépésként a fájl átmozgatásra kerül a hivó csomópontba, s ott nem a raktárnak lesz egy eleme, hanem a lokális fájl rendszerhez kapcsolódik. Ezzel megváltozik a fájlok csomópontokhoz rendeltsége. Mivel feltételezzük, hogy felujitást általában a fájl "tulajdonosa" ad ki, ez egy "jó" irányban történő elmozgatás, mivel várhatóan a tulajdonos csomópont változtatása miatt a legközelebbi felujitás szintén innen indul ki.

Az átmozgatást követően az összes másolatot érvényteleniteni kell mielőtt a felujitást végrehajtanánk, nehogy a többi csomópont alavult adatokon dolgozzon tovább. A fájl megjelölésével egy "elavult" üzenet kibocsátására kerül sor, amely a csomópontok raktáraiból egyszerüen törli az adott fájl másolatait. Mindezek után a fájl felujitható, s az ezt követő hivatkozások a felujitott fájlról készitenek másolatot.

Ez az alapalgoritmus, s mielőtt implementációs problémákba bocsátkoznánk vizsgáljuk meg mi az a környezet, amit algoritmusunk számára elképzeltünk.

3. Környezet, alkalmazás

Legyen egy személyi számitógépekből álló lokális hálózat. Az egyes gépeken a felhasználók önálló tevékenységet folytatnak, de időről-időre más felhasználók által előállitott adatokkal is dolgoznak. Ezek az adatok lehetnek különálló fájlok, de lehetnek egy adatbázis elemei is. Mind az egyes gépek lokális katalógusai, mind az azokat egyesitő globális katalógus szimbólikus hivatkozásokat tartalmaz, hogy mind a katalógus részletek, mind a fájlok maguk a gépek között tetszőlegesen áthelyezhetőek legyenek. Az egyes gépeken folyó tevékenységet például programfejlesztésnek képzelhetjük, ahol az egyik fejlesztő által előallitott modulokat több más fejlesztő épiti be a programjaiba. De lehet ez egy hivatali rendszer is, ahol a különböző osztályok egymás eredményeit használják fel, vagy a titkárnő a főnök feljegyzései alapján készit különböző dokumentumokat, leveleket stb.

A fent vázolt rendszerben algoritmusunk kettős szinten alkalmazható, egyrészt a katalógusra, másrészt a fájlokra. A leggyakrabban használt fa szerü katalógusokra algoritmusunk alapelve /a felhasználási helyre gyüjteni azon adatelemeket, amelyekre a közeljövőben valószinüleg hivetkozás történik/ kétszeresen is érvényes. Egyrészt, hogy az egyszer hivatkozott adatelem

valószinüleg ujra felhasználásra kerül, másrészt a hivatkozások nem függetlenek egymástól, hanem a katalógus strukturáját követik, azaz két egymást követő hivatkozás egymás logikai környezetébe esik. Például katalógusok esetén algoritmusunk kiegészithető azzal, hogy egy elemre való hivatkozásnál, nemcsak az elemről, hanem annak közvetlen logikai környezetéről is másolat készül. Ez Ethernet tipusu lokális hálózatok esetén még a hálózat terhelését sem növeli, mivel az optimális csomagméret több katalóguselemet képes magába foglalni.

4. Néhány szó a megvalósitásról

Egy lokális hálózatban, akármilyen elosztott rendszer megvalósitásánál, a következő problémával kell számolni. A lokális hálózat általában személyi számitógépekből áll amelyeket nem folyamatos üzemben használnak. Egy-egy gép lekapcsolásával vagy számitani kell arra, hogy az elosztott rendszer egy része hozzáférhetetlenné válik, vagy lekapcsolás előtt a gépen tárolt információt a rendszer többi részének át kell venni. Az egyes gépeken tárolt információ kikapcsolás esetén való teljes mentése, különlegesen nagy háttértároló kapacitások nélkül, a kikapcsolások előrehaladtával a rendszer teljes összeomlásához vezet. Mindenképpen kompromisszumot kell kötni a mentendő adatok mennyiségére vonatkozóan. Egy célszerü kompromisszumnak látszik a katalógus mentése és a fájlok "veszni hagyása". Esetleg ez azzal finomitható, hogy a felhasználó bizonyos fájlok mentését is előirhatja. Ezzel a módszerrel arra a kérdésre mindenképpen választ tudunk adni, hogy egy adott fájl létezik-e vagy nem, legfeljebb magát a fájlt nem tudjuk szállitani. Az elosztott rendszer a gépeknek egy kivétellel való kikapcsolásával centrálissá válik, s ezen megmaradt egy gépre hárulna az összes mentendő információ tárolása. Ezért célszerü a hálózatba beiktatni egy folyamatos üzemben

dolgozó csomópontot, viszonylag nagy háttértár kapacitással. Ez a csomópont venetné át a kikapcsolt gépektől a mentendő információt. Ezzel az archiválás is egyszerüvé válik, hiszen minden mentendő információ egy helyre gyülik /az osztott rendszerből centrális lesz/ amiről pl. az éjszaka során másolat készithető.

Egy másik implementációs probléma az elsődleges példányokról készült másolatok nyilvántartása, illetve elérése egy felujitás során. Az egyik véglet, hogy a másolatokról semmiféle nyilvántartás nem készül. Ennek az előnye, hogy a másolatok a hálózat többi csomópontjának értesitése nélkül törölhetőek, s nem kell nyilvántartás vezetni. Hátránya, hogy felujitás esetén a hálózat összes csomópontját értesiteni kell a felujitási szándékról, hiszen nem tudható, hogy melyik tartalmaz másolatot. A másik véglet, hogy minden egyes másolat helyét nyilvántartjuk. Ilyenkor egy másolat eleresztésekor a nyilvántartót vagy nyilvántartókat /pl. láncbafüzés esetén/ értesiteni kell, viszont a felujitás kevesebb üzenettel jár, hiszen csak azon csomópontokat kell értesiteni, ahol tényleges másolat létezik. Egy közbülső változat, ami akkor használható jó hatásfokkal, ha két felujitás között a másolatot készitő csomópontok száma nem tul sok. Az elsődleges példánynál nyilvántartjuk a másolatot kérő csomópontokat. Egy másolat törlésekor nem küldünk értesitést, igy valamivel több csomópont van nyilvántartva, mint ahány tényleges másolat létezik, de még mindig kevesebb, mint ahálózat összes csomópontja. Felujitáskor csak a felsorolt csomópontokkal kell üzenetet váltani a másolatok törléséhez.

5. Irodalomjegyzék

Howard L.Morgan és K. Dan Levin: Optimal Program and Data Locations in Computer Networks
CACM May 1977 315-322.

Enrique Grapa és Geneva G. Belford; Some Theorems to Aid in Solving the File Allocation Problem
CACM Nov 1977 878-882.
Raymond R.Ranko: Facing Basic Issues in Office Automation
Computer Networks 5/1981 391-399.
Peter J. Denning: The Working Set Model for Program Behavior
CACM May 1968 323-334.

HIRDETÉS MELLÉKLETEK

TERTA

TELEFONGYÁR
Budapest
Hungária krt 126-132.
1143

Computerta

USP - felhasználói programok készitését támogató programcsomag a TAP - 34 számára

Az USP /User Support Package/ programcsomag célja az, hogy megkönnyitse a helyi adatfeldolgozási feladatok megoldását a TAP-34 intelligens terminálon.

A programcsomagot alkotó szubrutinok elsősorban a klaviatúráról történő adatbevitelt, a hajlékony mágneslemezen történő adattárolást és visszakeresést, valamint kisebb adatfeldolgozást igénylő feladatok megvalósitásakor alkalmazhatók előnyösen. Ilyen feladat lehet például raktárnyilvántartás, számlázás, személyi adatnyilvántartás stb.

A programcsomag főbb funkciói

ADATBEVITEL

Az USP adatbeviteli rutinjai lehetővé teszik a programozó számára, hogy a felhasználói képernyő bármely sorában egy, legfeljebb 89 karakter hosszúságú input mezőt definiáljon, és azt a kezelővel kitöltesse.

Többek között lehetőség van a mezőbe bevihető karakterek ti-

pusának meghatározására /pl. numerikus mező/, a bevitt karakterek megjelenitésének előirására /pl. aláhúzva/, a mező karaktereinek a mező lezárása utáni igazitására /pl. jobboldali mezőhatárra igazitás/, numerikus mező esetén kezdő zérus feltöltésre vagy elnyomásra, a kitöltött mező tartalmának érték szerinti automatikus ellenőrzésére /pl. minimális érték/, stb.

ADATTÁROLÁS

Az USP hajlékony mágneslemezen történő adattárolásra vonatkozó rutinjai jelentősen kibővitik a TAP 34 "SYS" rutinjaival adott lehetőségeket, rugalmasabb file szervezést és kezelést tesznek lehetővé.

Az USP lemezkezelő rutinjai által létrehozott és kezelt file-ok rekordos szervezésüek. A file-okban az adatok logikai rekordokban helyezkednek el, a rutinok segitségével az adatok logikai rekordonként irhatók vagy olvashatók. Az egyes logikai rekordokat nem csak sorosan, hanem véletlenszerüen is el lehet érni. Egy file-on belül az egyes rekordok azonos hosszúságuak. Lehetőség van a file-okban tartalom szerinti keresésre is.

A lemezkezelés megfelel az IBM diskette formátumnak.

ADATFELDOLGOZÁS

Az USP adatfeldolgozó rutinjaival numerikus adatokon aritmetikai müveleteket, byte-sorozatokon pedig különböző manipulációkat hajthatunk végre.

Numerikus adatok ábrázolására kétféle lehetőséget nyujt a programcsomag. Az egyik esetben egész- és törtrészből álló fixpontos, decimális, karakteres formában tárolt, előjeles számokkal dolgozhatunk. Ekkor a számábrázolás pontossága 14 digit, ami gyakorlatilag minden ügyviteli, gazdálkodási feladat megoldására elegendő. A másik számábrázolási mód 16 bites, előjel nélküli, bináris egész számokkal dolgozik, ez elsősorban a programok belső szervezéséhez és számlálásokra használható.

Numerikus adatok között a négy alapmüveletet /összeadás, kivonás, szorzás, osztás/, összehasonlitást és kerekitést hajthatunk végre az USP rutinjaival.

A byte-sorozatokon számos müveletet hajthatunk végre, igy többek között feltöltés, irás, olvasás, beszúrás, törlés, másolás, összehasonlitás és keresés müveleteket.

ADATMEGJELENITÉS

Az USP formátumvezérelt adatmozgató rutinjai egy egyszerü módon megadható formátum alapján tördelve, rendezve, módositva és konstans részekkel kiegészitve jelenitik meg a kivánt adatokat. Eközben olyan hasznos segédfunkciókat is ellátnak a rutinok, mint például a nyomtatón a fejléc automatikus nyomtatása lapváltáskor és a lapok számozása.

HIBAKEZELÉS, ÁLLAPOTJELZÉS

Az USP minden rutinjának egységes, közös hibakezelése van. Az egyes rutinok a program futása közben ellenőrzik a bemenő paramétereket és adatokat, valamint a müveletek során keletkező eredményeket, és jelzik, ha valahol hibát észlelnek.

A hibajelzés rugalmasan alkalmazkodhat a kivánt feladathoz. A standard hibajelzés és hibaüzenet /amely információt ad a hiba keletkezésének helyéről és a hiba jellegéről/ helyett a programozó saját hibaüzenetet definiálhat, vagy akár az egyes hibák kezeléséhez saját hibakezelő programot irhat.

Az USP 1983 elején tapasztalatszerzésre néhány felhasználónak átadásra került, és az első tapasztalatok kedvezőek.

PÉNZÜGYI SZÁMITÁSTECHNIKAI INTÉZET

A PÉNZÜGYI INFORMÁCIÓRENDSZER BÁZISINTÉZETE

FELADATAI:

- A központi és a tanácsi pénzügyekkel összefüggő szervezési, számitástechnikai és koordinációs feladatok ellátása /rendszerfejlesztés, adatfeldolgozás/
- A gazdasági- pénzügyi szabályozás és irányitás számitástechnikai támogatása /szervezés, adatfeldolgozás, gazdaságmatematikai módszerek/
- Az Állami Biztositó, a pénzintézetek, PM vállalatok és intézmények részére szervezési és számitástechnikai szolgáltatások nyujtása
- A pénzügyi információrendszer központi és hálózati eszközeinek üzemeltetése, müszaki fejlesztése, karbantartása.

A KÖZPONTI NAGYSZÁMITÓGÉPEKHEZ CSATLAKOZÓ TPA HÁLÓZATI RENDSZER ÉS VIDEOTON TERMINÁLOK EGYARÁNT LEHETŐVÉ TESZIK AZ ADATFELDOLGOZÁSI FELADATOK MEGOLDÁSÁT KÖTEGELT ÉS INTERAKTIV ÜZEMMÓDBAN.

PÉNZÜGYI SZÁMITÁSTECHNIKAI INTÉZET
1023 Budapest, Lajos u. 17-21.
Telefon: 684-020

ROZMARING
KERTÉSZETI MG. TSZ.
NAGYKOVÁCSI

1525 BUDAPEST, Pf. 86
MNB: 380-22344
TELEX: központ: 22-6863 kertészet: 22-4262
TELEFON: 165-841, 365-937, 364-562

A RODATA GT.

VIDEOPLEX-3 tipusú adatrögzitő

rendszerre

SZERVEZÍST és ADATRÖGZITÍST

vállal

1983.IV. negyedévétől.

Cim: RODATA GT.
Budapest
Pinceszer u. 14-16.
1028

Tel.: 251-299/176
165-460

MEZŐGAZDASÁGI
ÜGYVITELSZERVEZÉSI IRODA
BUDAPEST II., Érmelléki u. 13.
Levélcím: BUDAPEST Pf. 230
1536

A számitástechnika alkalmazása a mezőgazdaságban

A mezőgazdasági nagyüzemek termelési jellegüket, gazdálkodási formájukat tekintve eltérnek az ipari termelő egységektől. Az eltérés elsősorban abban jelentkezik, hogy alapvető termelési profiljuk lényegében a területi adottságok, - az időjárás, a talajviszonyok, a környezeti infrastruktura - függvénye. A termelési ciklusok zártsága, a viszonylagosan kevés eszközráforditással járó profilváltás viszonyt rugalmas, a piaci viszonyokhoz gyorsan alkalmazkodó gazdálkodási módot tesz lehetővé. Az agráripari termelés sajátosságainak elemzésekor nem hagyhatjuk figyelmen kivül a mezőgazdasági szolgáltató ágazatokat, amelyek ma már elsősorban nem azzal tünnek ki, hogy a munkaerőt helyben foglalkoztatják, hanem az alaptevékenység volumenét megközelitő vagy esetleg meg is haladó termelési értékükkel.

A fejlett termelési rendszerekkel rendelkező mezőgazdasági nagyüzemekben az optimális termék és termelési struktura kialakitás az adottságok az aktuális gazdasági szabályozók, a várható kül- és belpiaci viszonyok figyelembevétele szélesköru szakmai ismereteket és nagymennyiségü információ kezelését és gyüjtését teszi szükségessé.

A Mezőgazdasági Ügyvitelszervezési Iroda Közös Vállalat Számitástechnikai Főosztálya elsősorban ezen információk gyüjtésében, tárolásában, kezelésében, feldolgozásában, valamint a számitógépes eszközök és rendszerek beszerzésében, telepitésében nyujt segitséget a termelőszövetkezeteknek.

Ebből következően a MÜSZI számitástechnikai tevékenysége ket-

tős: egyrészt adatfeldolgozó rendszerek szervezése, valamint az ezek müködtetéséhez szükséges számitástechnikai eszközök kiválasztása, azok beszerzésében, installálásában való közremüködés.

A korábban iskertetett információ igényekből kiindulva mindkét tevékenységi kör müvelésénél meghatározó, hogy a kapott adatokat milyen célra, milyen szinten használják fel, illetve hogy mely gazdálkodási terület kiszolgálását segiti elő a számitógép.

A termelési rendszerektől, helyi adottságoktól független területek - készletgazdálkodási, munkaügyi, bérügyviteli, pénzgazdálkodási, állóeszközgazdálkodási rendszerek - adatainak feldolgozását un.tipusrendszerek alkalmazásával oldottuk meg.

Ezeknél a tulajdonképpen ügyviteli rendszereknél jelenleg még az eseményt regisztráló adatfelvétel, a batch jellegü feldolgozás a meghatározó, de készülnek osztott rendszerü feldolgozások is, amelyek kihasználják a helyben lévő, tehát gyorsan hozzáférhető minigépek előnyét és a nagygépek komolyabb tárkapacitását és könnyebb programozhatóságát.

A speciális, un.profilérzékeny rendszerek általában a termelésirányitással és elszámolással kapcsolatosak, az ezeket feldolgozó programcsomagok kidolgozása összetettebb feladat, ezért a MÜSZI különböző gazdálkodási területeken végzett felméréseket és ezek alapján kezdett hozzá pl. az INRA szarvasmarhatenyésztési, elszámolási programcsomag honositásához.

Az ilyen tipusu számitástechnikai rendszerek már on-line feldolgozást, interaktiv adatelérést és adatkezelést feltételeznek, ezért szükségesnek érezzük, hogy a fizikailag osztott rendszerek szervezése mellett, felkészüljünk a távadatfeldolgozás, és azzal együtt az on-line adatfeldolgozás alkalmazására is.

Számitógépek alkalmazása a mezőgazdasági üzemekben

1. A mezőgazdasági üzemek információs rendszere

A mezőgazdasági üzemek többségére jellemző, hogy a tényleges müködtetést befolyásoló információs rendszerük nem teljes áttekintés alapján tervezett, hanem pillanatnyi állapot rögzitése vagy szokásjog szerint müködik. Ennek a gyakorlatnak az oka több irányban is kereshető:

- az üzemek alakulásának módja /összevonások/
- földrajzi tagoltság
- a termelés méretbeli növekedése és az információs rendszer fejlesztése közötti fázis különbség.

Az összevonások alkalmával biztos, hogy eltérő gyakorlatot folytató gazdaságokat vettek egy igazgatás alá, de - érthető módon - meghagyták a nem alapvető jelentésü müködési sajátosságokat. Ez az eltérő gyakorlat sok esetben azonos funkciókra különböző bizonylatok, kimutatások készitésének gyakorlatát is jelenti.

A mezőgazdasági üzemek feladataik jellegéből és történelmi okok alapján is viszonylag kiterjedt területen fejtik ki tevékenységüket. A területi tagoltság - erre való igyekezet esetén is - gátolja az egységes információs rendszer kialakulását.

A méretbeli növekedések és a technológiai fejlődés magával hozta az információs igények emelkedését, de ez igen gyakran kielégitetlen igény marad. Az uj helyzet esetleg nagyságrendben nagyobb mennyiségü adatfeldolgozását igényli még azonos szintü információnyújtás elérése érdekében is.

2. Az adatfeldolgozáshoz igénybe vett eszközök

Az irányitáshoz és gazdálkodáshoz szükséges információk előállitása a termelés állapotának megfelelő léptékü ismeretét jelenti, illetve az állapotot mutató adatok feldolgozása révén lehetséges. Az adatok gyüjtése és feldolgozása napjainkban is elsősorban a könyvelés igényei és szempontjai szerint történik. A feldolgozáshoz igénybe vett könyvelőgépek és azok fejlettebb változatainak is közös jellemzője a

- nagy előkészitési igény /bizonylatok rendezése, stb./
- szükös tároló és feldolgozó kapacitás
- centralizált müködtetésük és a gép lehetőségei sem tesznek lehetővé alkalomszerü lekérdezéseket.

Egyre több üzem ismeri fel a fejlettebb adatfeldolgozási technika szükségességét és jelentőségét. Most már több év tapasztalata alapján lehet értékelni és becsülni a Termelőszövetkezetek Számitógépes fejlesztési Együttmüködése /TSZFE/ vállalkozásban résztvevő TSZ-eknek a számitógépes adatfeldolgozás szélesebb körü elterjesztése érdekében tett kezdeményezését.

A pest-környéki TSZ-ek által vásárolt számitógépek /R-22, RC-3600/ és a kialakitott rendszerfejlesztő /szervezők,programozók/ szakember-gárda segitségével elértük, hogy ma már számos TSZ-ben támaszkodnak a számitógépek által nyujtott szolgáltatásokra. Az alkalmazási területek a gazdálkodás alapkategóriáinak adatfeldolgozását célozzák:

- anyag-fogyóeszköz
- állóeszköz
- munkaügy-bér
- pénz.

Az eddig megoldott és felsorolt területek közös jellemzője az igen nagy mennyiségü adat /esetleg többször 10.000/ és a ha - vinál nem gyakoribb feldolgozási igény. Az ügyviteli-gazdálkodási rendszereken túl készitettünk és müködtetünk állattartó telepek termelésirányitását segitő rendszereket is, melyek szarvasmarhatelepen és csincsillatelepen folyó termelő és te-

nyésztő munkát segitik. Ez utóbbi rendszerek jellemzője, hogy kevesebb adat /legfeljebb néhány ezer/ feldolgozását végzik, de a feldolgozás gyakorisága legalább dekád léptékü kell,hogy legyen. A számitógép szolgáltatta adatok felhasználhatósága és az események rögzitését jelentő adatfelvételezés közötti összefüggés rendkivül szoros, hiszen nincs elavultabb dolog, mint a telepi események megtörténte utáni "prognózis" a bekövetkező tartási eseményekre. A mezőgazdasági üzem és a feldolgozó számitóközpont közötti nagy távolság /az adatok utaztatása elkerülhetetlen/ egyes területeken az igazán hatékony számitástechnika alkalmazást nehezíti meg, mig más területeken lehetetlenné teszi a számitógép igénybevételét. Könnyen belátható, hogy a "párbeszédes" problémamegoldási kényszer a szakember és a számitógép közvetlen kapcsolatát feltételezi.

3./ Uj igények az adatfeldolgozó gép iránt

Az ügyviteli és ehhez kapcsolódó gazdálkodási munkát segitő számitógépes feldolgozások egy területen /pl. állóeszköz/ belül is oszthatók az egyes funkciók ellátásakor megkövetelt operativitási /nap, dekád, hó, stb./ szint szerint. Más területek /pl. pénzgazdálkodás/ nagyrészt napi feldolgozási gyakoriságot igényelnek. Az eddig felsorolt és további most nem részletezett okok uj igényeket támasztanak az alkalmazott számitástechnika iránt. Ezek az uj szempontok a következők:

- a felhasználók lényegében bármikor hozzáférjenek, vagyis a gép legyen a gazdaságban,
- adjon lehetőséget "párbeszédes" problémamegoldásra /a szakember feldolgozás közben véleményezhesse a részeredményeket/. Ez indokolt például egy takarmányösszetétel optimizálás esetén,
- a berendezés biztositsa a középgépek által előállitott másodlagos adathordozók /pl. lyukszalag/ feldolgozásának lehetőségét,
- beszerzési ára ne tegye elérhetetlenné a kisebb és közepes méretü üzemeknél sem.

Ezek az igények valamilyen kisszámitógéppel kapcsolatos elvárásokat körvonalazzák. Vállalatunk 1981 óta jelentős erőket mozgósitott a mikrogép alkalmazási igény kielégitésére. Tevékenységünk során elsősorban hazai gyártmányú mikrogépekre fejlesztettünk felhasználói rendszereket. Alkalmazott géptipusok: TAP-34, MØ8X, FLOPPYMAT-SP. Az egyes géptipusok az üzemi próbákon túljutottak, és 1984. évtől tömeges elterjesztésüket tervezzük.

TARTALOMJEGYZÉK

Kiadja: a Neuamnn János Számitógéptudományi Társaság
a Müszaki és Természettudományi Egyesületek
Szövetsége tagja

Szerkesztette: a II. Országos NJSZT Kongresszus Program-
bizottsága

ISBN szám: 963 8431 33 4 összkiadás 36 9

Készült: a KSH SZÜV nyomdában

SZÖV Nyomda 83,5365

F. v.: Antal Imréné