

DOKTORI ÉRTEKEZÉS

HALASSY BÉLA

1989

DOKTORI ÉRTEKEZÉS

**TERVEZÉSI MÓDSZERTAN
INFORMÁCIÓS RENDSZEREK
MODELLEZÉSÉRE**

**Készítette:
dr. Halassy Béla
Budapest**

1989

0. ÖSSZEFOGLALÓ

Ebben az összefoglalóban először a témaválasztást magyarázom. Majd ismertetem a kutatás körülményeit. Bemutatom a témakezelés módját, indokolva a dolgozat felépítését. Ezután röviden összefoglalom főbb mondanivalómat, a téziseket. A fejezet a munkában alkalmazott konvenciók felsorolásával zárul.

0.1 A 'SYDES'-GONDOLATRÓL

Minden tudományágban vannak ellentmondásos oldalak, ütköző koncepciók. Az eltérő nézetek és a tudományos viták jelentik a fejlődés motorját. Azonban ezeket az ellentéteket időnként szüneteltetni kell és a közös fejlődés érdekében szükséges harmonizálni az ellentmondásos oldalakat.

Elfogultság nélkül állíthatom, hogy a számítástechnika a legellentmondásosabb tudomány. Műszaki hátterét nézve természeti; céljait tekintve társadalmi; matematikai és nyelvi alapjait szemlélve módszertani diszciplína. A mérnök, a matematikus, a közgazda másként közelíti meg ezt az ismeretkört.

Az ellentétek legélesebbike a fejlesztésben mutatkozik meg. A felhasználók és a programozók nem beszélnek azonos nyelvet. Sőt, a különböző felhasználói és programozói csoportok belső érdekei is ütköznek. Ezért van szükség minden összetett rendszer fejlesztésénél külön rendszertervezőre.

Bár lényegét már tudományosan vizsgálják, a rendszertervezést még csak szakmának tekintik. Ennek oka egyszerű. A rendszertervezés sajátos ismeretköre - módszer. Tudományággá tehát csak akkor válhatna, ha lényege felől vizsgálnánk, vagyis **módszertanilag közelítenénk meg**. Úgy látom, hogy ezen a téren még óriási az űr.

Ezért alapvető célom az információs rendszerek tervezésének módszertani alapú vizsgálata és továbbfejlesztése.

Munkámban az 'e pluribus unum' elvét vallom. Kritikáimban nem az elmarasztalás a lényeg. A különböző elképzeléseket akarom harmonizálni úgy, hogy azokat 'megszüntetve megőrzöm'.

Alapállásomat leginkább az általam választott megnevezés fejezi ki. Információs rendszertervezési módszeremet 'SYDES'-nak hívom. Ez a betűszó a SYstem DEsigner System (rendszertervezési rendszer) kifejezés rövidítése. Olvasata (szájdísz) szerint oldalakat jelent.

Tudományos alapállásomnak az a lényege, hogy keresem az ellentétes nézetekben a közössé harmonizálható oldalakat.

Végeredményben ebben a munkában nem a részleteket tartom fontosnak. Azok csak arra szolgálnak, hogy alátámasszák a SYDES alapkoncepcióját, mint lényegét. Remélem, hogy ez az elv követőkre talál és egyben új tudományos kutatásokat indít meg.

0.2 A KUTATÁS KÖRÜLMÉNYEIRŐL

Nem kívánok a szubjektív szférába tévedni, de művem megértése érdekében röviden ismertetnem kell mindennapos munkámat.

1970-1982 között rendszerszervezőket tanítottam úgy, hogy 1972-1982 között vezetőként voltam felelős a rendszertervezési módszerek oktatásáért. Az oktatandó anyagokat igen gondosan kellett mérlegelni.

Mindez azt jelenti, hogy nemcsak óriási ismeretkészlettel találkoztam, hanem önkéntelenül is kifejlődött bennem egy elemzési, vizsgálati magatartás. Ezt az 1972-ben elkezdett publikációs tevékenységemben igyekeztem tudatossá érlelni.

1971 óta végeztem gyakorlati munkát, 1981-ig ad hoc jelleggel. 1982 óta - a legutóbbi évtől eltekintve - bevételkötelezett, gyakorlati munkából élő részleg vezetője voltam. Ebben az utóbbi időszakban tértem át a szoftverfejlesztésre is. Mindez azt jelenti, hogy tudományos vizsgálataimat azonnal a gyakorlattal kellett szembesítenem.

Tudományos tevékenységemnek igen sok részeredménye úgy folyt át a gyakorlatba, hogy nem jelent meg róla nyilvános publikáció.

Ugyanakkor itt kell utalnom arra, hogy ezek az eredmények elérték egy másfajta nyilvánosságot. Munkáim ma számos magyar egyetemen

és főiskolán alap-tananyagok, módszereimet pedig sok szervezetben alkalmazzák.

Kutatásom főbb állomásait a C.függelékben adom meg. Világos, hogy kutatási módszereim és körülményeim nem a szokványosak. Ennek van egy sajátos oka is. A témámban végzett vizsgálatok általában elméleti, azaz alapkutatások. Munkámból, funkcióból, szemléletemből fakadóan nem elégszem meg azzal, ha egy elmélet kerekén megfogalmazott, tudományosan igazolt.

Vizsgálataim gyakorlat-elméleti természetűk. Vagyis azt elemzem, hogy egy (jó) elmélet hogyan alkalmazható.

0.3 A TÁRGYALÁSMÓDRÓL

A tézis-antitézis-szintézis jegyében a tudományos dolgozatok nagy részében tétel(ek)re hivatkoznak, majd érvelésekkel kiegészítenek, illetve cáfolnak. Ezt az utat itt nem választhattam a téma átfogó jellege miatt. Ebben a témakörben csak implicit és átfogó 'tételek' vannak.

Innen fakad, hogy tételszerű megfogalmazásaim esetleg általánosítók, erőltetettek.

A hihetetlen mérvű tárgyalási anyag ésszerű átfogása és a deduktív gondolkodás érvényesítése érdekében; végül pedig azért, hogy ez a dolgozat a megírt 600 oldal helyett a célszerű 200 oldalba beférjen, olyan kettős megoldáshoz kellett folyamodnom, amely megfelel a 'SYDES' koncepciónak is.

Kettős szerkezetet választottam. A rendszertervezési módszerek vizsgálatait ma egy nemzeteket átfogó fórumot kaptak az IFIP ún. CRIS bizottságában (CRIS = Comparative Review of Information Systems [design methods]). Itt valóban alapos tudományos munka folyik [I12-13], amibe én is bekapcsolódtam [H29]. Munkám első három fejezete szorosan illeszkedik ehhez a kerethez. A felvezetés logikája az alábbi:

Először az információs rendszernek és annak tervezésének a lényegét mutatom be. Csak ezt követően jelenthetem ki, hogy a mai tervezési módszerek féloldalasak. Az első fejezet végén arra mutatok rá, hogy mi mindent kellene felölelnie egy valóban igényes, tudományos módszernek.

A második fejezetben az információs rendszer tervének a statikáját és dinamikáját mutatom be. Azt, hogy struktúrálisan hogyan épül fel a rendszerterv, mint metarendszer. Majd azt, hogy miként valósítandó meg dinamikusán a terv. A fejezet szoftver-logikai ismertetővel zárul.

A harmadik fejezet valójában egy definíció-halmaz. Ez azonban nem l'art-pour-l'art. Egyrészt a fejezet más koncepciókkal érvelően szembesít. Másrészt a fejezet végére felvázolódik egy fogalmi szintű információs rendszermodell.

Az átfogóbb gondolkodást igénylő fejezetek után a negyedik fontos részleteket tárgyal. Egyetlen témakörrel foglalkozik: a relációs normálformák adatmodellezési szerepével. A mondanivaló magva az, hogy a normálformák ugyan matematikai konstrukciók, de ha szemantikailag is értelmeznénk őket, akkor lényegesen többet nyújtanának a fogalmi adatmodellezésben. Mivel a normálformákkal könyvtárnyi irodalom foglalkozik, célszerűnek látszott, hogy tárgyalási keretként egyetlen művet alkalmazzak. Választásom az [I24]-re esett.

A dolgozat két érdemi függelékkel tartalmaz. Az A.függelék egyetlen célja az, hogy illusztrálja: a rendszertervezés tudományos kutatása valóban filozófiai mélységekbe vezet. A rendszertervezés nem csupán szakma, hanem valódi tudomány.

A B.függelék - mentegetődzés. SYDES-koncepcióm néhány részlete kísértetiesen hasonlít az RM/T relációs modellre. Azt kívánom bemutatni, hogy a SYDES eredeti gyökerű. (Nincs jogom azt mondani, hogy hatott az RM/T-ra. Tény, hogy akkori gondolataimat Codd úrral többszöri levélváltásban közöltem.)

0.4 AZ ALAPVETŐ TÉTELEKRŐL

Megállapításaimat két csoportba kell sorolnom. Némelyikük – nem tagadom - szinte (!) közhely. Csak arra szolgál, hogy felvezesse a gondolatokat. De vajon miként is lehetne másként elmondani azt, hogy mai tervezési módszereink rosszak, azért, mert...? Javításra szorulnak ott, hogy...?

Legáltalánosabb, bevezető kijelentésem az, hogy a ma széleskörűen alkalmazott rendszertervezési módszerek módszertanilag beszükültek

(S1). Technika- és eszközfüggők (S11); egyoldalú felhasználói szemléletek által korlátozottak (S12); a számítástechnikai divathullámoknak kitettek (S13). Vagyis két szóval: nem tudományosak.

Mindez arra vezethető vissza, hogy a tervezési alapkategóriák és főleg azok viszonyai nem átgondoltak (T21). A tervezésben nélkülözhetetlen absztrakció szintjei pontatlanok (T22). A tervezési algoritmusok beszűkültek, rugalmatlanok (T23).

Az általánosan megfogalmazott elméleti elvek a gyakorlatban használhatatlannak bizonyultak (T3). Az adatmodellt nem az integrált rendszer szervezőerejének, hanem szupertervezők elméleti játékszerének tartják (T31). Mivel az összefüggések igen szerteágazóak és felkutatásuk nemcsak fáradságot, hanem rizikót is jelent, a modellkonceptiók szemantikailag szegényes síkon maradnak (T32), főleg az alapjellemzők struktúrálása terén (T33).

Az adatszemplétű tervezőket - ebbe a körbe tartoztam jó ideig magam is - hosszú ideig 'elbűvölte' a normálformák és a normalizálás egzaktnek hitt matematikai koncepciója (T4). Erről kiderítettem, hogy elméletileg nem teljes (T41). Rámutattam a hagyományos csoportfogalom szükségességére (T42). Gyakorlati munkám is igazolta, hogy az elméletileg felvázolt normalizálási módszerek használhatatlanok (T43). Minden azt igazolja, hogy az adattervezésben a matematikával egyenlő szerepet játszik a szemantika, egymást kölcsönösen kiegészítő módon (T44). Az utóbbi kitétel és következményei számos segédttel vezethetők le.

Megállapításaim között vannak igen 'merészek' is. A 4.2.1 alpontban, például, megkérdőjelezem a mindenki által axiómaként kezelt additívítási Armstrong-szabályt. A dolgozat olvasójának itt is az értelmezésre kell figyelnie! Azt nem vitatom, hogy ha egy X tulajdonság meghatározza az Y-t és a Z-t, akkor megadja ezek együttesét is. Ahhoz viszont ragaszkodom, hogy amennyiben egy X jelenség megadja az Y nőnemű és Z férfi lényt, attól még X nem házasságot jelent! Vagyis az additívítás szabálya jó, érvényes, de csak adott szemantikai értelmezésben alkalmazható.

0.5 A HIVATKOZÁSI KONVENCIÓRÓL

Munkám **alapvető megállapításait**, tételeit kiemelten 'T' betűvel jelöltem. A számozás a fejezethez ill. ponthoz kötődik. A **kiegészítő megállapításokat**, segédteteleket 'S' betű mutatja. Lehetnek önállóak, vagy kapcsolódhatnak a tételekhez. A **meghatározásokat** 'D'-vel mutatom, de csak a legalapvetőbbeket, mivel a dolgozat számos definíciót tartalmaz. A kiemelten kezelt **példákat** fejezeten belül 'P' sorszámozza.

Hivatkozásaim kétfélék. A dolgozattal közvetlen munkakapcsolatban lévő művekre [Ixx] módon utalok, a munkámban való megjelenés sorrendjében. Saját kutatásaimat [Hyy] jelöli, kronológikus sorrendben. Itt kell megjegyezni, hogy az irodalomjegyzékben nem állt módomban felsorolni mindazokat az írásműveket, amelyek közvetve, de lényegesen hatottak elképzeléseimre. Egy részükre utalás történik publikációim forrásainál. Végeredményben e dolgozat mögött több száz publikáció több tízezer oldala áll.

A fogalmazással kapcsolatosan kell megjegyezni, hogy a rendszer, a rendszertervezés, a terv, a tervező stb. kifejezések elé mindig odakiváncozik az 'információs' jelző. Nem hanyagságból, hanem a rövidség kedvéért hagytam el a teljes megnevezést sok helyütt.

Köszönetet mondok a családomnak. Ők jelentették számomra a hajtóerőt. Munkatársaimnak, Lévai Máriának és Nagy Erzsébetnek. Ők végezték az aprólékos munkát. Vállalatomnak, a SZÁMALK-nak, amely biztosította számomra a kutatás – nem csak a dolgozat – technikai és anyagi feltételeit.

1. A TERVEZÉSI MÓDSZEREK KORLÁTAI ÉS KELLÉKEI

Munkám címe egy tervezési módszertan bemutatását ígéri. A módszer kifejtése előtt legelőször arra a kérdésre kell választ adnom, hogy egyáltalán **miért van szükség** egy sokadiknak mondható módszerre? Milyen tényezők ösztönöztek a módszertani kutatásra? Folytatólagosan: **Milyen célokat** kívánok elérni az új módszertan megalapozásával? Végeredményben: Milyen okok vezéreltek ennek a műnek az összeállításában?

Az elmúlt csaknem két évtizedben számos információs rendszer tervezési elvet, megoldást, gyakorlati eljárást, segédeszközt stb. tanulmányoztam át [I01-10]. Vizsgálataim során meg kellett állapítanom a következőt:

S1 A ma széleskörűen alkalmazott rendszertervezési módszerek elméletileg súlyos módszertani korlátokat mutatnak.

Ezt a kijelentést csak kiindulási alapnak szántam és sietek hozzáfűzni a következőket. A tervezési módszerek sokasága ma már nem teszi lehetővé azok teljes áttekintését. Akkor sem, ha erre komoly kísérletek történtek [I11,12], amelyeknek magam is aktív részese voltam [H29]. Másrészt az S1-hez hasonlóan azonnal tehetnék egy másik kijelentést is, mely szerint elméletileg és részleteiben is teljes rendszertervezési módszertant lehetetlen alkotni.

Az S1 tehát minősítendő. A bevettnek tartott módszerek alkalmazásával nem-integrált, változásérzékeny rendszerek születnek. A **módszertani korlátok** következtében az információs rendszerek fejlesztése és fenntartása erőforráspazarlást okoz fejlesztőnél és felhasználónál egyaránt. Vizsgálat alá kell vetni a jelenség megnyilvánulásait. Keresni kell a hiányosság mélyebb okait. Ezeknek függvényében kell meghatározni a tervezési módszerek tökéletesítésének céljait és módjait.

A vázolt gondolatmenetnek megfelelően ebben az első, bevezető jellegű fejezetben a következő témákat fogom tárgyalni:

- Először az információs rendszerre adok meghatározást.
 - Majd a rendszertervezés fogalmát igyekszem tisztázni.
 - Ezután tervezési szemléleti torzulásokra térek ki.
 - Összefoglalom a tervezési módszerek 'kellékeit.'
- Ennek kapcsán kutatási céljaimat összegzem.

Az érdemi tárgyalás előtt egy fontos megjegyzést kell tennem. Az ilyen műhöz képest a bevezetés talán kicsit hosszadalmasnak fog tűnni. Törekszem ui. az alapfogalmak pontos körülhatárolására. Az informatika - határtudomány. Ha célját és közegét nézzük (az embereket információkkal ellátni), akkor társadalmi. Eszközeit tekintve (számítógépek) viszont inkább műszaki, természettudományi. A fogalmak és összefüggéseik sokfélesége valószínűleg így nem engedi meg, hogy gondolataimat az általam is kívánt pontossággal adjam át.

1.1 AZ INFORMÁCIÓS RENDSZER LÉNYEGE

Az információs rendszer alapos meghatározása külön disszertációt igényelne. Az alábbi munkadefiníciónak ezért csak annyi a célja, hogy felvázoljam az információs rendszerre vonatkozó koncepciómat.

D11 A SYDES-koncepcióban az információs rendszert

- adatok (információk);
- a rájuk vonatkozó információs események;
- az ezek által határolt információs tevékenységek;
- az előzőekkel kapcsolatos felhasználók és erőforrások;
- valamint a mindezeket szabályozó szabványok/eljárások

szervezett együttesének tekintem.

Bár ezt a definíció impliciten megteszi, explicit módon is ki kell jelentenem, hogy az információs rendszert nem azonosítom a szoftverrel. Ennek megfelelően az információs rendszer tervezésében a szoftver-technológia csak adott részt képvisel és a két fogalom nem

fedí egymást, amint azt sokan vallják. A 'szervezett' szó jelzi, hogy az adott tényezőket egyenragúaknak, egyenlő fontosságúaknak tartom.

Az egzakt kifejtés igénye nélkül a meghatározásban felsorolt tényezők mindegyikéről teszek néhány kijelentést azért, hogy a SYDES-felfogás alapjait megvilágítsam.

1.1.1 ADATOK ÉS INFORMÁCIÓK

Ebben a műben az adat és az információ kifejezéseket felcserélhető szinonimákként fogom használni, a mindennapos gyakorlatnak megfelelően. Le kell azonban szögezmem, hogy a két fogalom mögötti lényegyet nem tartom azonosnak. Ennek a következményeiről lesz még szó a módszertan bemutatása során.

A [H14]-ben kifejtettnek megfelelően a valóság nem értelmezett, de értelmezhető tükörképét nevezem **adatnak**. Mint ilyen, az adat közös tulajdon, széleskörűen használható erőforrás. Viszont **információnak** az adatösszefüggéseken végrehajtott értelmezés eredményét tekintem.

Mivel az értelmező személyek ismeretei, funkciói, céljai stb. eltérőek, ugyanazt az adatösszefüggést más-más információvá értelmezhetik. Az információ tehát csak szűk körben használható erőforrás.

A fentiek nem definíciók és ekképpen nem vita tárgyai. Csak szemléltetik a tervező feladatát. A tervezőnek fel kell kutatnia és meg kell határoznia azokat az adatösszefüggéseket, amelyeket a felhasználó könnyen a kívánt információvá tud értelmezni. Az adattal szemben az információ szubjektív lényeg. Ezért a tervező feladata nem csupán technikai jellegű.

1.1.2 INFORMÁCIÓS ESEMÉNYEK ÉS TEVÉKENYSÉGEK

Az információs tevékenységek rendkívül sokfélék. Ebben a műben nem lesz szó olyan általános társadalmi tevékenységekről, mint az informálás, a tömegkommunikáció stb. Nem térek ki a technikai segédtevékenységekre (másolás, adatátvitel, rendezés stb.) sem. A továbbiakban **információs tevékenységnek** azokat a munkálatokat nevezem, amelyeknek közvetlen feladata a felhasználó által igényelt

adatösszefüggések tartalmi összeállítása ill. ennek biztosítása. Hétköznapi értelemben ezt szoktuk 'adatfeldolgozásnak' nevezni.

Az információs tevékenységet **információs esemény** váltja ki. Hagyományosan az esemény fogalmát valamilyen változáshoz kötik. Szűkebb értelemben az adategységek ún. életciklusához; vagyis azok létrejöttéhez, módosulásához és megszűnéséhez. Tágabb értelemben a feltételrendszerben/körülményekben bekövetkező változás is esemény. Így pl. az idő- vagy az állapotváltozás is tevékenységet indít el ('trigger').

A SYDES eseményfogalma az előzőnél két szempontból is tágabb. A tudatos felhasználó nemcsak nyomon követi a változásokat, hanem elemzéssel, tervezéssel azoknak elébe is megy. Információfeldolgozást indít, hogy a valóságot befolyásolhassa. Ezt a konkrét igényt - pl. azt, hogy a felhasználó leül a terminálhoz - szintén eseménynek tekintjük. (Nijssen úr szóbeli úton gondolatom helyességét megerősítette az [I13] vitáján.) Másrészt a tevékenység maga is változásokat okoz, ill. újabb igényeket generál. Ezért nemcsak az információs tevékenységet kiváltó, hanem az azt lezáró jelenség is esemény.

1.1.3 FELHASZNÁLÓK ÉS ERŐFORRÁSOK

A **felhasználókat** két nagy csoportba szokták sorolni. A rendszer kialakítását végző fejlesztőket éppúgy felhasználóknak tekintem, mint az információkat szolgáltató, illetve alkalmazó végső-felhasználókat. Ennek nem az az oka, hogy a kellemes fejlesztési eszközök miatt a két réteg átfedőnek tűnik. Ez csak felszín. Egyre bonyolultabb szoftverek jelennek meg és ezért a két réteg valójában távolodik egymástól. Viszont a tervezési módszereket és eszközöket a két réteg csak közösen használhatja sikerrel!

Bár az információ személyes 'tulajdon', a közös háttérű, célú és funkciójú személyek csoportjait, szervezeteit szintén felhasználóknak tartom. Hiszen feltételezhetően közös adatösszefüggésekből merítenek információt. Ezzel szemben a felhasználók köréből kizárom az automatákat. Ezek ui. nem tudnak adatot információvá 'értelmezni'. A folyamatirányítás egyébként is az itt tárgyalandóktól (részben) eltérő módszertani kérdéseket vet fel.

Tág értelemben a D11 valamennyi tényezőjét **erőforrásnak** szokták tekinteni. A továbbiakban az erőforrásnak a szűkebb fogalmával élek. Az információkat feldolgozó tevékenységek végrehajtásához illetve az információs rendszer kialakításához szükséges technikai és pénzügyi eszközök az erőforrások.

1.1.4 SZABVÁNYOK ÉS ELJÁRÁSOK

Az előző három alpontban felsorolt tényezőket korlátok, megkötések, előírások, feltételek, egyszóval **szabványok**, illetve cselekvési szabályok, azaz **eljárások** irányítják. (Az eljárás itt nem programrészt jelent!) A szabványok/eljárások lehetnek külsők (a rendszert kívülről meghatározóak) és belsők. Lehetnek tartalmiak és formaiak. Általánosak és kivételesek.

A szabványok/eljárások a tervezésben hármass szerepet töltenek be. Irányítják magát a tervezést. A tervezőnek be kell építenie azokat a rendszer tervébe. Mint a rendszer elemei, a valós információs rendszer működését is vezérlik. Ezért tartottam fontosnak e tényező szerepének hangsúlyozását.

1.1.5 AZ INFORMÁCIÓS RENDSZER VETÜLETEI

Az információs rendszer három **vetületét**, aspektusát különböztetem meg a SYDES-koncepcióban. Ezek a következők:

- adat/információ;
- feldolgozás;
- környezet.

A D11-re vonatkoztathatók e vetületek. Az adatvetület a felsorolás első elemét tartalmazza. Az események és a tevékenységek tartoznak a feldolgozásvetületbe. A felhasználók, erőforrások és szabványok alkotják a környezetvetületet.

A megkülönböztetésnek komoly módszertani indokai vannak. A vetületek sajátosságaira a következő két fejezet impliciten rámutat. Itt

előzetesen annyit jegyzek meg, hogy a környezet vetület a másik kettőtől az absztrakció módjában tér el, tervezési szempontból. Az adat- és a feldolgozásvetület közül, ugyancsak tervezési szempontból, az előbbi képviseli a statikusabb és objektívabb oldalt.

1.2 INFORMÁCIÓS RENDSZERTERVEZÉS

A rendszertervezés feladatát, tevékenységét, illetve ennek a részeit, aspektusait a gyakorlatban számos kifejezéssel illetik. Használjuk, például, a fejlesztés, szervezés, specifikáció, elemzés, kialakítás, felépítés, konstruálás, menedzsment stb. kifejezéseket. Ebben a fejezetben a tervezés fogalmát csak körülírhatom. Határozottabb definíciót csak a SYDES-koncepció bemutatása után (3.fejezet) adhatok. Munkameghatározásként álljon itt a következő:

D12 **Információs rendszertervezésnek** az információs rendszer (D11) tényezőinek a feltárását, meghatározását és tudatos összehangolását nevezem.

Az alábbiakban a tervezés fogalmát megpróbálom körülírni. A magyar nyelv adottságai miatt kénytelen leszek angol terminológiához fordulni, aminek viszont szintén vannak buktatói.

1.2.1 TERV (PLAN) ÉS TERV (DESIGN)

Az angolban külön kifejezés illeti azt a tervet, ami valaminek a megvalósítási módját jelöli (plan) és magának a megvalósítandó dolognak a tervét (design). Magyarban talán a kivitelezési terv és a tervrajz felel meg e két fogalomnak. Ez nemcsak játék a szavakkal. A tervezési módszerek többsége a tervezési feladat - 'projekt' - irányítására koncentrál anélkül, hogy a terv (design) megalkotásához módszertani segédletet nyújtana. (Tipikus példa erre a sokak által használt SDM [I05], ahol az ember elveszik a papírok kitöltése közben.)

A SYDES-koncepcióban a rendszer ki- vagy átalakítására szolgáló tevékenységek együttesét hívom **fejlesztésnek** (angolul: development). A fejlesztés irányítására szolgáló elképzelés a **projektterv** (plan). A rendszer képi kialakítását hívom **rendszertervezésnek** (design). A fejlesztés kétféle tervezést foglal magában, de nemcsak tervezésből áll.

Az információs rendszerek fejlesztésének a szakaszait sokféle módon meghatározták. Annyi bizonyos, hogy a különböző szakasznevezések és -tagolások végeredményben mindig azonos lényegre takarnak. Az alábbi szakaszolást sem a vita célzatával, hanem a továbbiak megalapozásaként adom közre:

- célkitűzés;
- elemzés;
- tervezés;
- specifikáció;
- kialakítás;
- bevezetés;
- működtetés.

A szakaszokról alább lesz szó. Itt megjegyzem, hogy a 'fejlesztésnek' titulált módszerek/eszközök egyáltalán nem azok, mivel e szakaszok nem mindegyikét támogatják. A SYDES 'csak' tervezési módszer. Azt, hogy ez mit jelent, a következő alpontok világítják meg.

1.2.2 ELEMZÉS ÉS TERVEZÉS

Időnként viták lángolnak fel arra nézve, hogy a rendszeralkotásban az **elemzés** (analysis) és a **tervezés** (design) azonos vagy eltérő tartalmú lényegeket-e. Jellemző, hogy a CRIS hol azonosságuk [I11], hol eltérésük [I12] mellett kardoskodott. A zavart fokozza, hogy a 'rendszertervezőt' az angol irodalom hol systems-designer-nak, hol systems-analyst-nek nevezi.

A problémát a SYDES-ban az alábbi módon oldom fel. Különbséget teszek elemzés, mint fejlesztési szakasz és mint általános gondolkodási tevékenység között. Mint szakaszmegjelölés, az elemzés problémafeltárást, értékelést jelent. Mint tevékenység az elemzés azt

jelenti, hogy a tervező bizonyos kritériumoknak megfelelően fogalmazza meg, illetve azokkal összevetve rögzíti a tervét.

A rendszer nemcsak a tervelemeinek összehangolatlansága miatt működhet hibásan. Az elemzési szakaszban szubjektív hibákra is fény derülhet. A SYDES nem a szakaszt, hanem a tevékenységet támogatja. Meglévő rendszerek objektív struktúrálási hibáit éppen úgy feltárja, mint a tervezett rendszerek hiányosságait. Konceptiómban a tervezés kritériumoknak megfelelő kialakítást jelent. (ld. 1.4.4 alpont.)

1.2.3 TERVEZÉS, SPECIFIKÁCIÓ ÉS KONSTRUKCIÓ

A hagyományos módszerek különbséget tesznek a 'lényeg' meghatározása (tervezés) és a részletek kifejtése (specifikáció) között. Az, hogy általában milyen jellegű tényezők jelentik a lényegét és mik a részletek, rendszerenként változhat. Így kialakult az a gyakorlat, hogy 'lényeg'-nek a főbb struktúrális összefüggéseket, részletnek pedig a kifejtést, a leírásokat tekintették. Egy szélsőséges, abszurd példával szemléltetem az effajta logika helytelenségét.

P11 Biztosítások kezeléséről van szó. A tervező a programleírásokban adja meg a módozatok feldolgozását. Holott a biztosítási módozatok, jó tervezésnél, befolyásolhatják az adat- és program- szerkezetet (a lényegét) egyaránt.

A SYDES-ban nem teszek ilyen különbséget lényeg és részlet között. Helyettük a pontosabban meghatározott (fogalmi, logikai, fizikai) **tervezési szintek** koncepcióját alkalmazom (ld. a 2.fejezetet!). Az sem zárja ki az iteráció esetleges szükségességét. Előnye viszont, hogy a lényeges részleteket az elhanyagolható struktúrális 'lényegeket' előtt meg lehet adni.

A tervezési módszer lezárul a specifikációval. A **megkonstruálás** már túlmutat rajta. Vannak azonban olyan fejlesztési segédeszközök, amelyek - módszerbe ágyazottan - generálják a majdani szoftver részeit. Prototípust alkotnak, tesztadatokat generálnak. A SYDES is errefelé fejlődik. A mai kísérletek azonban azt bizonyítják [I14], hogy a tervből automatával csak a legprimitívebb funkciókra készíthető közvetlenül működő program (rendszerész).

1.2.4 TERV, TERVEZÉS ÉS DOKUMENTÁCIÓ

A magyar nyelv terv (plan) és terv (design) között nem tud különbséget tenni. Az angol pedig a tervezés (design) és a terv (design) között. Ezért az [I15] jogosan vezette be a következő terminológiát, amit a SYDES-ba én is átvettem.

Terv-terméknek (design-product), vagyis **tervnek** nevezzük a tervezési munka eredményét. Ez mindenféle leírásokban testesül meg, amiket közösen terv-dokumentációnak nevezünk.

Tervezési-folyamatnak (design-process), vagyis **tervezésnek** hívjuk a terv megalkotásához szükséges tevékenységsorozatot. Ez nem azonos a fejlesztéssel. A tervezési algoritmust öleli fel (lásd az 1.4.5 alpontot!).

A logikus ember úgy véli, hogy a tervezési-folyamat végeredménye, a terv-termék valóban terv-**dokumentáció** formáját ölti úgy, hogy a terv valósággá is válik. A legkorszerűbb tervezési módszer esetében is mutatkozni fognak eltérések a szándékolt és a megvalósult terv között. Mai módszereinknél már a szándékolt és a dokumentált terv sem azonos. Módszeremben a kettő közelítésére törekszem. A végcél egy olyan információs erőforrás-szótár, amely nemcsak, hogy maximálisan megfelel a tervnek, de egyben az élő rendszert is vezérli.

1.2.5 TERVEZÉS ÉS SZERVEZÉS

Az angolszász rendszertervező vagy -elemző kifejezés helyett a magyarban a rendszer- és folyamatszervező páros terjedt el. Az utóbbi annyira értelmetlen, hogy szót sem érdemel. A **szervezés** személyek, szervezetek és erőforrások összehangolását jelenti. A projektirányítás kimondott szervezési feladat. A rendszer életciklusában a bevezetés és a működtetés igényel különösen alapos szervezést. Munkámban azonban nem térek ki a velük kapcsolatos tevékenységekre.

1.2.6 TERVEZÉS ÉS MODELLEZÉS

Az alábbiak majd mutatni fogják, hogy a **modellezés** és a tervezés szinte azonos fogalmak. Annak, hogy nem szinonimákként használom

őket, két oka van. Első a félreértés. Egyes szakemberek a modellezést a tervezéssel azonosnak tartják és így fontosságát - mi már ismerjük címen - lebecsülik. Valami új varázsszót ('buzzword') sejtenek csak mögötte. Mások viszont éppen az új fogalmak miatt megrettennek, gyökeresen újat várnak és a félelem gátolja őket az új módszer elsajátításában.

A másik ok az angol 'design' szó harmadik értelmében rejlik. Ez magyarul iparművészetet, **formatervezést** is jelent. Azaz a rendszert nem elég megtervezni egyszer. Nem elég egy működőképes tervet alkotni. A tervnek 'elegánsnak' is kell lennie! A modellezés – tervezés, sőt több: eleganciára törekvő, magasfokúan tudatos tervezést jelent.

1.3 TERVEZÉSI SZEMLÉLETI TORZULÁSOK

A mai tervezési módszerek megalkotói és alkalmazói az összetett információs rendszerek lényeges tényezőit és összefüggéseit szemlélik egyoldalúan. A felhasznált módszerek korlátai miatt a tervező - legtöbbször tudtán kívül - háromféle szemléletzavarban szenved. Ezek a zavarok a terv(ezés) három dimenziójával függenek össze. Ebben a pontban a következő szemléleti torzulásokra mutatok rá:

- eszköz-meghatározottság;
- feladat-kötöttség;
- időszakos divatorientáció.

1.3.1 ESZKÖZ-MEGHATÁROZOTTSÁG

A mai átlagos tervező az információs rendszerterv részleteinek a kialakításakor elválaszthatatlanul összeköti a kívánt tartalmat a megvalósítás módjával. Ezt a jelenséget fizikai (ábrázolási, eszköz, hozzáférési stb.), másképpen vertikális függőségnek nevezzük.

S11 A vertikális függőség hordozhatatlan, szuboptimális, merev információs rendszerhez és erőforrásvesztésre vezet.

Nyilvánvaló, hogy a tervezőnek az adott tervrészlet kialakításakor tekintettel kell lennie a megoldás technikai lehetőségeire is, amelyek esetleg visszahatnak magára a feladatra. A problémát az 'elválaszthatatlanul' szó mutatja. A tervező magában a tervezési folyamatban és a végső tervmeghatározásban annyira összekapcsolja a megoldandó feladatot a kiválasztott megoldási formával, hogy az előbbi elveszti saját lényegét. Ebből a következő gondok fakadnak:

- **Hordozhatatlanság.** Ma gyakran változik a technikai környezet a műszaki fejlődés következtében. Egyre jellemzőbb, hogy egy felhasználói környezetben is eltérő adottságú eszközöket használnak. Ha a tervező a tervrészletet 'bedrótozza', akkor környezetváltáskor az egyébként változatlan tartalmat is újra kell elemeznie, át kell terveznie.
- **Elemzési hiány.** Mint majd bizonyítani fogom, a rendszer elemeit és azok összefüggéseit lehet és kell elemezni eszköztől függetlenül. Viszont ha a tervező 'ab ovo' technikai részletekbe ágyazza tervét, akkor az elemzés vagy elmarad, elkésik, vagy a fizikai tényezők lefejtési igénye következtében nehézkessé ill. lehetetlenné válik.
- **Munkatöbblet.** Kis példával szemléltetem ezt a tételt. A tervező formailag is részletezett (!) kimenetekre alapozta a tervét. Ha fény derült valamelyik adattétel helytelen értelmezésére (tartalom), akkor a tervezőnek többletmunkával újra kellett rajzolni a (forma) a kimenetet. Ez a többlet elkerülhető, ha a tartalom alapos elemzése megelőzi a forma gyors, elkapkodott meghatározását.
- **Változások.** Az előző probléma fennáll a teljes dokumentációra is. A hagyományos rendszertervben a tartalom ('mit') feloldódik a megvalósításban ('hogyan'). Ha netán csak az utóbbi változna, akkor is újra kellene tervezni és dokumentálni a tartalmat is.
- **Kommunikáció.** A tervezők nagy része csakis technikailag kötött kifejezőkészlettel képes közölni tervét a felhasználóval. Az ezt visszautasítja, vagy - és ez még rosszabb - álmegértést mutat. Ha a tervező technikai trükkökbe ágyazza a feladatot, akkor azt más tervező - később még saját maga - sem fogja megérteni.

Az eszközmeghatározottság súlyát két tényező még növeli is:

A képzéshiány. Ma is lyukkártyás (hierarchikus) kódot és szalagos (szekvenciális) szerkezetű állományokat tervezünk. Adatbáziskezelőket használunk egyszerű célokra és relációsan próbálunk megtervezni egyetlen adatállományt/programot.

A módszerhiány. A hardver fejlődése féltucat évvel előzi meg a szoftverét; azé pedig egy tucattal a módszerét. (Pl.: Lassan lesz egy normális dBASE szoftver kézikönyv. A tervezésinek talán még a gondolata sem született meg...)

A fizikai függőség szembeötlő problémái csak egy úton oldhatók fel: a tervezési szintek tudatos szétválasztásával.

1.3.2 FELADAT-KÖTÖTTSÉG

A mai tervező a rendszerterv **belső részleteiben** is az általa kiszolgált végső-felhasználó kívánságaira alapoz. Ezt a jelenséget **logikai** (funkcionális, szemléleti, kimeneti.), másként **horizontális függőségnek** nevezzük.

S12 A horizontális függőség egyoldalú, redundáns, merev információs rendszerhez és erőforráspazarláshoz vezet.

Nem kétséges, hogy a tervezőnek ki kell elégítenie a felhasználó ésszerű igényeit. A problémát itt a 'belső részleteiben' kitétel mutatja.

A tervező a tervezési folyamatban és a terv meghatározásában közvetlenül (!) összeköti a felhasználói igényből fakadó külső struktúrákat (kimenet) a belsővel (állomány). Ebből a következő gondok keletkeznek:

~ Egyoldalúság. A problémát szemléltesse egy példa! A tervező azértékesítési részlegnek dolgozva tervében a rendeléseket és azok cikkekre vonatkozó tételeit szorosan összekapcsolja. Így könnyen kikereshetők az egy rendelésben érintett cikkek. Arra a kérdésre viszont - ami mellelleg izgalmasabb, gazdálkodóbb színezetű -, hogy egy cikket milyen rendelésekben kértek, nehézkes a válasz.

~ Párhuzamosság. Folytassuk az előző példát! A rosszul kiszolgált raktárgazdálkodó úgy fog dönteni, hogy saját rendszert épít. Ezzel az adatok és műveletek (adatbevitel) megkétszereződnek. Semmi sem garantálja a párhuzamos rendszer-részekben az időbeli egyeztetést (eltérő 'korú' információpéldányok találhatók az ilyen 'rendszerekben'). Semmi sem zárja ki, hogy a két rendszer között 'szürke-foltok' keletkezzenek azokon a területeken, amelyekért egyik felhasználó sem érzi magát elsődleges felelősnek.

~ Merevség. A felhasználó nem érzi - a tervezőnek tudnia illenék -, hogy szemlélete majd változni fog. Ma a dolgozóknak csak kéthavi béradatára kíváncsi. Ha a tervező a rendszerrészletbe ezt a korlátot struktúrálisan is beépítette, akkor a holnapi, tizenkéthavi béradatra vonatkozó igény vagy nem teljesül, vagy duplikációt, ill. dupla munkát okoz.

~ Átszervezés. Az egyoldalúság, párhuzamosság és merevség egy idő után akkora feszültséget okoz, hogy a vezetés megűnja az erőforrás-pazarlást, döntés-képtelenséget. Elrendeli az újraszervezést és ezzel - ugyanezen módszerek használata esetén - a problémákat magasabb szintre helyezi.

Megjegyzés: A fenti gondolatsor híven tükrözi a mai információs rendszerek életútjának állomásait is.

A problémák, a logikai függőségek kettős gyökerűek.

~ **Áttekintéshiány.** A felhasználót és a tervezőt mi sem kényszeríti arra, hogy a mai problémákon időben és térben túltekintsen. Sőt, érdekeltségi rendszere az átlátás ellen szól.

~ **Módszerhiány.** Az átlagos tervező nem kap képzést arra, hogy az egyedi, változékony felhasználói szemléletből miképpen vezethet le egy általánosabb, stabilabb tervrészletet.

A logikai függőség nyilvánvaló problémái csak egy úton oldhatók fel: a globális és az egyedi szemlélet szétválasztásával.

1.3.3 DIVATORIENTÁCIÓ

Mivel a gyakorlati életben túl nagy a súlyuk, az elméletben sem hallgathatók az időközönként fel-felbukkanó 'módszertani' divatokról.

Az idézőjellel azt kívántam jelezni, hogy szerintem nem valódi módszertanokról van szó, legfeljebb részmodszerekről vagy még inkább: szokásokról. Ezek a szokások részben eszközfüggők, részben módszerekhez tapadnak. Nincs közös vonásuk: csak jönnek és letűnnek. Ezért joggal hívom a jelenséget divatfüggőségnek.

S13 A divatorientált módszerek a logikai és fizikai függés negatívumait implikálják egyoldalúságuk következtében.

Mielőtt az elfogultság vádja érne, sietek leszögezni, hogy ebbe a hibába magam is beleestem. A divatorientációban egyébként két problémát látok. Egyrészt azt, hogy az adott módszerhez szokott tervező **minden feladatra** azt akarja alkalmazni mérettől, eszköztől, környezettől, felhasználótól függetlenül. Ilyen az, amikor valaki egy program egyetlen adatállományát is normalizálni kívánja. A másik baj az, ha valaki nem látja a módszer általános **elvi korlátait**. Ha például adatbázis-lekérdező programot is 'struktúráltan' tervez.

Az alábbiakban néhány divatos koncepciót tekintek át, a teljesség igénye nélkül. Céлом mégcsak nem is a kritika: ahhoz alaposabb elemzést kellene nyújtanom. Pusztán a divatorientációból fakadó egyoldalú szemléletekre kívánok példát mutatni.

- **Struktúrált.** A struktúrált programozás és tervezés nagy lépést jelent a feldolgozás változás-érzékenységének a feloldása, a könnyebb programkövetés felé. Hiba lenne azonban nem látni, hogy az elv a fizikai függőséget nem csökkenti, a logikait pedig egyenesen növeli. A struktúrált elv nem ad módszert – csak technikát, és az nem ugyanaz! - a valóságban hálósan kapcsolódó tevékenységek hierarchikusnak célzott lebontására. Példa: Vajon az értékesítési vagy a pénzügyi funkció része-e a számlázás? Hasonlóképpen: Egy relációs adatbázis lekérdezésének a tervezésénél a struktúrált programozás nem vezet jó megoldásra.

- **Adatmodellezés.** Az adatbázis-tervezési módszerek megmentenek bennünket a programfüggőségtől. Sokáig nem vettük észre, hogy ennek áraként kidobtuk a régi feldolgozástervezési módszereket, ill. nem alkottunk ilyeneket az 'elavultak' helyett. Kiderült, hogy az adatmodellezés valahol mégis csak eszközfüggő és fizikai függést generál. (Azáltal, hogy az adott eszközhöz egyéb tervezési elveket nem fogalmaznak meg. A dBASE esetén, például, a relációs tervezés csődöt mond, mert minden az indexeken múlik.) A korszerű nek hitt normálforma elvek pedig revízióra szorultak.
- **Fejlesztési eszközök.** Itt a program- és alkalmazásfejlesztési eszközökre és az ilyenek mondott xGL-re gondolok. Nem is alapos vizsgálattal kideríthetjük, hogy a programfejlesztési eszköz csak kódolásra való. Az alkalmazásfejlesztéssel már teljes feldolgozást tervezhetünk. Azonban program- és főleg adatrendszert egyáltalán nem. A 4-5GL pedig még a nevében sem ad módszert.

A problémát az jelenti, hogy a módszertani hiányban szenvedő, a módszerekre éhező fejlesztők kapva-kapnak minden 'módszerszerű' valamin, annak lényegének átgondolása nélkül. Ezek a gondok csak egyféle úton oldhatók fel. Olyan általános módszertani képzéssel, amelynek **kereteibe** tudja illeszteni az egyedi fejlesztő is a mindenkor kézhez kapott 'technológiát'.

Nem utolsósorban a 'módszeresség' alapvető feltétele az is, hogy a tervezők reális projekt-határidőket kapjanak. Ha a felhasználó nem akar igényes rendszert, a vezető pedig komoly, érett munkát, akkor mi ösztönözzön a jó módszerekre?

1.4 A MÓDSZEREK KELLÉKEI

A módszerek hiányosságaiért végülis nem az azokat felhasználó tervezőket, hanem a módszeralkotókat kell hibáztatni. A technikai és társadalmi körülmények mindig gyorsabban fognak változni, mint a módszerek. Egyébként is annyira összetettek, hogy egyetlen, mindent átfogó, minden helyzetben alkalmazható módszer elvi képtelenség.

Hibás elvektől mentes, kerek, az egyéb részmodszerek befogadására is képes, a szükséges kelléktár terén átgondolt módszertani **keret** azonban nem elképzelhetetlen. Ilyen keretet nincs szerencsém ismerni, de céloban van megalkotni. Az első lépésként azt kell meghatároznom, hogy mik a módszerek **kellékei**. Vagyis azt, hogy mi mindenre kell(ene) figyelnie a módszeralkotónak és -alkalmazónak egyaránt.

Hét ilyen kelléket azonosítottam [H29]. Ezeket a továbbiakban sorban, egyenként bemutatom. Listájuk a következő:

- alapkoncepció;
- struktúra/terminológia;
- megfogalmazási mód;
- kritériumrendszer;
- tervezési algoritmus;
- dokumentációs forma;
- technikai támogatás.

1.4.1 ALAPKONCEPCIÓ

Alapkoncepciónak azon elképzelések magvát nevezem, amelyeknek megfelelően a módszertan-alkotó és -alkalmazó a módszer tárgyának (esetünkben: az információs rendszernek) az alapvető tényezőit, illetve azok összefüggéseit szemléli. Az információs rendszer különböző természetű tényezők szervezett, inhomogén együttese, amint azt az 1.1 alpontban kifejtettem.

Alapkoncepció szempontjából részleges módszernek kell tekintenem az olyan szoftver-technológiát, amely az információs rendszert programoldalról akarja megragadni és nem törődik, például, az emberek közötti kommunikációval. Féloldalasak az adatmodellezési módszerek, ha az adatösszefüggések vizsgálata háttérbe szorítja a feldolgozáselemzést. Csak részmodszerek a struktúrált tervezés, amely adatszerkesztésre nem ad megoldási módokat. Nem kell kifejtenem, hogy mennyire egyoldalúak az ún. 'fejlesztési' eszközök, amelyek az információs rendszert nem totális egészként szemlélik.

Minden módszernek sarkköve az alapkoncepció. A SYDES-módszerben arra törekszem, hogy az információs rendszert teljes egységként ragadjam meg, mint módszerem tárgyát.

1.4.2 STRUKTÚRA/TERMINOLÓGIA

Az információs rendszer inhomogén együttes. Stabil információ-szerkezetet, a dinamizmust képviselő tevékenységstruktúrát, mozgatóerőkként működő felhasználókat stb. ölel fel. A horizontális vetületeken túlmenően az információs rendszer vertikális szintjeit is megkülönböztetjük. Ez a rendszer ui. absztrakció eredménye, és az absztrakció lehet több fokozatú.

A tudatos módszert alkalmazó tervezőben élnie kell egy struktúrális képnek. Arról, hogy az információs rendszer tényezői - pl. az adat- és tevékenység egységek - miképpen függnek össze egymással. Nemcsak a vetületek között, hanem azokon belül is kapcsolatokat, vagyis **struktúrákat** kell elképzelnie. Tisztában kell lennie a szintek közötti viszonyokkal is. Pl. azzal, hogy miképpen képződnek le a fogalmi szintű egyedek logikai szintű rekordokra vagy adatszegmensekre.

Az információs rendszer tényezőtípusait és azok tipikus viszonyait a különböző módszerekben eltérő névvel illethetik. Vagyis a módszertanok sajátos terminológiával rendelkeznek. A módszertan meghatározónak világos **terminológiát** kell adnia. Az alkalmazónak viszont ismernie kell a fogalmak pontos tartalmát. Például: tudnia kell, hogy a szegmens mennyiben nem rekord; a rekord és a mondat szinonimák; a rekord és a tuple mikor tekinthető azonosnak stb. A Chen-féle és a SYDES-féle egyed, például, nem teljesen azonos lényegét takarnak.

A gyorsan változó technikai környezet miatt a tervezőnek különböző részmódszereket kell egyidejűleg alkalmaznia és nem ritka, hogy eltérő kifejezések mögött azonos a tartalom és fordítva. A módszertan-alkotó felé két követelményt kell támasztani: terminológiája legyen egyszerű és feleljen meg a széleskörűen alkalmazott konvencióknak.

Ezen túlmenően a jó módszer nem lehet zárt. Egyrészt be kell tudnia fogadni 'idegen' részmódszereket. Másrészt nyitott kapukat kell hagynia a fejlődésre. A SYDES fejlesztésekor erre törekszem. A SYDES nem csupán egy további módszer, hanem módszertani keret (frame).

1.4.3 MEGFOGALMAZÁSI MÓD

Az információs rendszer terve maga is információ és ekképpen megkülönböztethető annak tartalma, ill. közlési formája. A tervezési módszerek igen különböző gyökerekből fakadtak és ennek megfelelően ugyanaz a lényeg másként kerül bennük kifejezésre. A terv közlésének a formáját, illetve ennek alapjait nevezem **megfogalmazási módnak**. Néhány 'irányzatra' hívom fel a figyelmet a továbbiakban.

A hagyományos rendszerterveket a faktografikus, vagyis adatszerű megfogalmazás uralta. A tervező a rendszertervet tervezési információ-típusok (metaadatok) értékeiben fejezte ki. A tervezési adatokat sokszor megadott formákon, tervezőbizonylatokon kellett kitölteni. A terv számos részlete ábrákban testesült meg.

Egyetlen terv sem nélkülözhetette a természetes nyelvi leírást. Egyes irányzatok erre a tényre alapozva a tervet kimondottan szemantikai megfogalmazású lényegnek tartják. Előtérbe kerültek a tervező-leíró kvázi-természetes nyelvek is. A pontosság igénye miatt és a matematikai elemzés szükségessége következtében sokan a matematikai(-logikai) nyelvi megfogalmazást tekintik üdvözítő megoldásnak.

A megfogalmazási mód kiválasztásánál hiba volna eltekinteni a terv felhasználóinak háttérétől és a jelen információkezelési gyakorlattól. Azt hiszem, hogy az alábbi gondolatsor a realitásokon alapul:

- Kíváncsú volna, hogy a rögzített metarendszer vezérelje az információs rendszer működését (aktív szótárrendszer).
- A tervezési (meta) és felhasználói (alap) információk között nincs éles határ. Kedvező lenne, ha a felhasználói és a metarendszert egyazon kezelőrendszerrel kezelhetnénk.
- A kétféle rendszerben tehát azonos a megfogalmazási mód.
- Következésképpen ez csak a felhasználó által is érthető mód lehet.

A felhasználó aktív tervezési közreműködése nélkülözhetetlen. A terv a kommunikáció eszköze. Mindaddig, ameddig a felhasználók képzettsége nem emelkedik (alapvető matematikában), a matematikai megfogalmazást egyéb erényei ellenére sem tartom célszerűnek. Mindaddig, ameddig nem áll rendelkezésre okos természetes-nyelvi

fordító, a tervezőnyelv alkalmazását (ami mindig angol lesz) nem tartom elfogadhatónak. (Egyébként redundanciája miatt sem.)

A SYDES-ban a fentiek miatt alapvetőnek a faktografikus (metaadat) megfogalmazási módot választottam. A tervező és a felhasználó forma-orientáltan kommunikál az eszközzel. A fejlesztés jelen szakaszában a grafikus bemenet és kimenet kidolgozását végezzük. A SYDES matematikai alapú elemzéseket végez. Nem tartom kizártnak az ilyen alapú megfogalmazások beépítését sem. Vagyis a SYDES más, mint a módszerek többsége. Azokban a megfogalmazási mód határozza meg a teljes módszer lényegét. Nálam a megfogalmazás csak eszköz, a módszertani kellékek egyike.

1.4.4 KRITÉRIUMRENDSZER

Az információs rendszer terve olyan absztrakt kép, amely a módszer alapkonceptiójára alapozottan, az azáltal feltételezett struktúrában és nyelvezetben, az alkalmazott megfogalmazási mód természetének megfelelően rögzíti elképzelésünket. Mindez azonban csak keret, amelynek tartalmát a mindig aktuális alkalmazási rendszer információs képével kell kitölteni.

Vannak olyan módszerek, amelyek csak arra adnak eligazítást, hogy miképpen kell összeállítani a tervet. Az egyértelműség fokozására, a kommunikáció elősegítésére egyre több módszer alkalmaz formai szabványokat vagy legalábbis ajánlásokat. Az igazán fejlett módszerek eligazítást adnak arra nézve, hogy a terv (adott részlete) mikor érthető, egyértelmű, minimális, teljes és megvalósítható. Azaz egy szóval: mikor optimális. A korszerű tervezés - modellezés. A korszerű módszer nélkülözhetetlen kelléke a célszerű **kritériumrendszer**. Ugyanakkor a tervezőnek tisztában kell lennie azzal, hogy a módszer alkalmazása esetén melyik kritérium jelent az információs rendszer szempontjából valódi optimumot és melyik fontos kritérium fakad a módszer/eszköz korlátaiból. Azaz melyik 'optimumkritérium' - korlát, szuboptimum.

Többen is elismerték a [H29] kapcsán, hogy a SYDES erős oldala a célszerűen megválasztott optimumkritérium-készlet. Vizsgálataim jelentős része arra irányul, hogy feltárjam a rendszerek tényezői közt meghatározható célszerű összefüggéseket. Elemzéseimből ízelítőt fog adni a 4.fejezet.

1.4.5 TERVEZÉSI ALGORITMUS

Nem a rendszerfejlesztés szakaszairól, fázisairól van szó. **Tervezési algoritmus**nak azt a tevékenységsorozatot nevezem, amelynek megfelelően a tervező a rendszer modelljének a részleteit - egyiket a másik után, illetve egymással összhangban - kialakítja. Evidens, hogy ez a lépéssor valahol elkezdődik és valahol lezárul. Nyilvánvaló az is, hogy minden tervezési algoritmus feltételezi valamiképpen az iterációt.

Számos tervezési (rész)algoritmust ismerünk. Ilyen az outputból kiinduló tervezés; a fentről-lefelé való lebontás; az adatmodell-dekompozíció és -szintézis; a szemléletintegráció stb. Evidens, hogy a feladat mérete (mennyiség) és jellege (minőség) szerint más-más tervezési algoritmus alkalmazása lehet célszerű. Nem véletlen tehát, hogy a tervezés minden lépését megkötni óhajtó merev algoritmusok nem számíthatnak tartós, széleskörű sikerre. Az információs rendszer tervezése sohasem lesz pontos algoritmusokban leírható feladat.

Az utóbbi kitétel miatt terjedtek el újabban azok a rugalmasabb algoritmusok, amelyek nem a lépéssort, hanem annak eredményét kötik meg. Egyik válfajuk a prototípus-alkotás, amelyben az algoritmus még részlegesen kötött. A másik a 'laza' (fuzzy) tervezés, amelyben a tervező az ismereteket tetszőleges sorrendben rögzítheti, de a terv végső összeállítása előtt az összefüggések helyességét ellenőriznie kell.

A SYDES ebből a szempontból is kettős-arcú koncepció. Kézikönyvében [H28] alternatív, fix-lépéssorozatú tervezési algoritmusokat adtam meg (ezek 8 változatot jelentenek!). Világos ui., hogy a tervezőnek nem egyetlen algoritmust, hanem **algoritmus-készletet** kell adni, hogy a feladatnak és az ízlésének megfelelőt kiválaszthassa. Ugyanakkor a SYDES, az előző alpontban említett kritériumrendszere következtében és az alkalmazott szoftver sajátosságai miatt, kiválóan alkalmas a 'fuzzy' tervezésre. Jelenleg dolgozom azon, hogy a rendszer prototípus-alkotásra is képes legyen.

1.4.6 DOKUMENTÁCIÓS FORMA

Az információs rendszer terve kettős célt szolgál. Az alkalmazás fázisában a terv, mint dokumentáció (szótár) vezérli ideális esetben a

rendszer működését. Ennek érdekében a terv leírásának technikailag kezelhetőek kell lennie. A fejlesztés fázisában a tervezősoporton belüli, illetve a fejlesztők és a felhasználók közötti kommunikáció eszköze. Ezért pedig leírásának emberileg kezelhetőnek kell lennie.

A dokumentáció meg kell, hogy feleljen az alapkoncepciónak és a struktúrának. Evidens, hogy benne tükröződik a terminológia és a megfogalmazási mód; összeállításánál szerepet játszik az alkalmazott tervezési algoritmus. Ettől függetlenül és annak dacára, hogy a terv és a működő modell csak ideális esetben vág egybe, a dokumentációnak van 'önálló élete' is, önálló használata, kezelése, **formája**.

A dokumentáció-alkotás során az egyik hiba az lehet, hogy a dokumentációt elszakítjuk az eddig tárgyalt módszertényezőktől. Tipikus példa erre a dokumentáció olyan módosítása, amikor a módosult tartalmat nem vetik össze a kritériumokkal. A másik gyakori hiba az, hogy a dokumentáció szerkezete nem felel meg a rendszer célszerű tagolásának. Ilyenkor a tervező gyorsan és kényelmesen alkothat fizikai-, logikai- vagy divatfüggő rendszert. Például akkor, amikor a logikai és fizikai szintű tervinformációk egy lapon összekeverednek.

A legtipikusabb hiba a dokumentáció hierarchikus szervezése. A természetszerűen hálósan összefüggő tényezőket (rekord-adatmező, program-rekord stb.) a kézi dokumentációban (!) mesterségesen hierarchikusan mutatják be. Ez nem hiba. Baj az, ha hiányoznak vagy nem karbantartottak a keresztreferenciák. Baj, ha a számítógépes dokumentáció hierarchikusan szervezett.

A SYDES-koncepcióban a dokumentáció számítógéppel kezelt. Formáját alapvetően ez határozza meg. Belső szervezésében a természetes m:n-es összefüggéseket tükrözi. Külső megjelenítéseiben a kért hierarchikus viszonyokat adja vissza. Tökéletesítése, főleg a láncszerű összefüggések visszakeresése terén, állandóan szem előtt tartott feladat.

1.4.7 TECHNIKAI TÁMOGATÁS

Tegyük fel, hogy egy tervező ragaszkodik a módszertan alap-koncepciójához; ismeri és helyesen alkalmazza annak struktúráit

ésterminológiáját; ragaszkodik a megfogalmazási módokhoz; betartja az optimumkritériumokat; választékosan használja a tervezési algoritmusokat és tökéletesen jól fogalmazza meg a dokumentációt, illetve ismeri ki magát abban! Nem vállalom nagy kockázatot azzal, ha kijelentem: pusztán kézi módszer használata esetén ez képtelenség.

A tervező támogatására és ellenőrzésére számítógépes eszközre van szükség. Ma a tervező-szoftvereket CASE-nek (Computer Aided Systems Engineering) nevezik. A SYDES-ot mint CASE szoftver-eszközt a következő fejezet végén fogom bemutatni.

2. A SYDES-KONCEPCIÓ ALAPVETŐ VONÁSAI

Az előző fejezetben bizonyítottam, hogy a ma széleskörűen használt tervezési módszerek elméleteikben és részleteikben is korlátokat mutatnak. Érveimet expliciten, az orientáltsághoz kapcsoltan és impliciten, a tervezési módszerek kellékeinek a bemutatásával fejtettem ki. Az érveléshez kapcsoltan igyekeztem megvilágítani saját célkitűzéseimet, azaz a SYDES-konceptió kifejtésének vezérfonalát.

Konceptiómnak nem véletlenül adtam a SYDES nevet. Az angol betűszó a **SY**stems **DE**signer **S**ystem kifejezés rövidítése. Ez magyarul jelenthet rendszertervezési és -tervező rendszert is. Elképzelésemnek három oldala van. Rendszer(terv) koncepció, ha úgy tetszik információs rendszer **filozófia**. Ez adott háttérrel a SYDES sajátos tervezési **módszer**ének. A módszerre alapul és azt támogatja a SYDES tervezési **szoftver**. Mondani sem kell, hogy a három oldal között nem egyirányú a meghatározottság, hanem kölcsönös.

Céljaimat a név (SYDES = szájdsz = oldalak) híven tükrözi. A ma általánosan használt módszerek között vannak jelek, széleskörűen elterjedtek, alkalmazásukban kifizetődőek. Azonban szinte kivétel nélkül elmondható róluk, hogy **egyoldalúak**. Egy eszköz oldaláról szemlélik a feladatot (1.3.1). Egy felhasználó elképzeléseit a többiek fölé emelik (1.3.2). A mindig divatos áramlat sodrába kerülnek (1.3.3).

Ettől függetlenül merev az alapkoncepciójuk (1.4.1); nem ismernek változatokat a struktúrában és ezért szűk a nyelvük (1.4.2); szomorú az összehangolatlanság a megfogalmazási módban (1.4.3); kétségbe ejtően kevés a jó terv optimum-kritériuma (1.4.4); orientáltak a tervezési algoritmusok (1.4.5); valóban féloldalasak a dokumentációk (1.4.6) és még várni kell sokoldalú tervezési szoftverre (1.4.7).

A fentiekben nem vígasztalan tervezési képet akartam festeni, hanem tudatosítani kívántam a módszerek fejlesztése terén adódó feladatokat. Egyben egy olyan 'etalont' is fel akartam vázolni, amihez az ebben a fejezetben elmondottak mérhetőek, ill. megmérettethetnek.

Az információs rendszerek tervezési módszertanának a kutatásában a SYDES csak picinyke állomás. Erénye nem teljességében vagy a részletek helyességében van. Hanem szemléletében, amely elismeri a különbözőségben rejlő előnyöket.

Ebben a fejezetben implicit módon megpróbálom az előző gondolatmenetéhez kapcsolódni, elsősorban a módszertani kellékek vonatkozásában. A hangsúlyeltolódások miatt végül a fejezet az alábbi témákat fejt ki:

- Először az alkalmazási modell tényezőiről lesz szó.
- Azután a tervezési rendszer összefüggéseit fejtem ki.
- Rámutatok a két rendszer/modell hasonlóságaira.
- A SYDES keretmódszer (frame) ismertetése következik.
- Legvégül a SYDES szoftver fővonásait világítom meg.

2.1 AZ ALKALMAZÁSI MODELL TÉNYEZŐI ÉS FELÉPÍTÉSE

A **modell** szót igen sokféle értelemben használják [I10]. A festő modell alapján alkot szépet; a műhely modell alapján gyárt sokat. Modell a vegyi, a fizikai, a matematikai képlet. Ilyen a kísérleti példány is. A ma annyira divatos prototípus tehát nem más, mint modell. De - és ez mondanivalóm lényege - az információs rendszer terve sem más, mint modell.

Minden modellfogalomban közös, hogy az alkotó modelljében a valóság valamilyen tipikus jegyeit fogalmazza meg. (Pontosabban: az általa tipikusnak tartott vonásokat.) A rendszertervező az általa megvalósítandó információs rendszerrel kapcsolatosan is ezeket a jegyeket keresi és határozza meg.

Ebben a pontban az ún. **alkalmazási modell** tényezőit és struktúráját határozom meg. Az 'alkalmazási' jelző itt azt tükrözi, hogy az adott szinten a végső-felhasználónak szóló terv fővonásairól beszélek. Ezzel a jelzővel szeretném kikerülni a 'valós-rendszer' régi megnevezéssel összefüggő csapdákat.

2.1.1 TÍPUS, ELŐFORDULÁS, HALMAZ

A tervező az alkalmazási rendszer jelenségeit azok (viszonylag) lényeges, közös és tartós jegyei alapján osztályokba sorolja. A

továbbiakban ezeket nevezem **típus**oknak. A besorolástól fogva az adott osztályba tartozó elemeket **előfordulás**oknak nevezem. Minden egyed típus az előfordulások specifikus **halmaz**ával rendelkezik adott környezetben és adott időszakban.

P21 Kovács Eszter és Szabó Zoltán vállalatunknál tevékenykedik. Így megfogalmazható a 'dolgozó' és a 'név' típus. Ezeknek két előfordulása Kovács és Szabó. 19..-án vállalatunkban 1241 dolgozó volt (előfordulás-halmaz).

Lényeges jegyekről van szó. A vállalatban sem közömbös, például, hogy a dolgozó beteg. A betegség fajtája viszont csak az egészségügyi információs rendszerben fontos.

Általános vonásokat írunk le. Vállalatunkban egyetlen mohamedán dolgozó van, akinél nem törvénytelen a többnejűség. A tervben azonban dolgozónként csak egy feleséget feltételezünk.

Tartós információkat rögzítünk. Nincs értelme modellezni az életkort. Helyette a nem-változó születési dátumot alkalmazzuk.

Típusról és előfordulásról nemcsak az adatvetület elemei esetében lehet és kell beszélni. Az eseményeket, sőt a tevékenységeket is ilyen kettős szemléletben kell értelmezni. Belépnek dolgozók. Ez - típus. Belépett Kovács Eszter. Ez - előfordulás. Aligha zajlik le minden belépés azonos módon. Így a 'dolgozó-felvétel' program is olyan típus, amelynek különböző ágait járjuk be konkrét felvételek alkalmával. Azaz a program futtatása az előfordulás. A rendszerterv szempontjából nem közömbös, hogy adott helyen és időszakban hányan léphetnek be és mennyi a 'dolgozó-felvétel' futtatások száma.

2.1.2 ELEM, VISZONY ÉS STRUKTÚRA

Első menetben a tervező a jelenségeket másoktól függetlenül vizsgált osztályokba sorolja, azaz **elemtípus**okat alkot. Majd - mivel rendszert tervez - elemzés alá vonja a jelenségek közös, tartós és lényeges viszonyait is. Ezeket az összefüggéseket is osztályokba absztrahálja és **viszonytípus**okat alkot.

P22 Kovács Eszter a bérosztály dolgozója. Szabó Zoltán lakatos az A. műhelyben. A dolgozók és a szervezeti egységek közötti viszony általános, tipikus.

A viszonytípusoknak is vannak előfordulásai (Kovács itt, Szabó azon a részlegen dolgozik) és itt is megadandó az előforduláshalmaz (minden dolgozó munkálkodik valahol - és fordítva).

A SYDES-koncepcióban nem kizárt, hogy egy viszonytípus elemtípusként viselkedjék, azaz maga is további viszonyokat létesítsen. Mivel a túlzott - és szoftverrel is alig nyomomonkövethető - komplexitást el akartam kerülni, a további kapcsolati lehetőségeket (kapcsolat-kapcsolatának-kapcsolata) kizártam.

A tervező tehát megvizsgálja az alkalmazási rendszer jelenségeit. Ezeket és ezek viszonyait tipizálja. Elemtípusokat és azok között viszonytípusokat határoz meg. Ezzel megteremti az alkalmazási rendszer **modellstruktúráját**. Ez a modell véges számú elem- és viszonytípus szervezett együttese.

2.1.3 ALKALMAZÁSI MODELL - LEÍRÁS ÉS TERMINOLÓGIA

Egyes koncepciókban a modellstruktúrát - helytelenül - magával az alkalmazási modellel azonosítják. Az sem ritka - erre a 4.fejezetben mutatok rá -, hogy a modell optimumát csak a szerkezet oldaláról vizsgálják. Előfordulhat így, hogy a struktúrálisan optimális modell teljes megfogalmazásában hallatlan redundanciákat tartalmaz.

Mindennek az az oka, hogy az információs rendszer egyes fontos tényezői nem fogalmazhatók meg struktúrálisan, illetve tipizáló, metaadatszerű meghatározásuk nem célszerű.

P23 Nem lehet munkatársunk 14 éven aluli személy. Ha valaki egyszer nő volt, többé sohasem lehet agglegény családi állapotú. Az X program csak Y feltétel esetén indítható.

A SYDES-koncepcióban a korlátokat, feltételeket, kitételeket, meghatározásokat, észrevételeket, szabályokat, szabványokat, megjegyzéseket, megkötéseket stb., tehát mindazt, ami az információs rendszer szempontjából igen fontos, de struktúrálisan (metaadatként)

nem fogalmazható meg, **leírások**nak fogom nevezni. A matematikai alapú kifejezés is leírás.

Az **alkalmazási rendszer modellje** ezért véges számú elem- és viszonytípusból áll úgy, hogy azok mindegyikére véges számú leírás vonatkozhat. A típusok megalkotásakor a tervező szükségszerűen átveszi az alkalmazó hivatkozási rendszerét. Azaz jártasnak kell lennie az alkalmazó sajátos terminológiában.

2.2 A TERVEZÉSI MODELL TÉNYEZŐI ÉS FELÉPÍTÉSE

Egyes elméleti és gyakorlati szakemberek ellenszenvet mutatnak a 'meta...' jelölésű fogalmak iránt. Szubjektív szempontból ez érthető, mivel a metafogalmakkal néha visszaéltek. De objektíven a 'meta'-gondolatot aligha kerülhetik el, hiszen a tervezők óhatatlanul metafogalmakban, metarendszerben dolgoznak és gondolkodnak. Ezt fogom kifejezni az alábbiakban.

A 'meta' szó eredetileg cserét, változást jelent. Esetünkben azt, hogy vizsgálatunk szintjét eggyel magasabbra helyezzük. Most nem az alkalmazási, hanem a tervezésinek nevezett rendszer síkjában fogunk vizsgálgatni. Az információs rendszer tényezőire vonatkozó információkat metainformációknak, azok rendszerét metainformációs rendszernek fogjuk nevezni.

A rendszerterv ismereteket rögzít az alkalmazási rendszer tényezőiről. Vagyis minden tervező metainformációkat határoz meg. Kíváncsú lenne, ha ezt tudatosan tenné. Ehhez fogok támpontot nyújtani az alábbiakban.

2.2.1 KATEGÓRIA, METASTRUKTÚRA, METALEÍRÁS

A tervező - ösztönösen vagy tudatosabban - a tervezési rendszer jelenségeit is osztályokba sorolja azok (viszonylag) lényeges, közös és tartós jegyei alapján. Így használja a típusok típusait, a metatípusokat, amiket **kategóriáknak** hívok. A kategóriákba tartozó elemeket **metaelőfordulásoknak** nevezem. Ezek nem mások, mint az alkalmazási rendszer típusai. Minden kategória az előfordulások

meghatározott **halmazával** rendelkezik adott tervezési környezetben specifikus időpontban.

P24 A tervező megadja a 'dolgozó' és a 'szervezeti egység' típusokat. Ezeket az egyed kategóriába sorolja, szemben a 'névvel' és a 'szervezetkóddal', amiket a tulajdonság előfordulásainak tekint. A környezetben ismert a kategória (pl. egyed) előfordulásainak a száma, azaz a metatípus (kategória) halmazának a számossága.

Metaszinten is megkülönböztetünk elem- és viszonytípusokat. Elem-kategória, például, a program és a rekord, míg a program kezeli-a-rekordot összefüggés viszonykategória. Az elem- és viszonykategóriák szervezett együttese alkotja a tervezési-modell struktúráját. Ez a **metastruktúra** még ezen a szinten sem azonos a modellel. A tervezési fogalmak között akadnak olyanok, amelyek nem egyértelműek, ill. amelyek pontos tartalmára a tervező adott pillanatban nem emlékszik.

Például: csak gyakorlott elemző ismeri az ötödik normálforma pontos lényegét. Ha a program végrehajtási módját említjük, nem mindenki kapcsol, hogy az on-line és köteget lehet. Azt, hogy mi a normálforma és a végrehajtási mód, dokumentálni kell. A tervezési tényezőkre is vonatkoznak definíciók, korlátok. Őket metaleírásokban adjuk meg. A metamodell, a tervezési rendszer modellje a véges számú elemből álló metastruktúra és a metaleírások együttese.

2.2.2 JELLEMZŐK ÉS ÉRTÉKEK

Az alkalmazási rendszer szintjén a tervező a jelenségeket típusokba sorolja. Nyilván azon az alapon, hogy a jelenségek hasonló, illetve eltérő jegyekkel rendelkeznek. Kovács és Szabó személyek, sőt, dolgozók. Van nevük, nemük, foglalkozásuk, végzettségük, lakcímük stb. A szervezeti egységek is 'hasonlítanak' egymásra. De nincs nemük, foglalkozásuk, lakcímük. Ezért kerül Kovács a dolgozó, a bérosztály a szervezeti egység típusba. A típusok tulajdonságaikban különböznek egymástól.

Ugyanez vonatkozik a metatípusokra, a kategóriákra is. A módszer-alkalmazónak már evidens, hogy pl. a szegmens és a rekordtípus két eltérő kategória. A módszeralkotónak viszont el kell gondolkodnia

azon, hogy vajon tényleg két különböző metatípusról van-e szó, vagy egy típus altípusairól. Az viszont kétségtelen, hogy a kategóriák is azért különböznek egymástól, mert mások a tulajdonságaik.

P25 A programokat leírja a programnyelv, a végrehajtási mód, a futási gyakoriság stb. A rekordtípusokra mindez nem jellemző. Van viszont elemszámuk, állományméretük, rekordhosszuk, hosszjellegűk (fix/változó) stb. A programra nem mondjuk, hogy fix vagy változó hosszú-e.

A kategóriákhoz tapadó sajátosságokat, a metatulajdonságokat a SYDES-konceptcióban **jellemző**knek hívom. A jellemzők 'előfordulásai' a (meta)**érték**ek. Minden jellemző adott időpontban és környezetben meghatározott **értékkészlet**tel rendelkezik. Például: a végrehajtási mód lehet kötegelte és on-line.

A SYDES-konceptcióban a jellemzőket több szempont szerint is osztályozom. Itt három ismérvet mutatok be:

- **Általános és speciális** jellemzők. Vagyis kategóriafüggetlen és kategóriafüggetlen jellemzők. Minden típusnak lehet neve, van tervezője, tervezési-állapota és -időpontja. Ezzel szemben sem a programnak, sem a rekordnak nincs ábrázolási módja, míg az adattételnek van. A név, a tervező általános; az ábrázolási mód, rekordhossz. speciális tervezési információ típus, azaz jellemző.
- **Kötelező és opcionális** jellemzők. Vagyis olyanok, amelyek nélkül a típus nem tervezhető, illetve amelyek a tervből elmaradhatnak. A program nyelvét, végrehajtási módját mindig ismerni kell. A program mérete hasznos, de nem kötelező ismér. Rekordtípust nem lehet tervezni az előfordulások száma nélkül, mert nem becsülhető a helyigénye. A rekordhossz is kötelező ismér. A mezők száma a legtöbbször opcionális.
- **Természetes és kódolt** jellemzők. Vagyis olyanok, amelyek értéke önként adódik és olyanok, amelyek értékei megegyezések (konvenciók) eredményei. A rekordhosszat bájtban vagy karakterben fejezzük ki. A 132-es rekordhossz nem igényel különösebb magyarázatot. A program végrehajtási mód lehet

on-line és köteget, amit az 'O' és 'B' betűkkel jelölünk. A kódértékek magyarázatot igényelnek (B = batch, köteget).

2.3 MODELL ÉS METAMODELL

Ez a pont tűnik alkalmasnak arra, hogy eloszlassam a 'meta' körüli félreértéseket. A meta-fogalom a módszertan-alkotók körében is óriási vitákat vált ki. (Lásd az [I13] vitáját!)

Az egyik oldalon áll a fejlesztendő rendszer, az alkalmazási. A tervező meghatározza a típusokat (vevő, cikk stb.) azok jellemzői alapján úgy, hogy majd az alkalmazási felhasználó adja meg az előfordulások értékét. Például: rögzíti a Videoton (vevő) címét. A tervező csak akkor alkothat sikeres rendszert, ha ismeri az alkalmazói terminológiát (például: tudja, hogy mit jelent a nyereségadó).

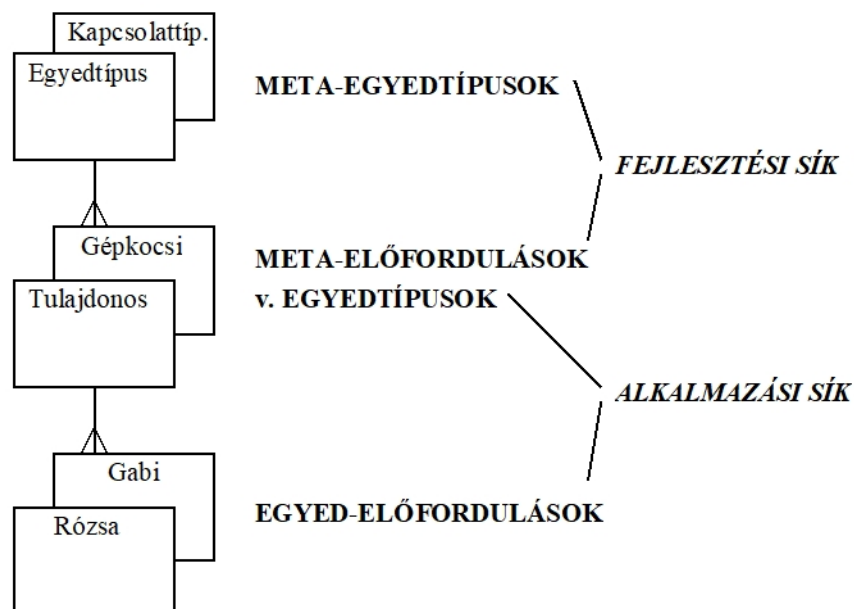
A másik oldalon áll a fejlesztési rendszer, a tervezési. A tervező ebben 'felhasználó'. A (meta)típusok, vagyis kategóriák (rekord, program stb.) (meta)jellemzőinek (rekordhossz, nyelv) a konkrét (meta)értékeit ő jelöli ki a (meta)előfordulásokra, azaz a típusokra vonatkozóan. Például: megadja a vevő rekord hosszát. A tervező csak úgy alkothat sikeres rendszert, ha ismeri a tervezési terminológiát. Tudja, például, hogy mit jelent a normálforma, az adatábrázolás stb.

A 'meta' fogalomtól felesleges félni. Célszerű használni, mivel mindennapi életünk része. Nem szabad viszont túlhajtani. Egyesek szerint a módszertan-készítők 'meta-meta' koncepciókban mozognak. Ez tévedés. Mindig elegendő két síkban, a tervezettben (alap) és a tervezésiben (meta) gondolkodni.

Az alkalmazási és tervezési szint viszonyát a következő oldalon lévő ábra mutatja.

A tervező feladata, hogy meghatározza az alkalmazási rendszer elem- és viszonytípusait úgy, hogy leírja azokat a kategóriájukhoz tapadó jellemzők célszerűen kiválasztott értékével. Például: a vevő rekordtípus hossza 132 karakter.

Tekintettel az alkalmazási és tervezési síkok szoros összefüggésére, valamint arra, amit a dokumentációval kapcsolatban (1.4.6 alpont) kifejtettem, rá kell mutatnom egy súlyos módszerbeli hiányosságra.



2-1. ábra: SYDES modell-szerkezet

T21 A tervezési módszerek jelentős részében nincsenek tekintettel a kategóriák természetes összefüggéseire, ill. azokat féloldalasan definiálják. Ez szükségszerűen szuboptimális tervekhez vezet.

A problémát bemutathatnám a relációs modellen [I16] is. A reláció, mint kategória csak a tulajdonságaitól való függésben létezik, önállóan nem. A tükrözendő jelenség (pl.: vevő) saját tervezési információkkal nem írható le. (Ezen a hibán segít a kiterjesztett relációs modell. Ld. a B.függelék.)

A hagyományos módszerekben [pl. I04] a hiba pont fordított. A tervező nem adhat meg adattételeket mindaddig, ameddig a rekordtípust meg nem határozta. Az adattételnek nincs önálló létjogosultsága: csak rekordhoz kapcsoltnak tervezhető. Ebből a kötöttségből a következő problémák fakadnak:

- . **Redundancia.** A tervben több helyütt is rögzíteni kell az adattétel tervezési sajátosságait (leírását, felépítését, validálását stb.).

- . **Inkonzisztencia.** Az előbbiből fakad. A több helyen leírt adattételre semmi sem garantálja, hogy mindenütt azonosak lesznek a tervezési információi.
- . **Egyértelműség-hiány.** Ha az adattételek hierarchikusan kötődnek a rekordokhoz, akkor semmi sem gátolja a homonimák és szinonimák alkalmazását.
- . **Karbantartás-gond.** Ha az adattételnek változik valamely tervezési ismérve, úgy a többszörös tételt nehéz visszakeresni. A szimultán karbantartás nem garantált.
- . **Rugalmatlanság.** Hiába ismeri a tervező egy adattétel valamennyi sajátosságát, ezt a tervezési ismeretet nem tudja dokumentálni a dokumentáció hierarchikus szerkezete miatt.

A felsorolt problémák gyökere közös: egyes tervezési módszerekben az önálló lényegként megfogalmazható elemi konstrukciókat viszonytényezőknek tekintik. A hálósan összefüggő dolgokat hierarchikusan kapcsolt kategóriákba kényszerítik.

Az effajta módszerek alkalmatlanok az integrált tervezésre. Ha a típusok szintjén megköveteljük a normalizálást (ld. 1.normálforma, 4.5.1 alpont), akkor logikusan ezt kell tennünk a metatípusok esetében is. A SYDES-módszerben törekedtem a kategóriák közötti helyes, természetes összefüggések kialakítására. A SYDES-szoftver ebben nagy segítséget jelent, amennyiben lehetővé teszi a sokoldalú keresztreferencia-kezelést.

2.4 A SYDES KERETMÓDSZER LÉNYEGE

Ebben a pontban a SYDES módszer alapvető vonásait mutatom be. Az eddigiek során többször utaltam módszertani egyoldalúságokra, illetve előrevetítettem a sokoldalú módszerek iránti igényt ('kellékek'). Meggyőződésem, hogy a hibák csak akkor kerülhetők el, a többoldalú követelmények csak akkor teljesíthetők, ha egyetlen módszer helyett módszerkeretet (frame) alkotunk és alkalmazunk.

A SYDES-módszerben az 'egyben több' elvét valósítottam meg. Ugyanakkor a SYDES-et egy további kettősség is jellemzi. Tartalmaz ui. előre megfogalmazott, konkrét konstrukciókat (testreszabottan) éppenúgy, mint szabadon kitölthető vázakat (általánosítottan). Ez a kitétel nemcsak a módszerre, hanem a szoftverre is vonatkozik.

A pontban a tervezési szinteket és a SYDES tervezési 'létrát' mutatom be. Eközben kitérek az ANSI/SPARC konstrukció [I17] elméleti bírálatára.

2.4.1 FIZIKAI, LOGIKAI ÉS FOGALMI SZINT

A SYDES-ban három valós és három 'ál' szintet különböztetek meg. Ennek kapcsán meg kell fogalmaznom az alábbi tételt:

T22 Az általánosan elterjedt tervezési módszerek nem adnak szintek szerint kellően differenciált tervezési modellt.

A hagyományos módszerekben is felismerik a megvalósítási mód és a tervezendő tartalom viszonylagos függetlenségét. Ennek megfelelően **logikai**nak nevezik azokat a tervezési információkat, amelyek az adatok, események, tevékenységek stb. tartalmát és szerkezetét tükrözik. Ezzel szemben a **fizikai** szintű terv az ábrázolásra, az eszközökön való elhelyezéssel/hozzáféréssel kapcsolatos technikai részleteket határozza meg a logikai tartalomra vonatkozóan. A logikai-fizikai szintpáros kezdi kiszorítani a lényeg-részlet (terv-specifikáció) kevésbé egzakt kettősét.

Mintegy másfél évtizede, az ANSI/SPARC [I17] pozitív hatására kezdtük el használni a fogalmi szint kifejezést is. Az adott forrásban a **fogalmi** szint tartalomként áll szemben az ún. belső szinttel, a megvalósítással. Tehát a fogalmi-internális szintpáros a logikai-fizikait helyettesíti. Ezzel kapcsolatosan a következő kijelentést kell tennem:

S221 Az ANSI/SPARC fogalmi, külső és belső szinthármas nem felel meg a tudatos tervezés követelményeinek.

Mivel ezt az állítást másutt [H22] már igazoltam, csak röviden térek ki a lényegére, hogy a SYDES-szintek magyarázatát alátámasszam. A

belső szinttel szemben csak azt vetem fel, hogy a megnevezés nem szerencsés, amennyiben a gép belsejére utal a lényeg, a sajátos ábrázolás és kezelés helyett. (Egy könyvtári kartotékrendszernek nincsen 'belső' sémája?)

A 'külső szint' még rosszabb kifejezés, amennyiben nem szintről, hanem szemléletről van szó. A 'view' megjelölés [I18] használata célszerűbb lenne. Akkor azt sem felednénk el, hogy a szemléletnek is van sajátos ábrázolása és kezelése, belső vagy fizikai szintje, nemcsak tartalma. Az ANSI alapján két külön konkrét externális sémát kell alkalmazni akkor, ha egyazon tartalmat kétféle sorrendben akar látni a felhasználó. A keresztösszefüggések és a minimalitás szempontjából ez nyilván igen kedvezőtlen felfogás.

A SYDES-ban megkülönböztetnek **globális** és **egyéni** szemléleteket. Az utóbbiak az előbbinek kell, hogy megfeleljenek. Az elhatárolás célja a horizontális vagy logikai adatfüggetlenség biztosítása. (A horizontális szó az ábrázolásra utal. Ld. a 2.2 ábrát!) Mindkét szemléleti síkon - nem szinten! - különbséget teszünk tartalmi/fogalmi, ill. ábrázolási/fizikai szint között. Ezzel az a célom, hogy a vertikális vagy fizikai (eszköz) függetlenséget biztosítsam.

A fentiek teremtenek alapot az 1.3 pontban vázolt szemléleti egyoldaloságok kiküszöbölésére. A SYDES módszer elődjeinek a gyakorlati alkalmazása során [vö. H30] arra is fény derült, hogy a tartalom korántsem egyetlen minőség. Van, mondhatni, abszolút és relatív tartalom. Egy példával szemléltetem ezt.

- P26 Dolgozóinknál nyilvántartjuk előző munkahelyüket. Ez ritkán használt információ. Így számítógépes kezelésnél bizonyára csak a legutolsó előző munkahelyet tároljuk. A dolgozó egy adata fogja tükrözni ezt az információt és nem alkalmazunk külön előző munkahely rekordot. Ehhez ui. kezelni kellene a létező előző munkahelyek adatait.

Ez a tervrészlet nem fizikai szintű, hiszen csak tartalomról szól; ábrázolásról, elhelyezésről, eszközről nem. A függetlenség jegyében nem is lenne célszerű 'belső'-ként tekinteni. Azonban nem is fogalmi szintű. A fogalmi kép ui. az, hogy léteznek munkahelyek és dolgozók, akik között fennáll 'az előző munkahely' M:N-es kapcsolat. A tervrészlet a fogalmihoz képest relatív tartalmat ad vissza.

A felhasználói - tehát nem a technikai - korlátok által megkötött tervet nevezem logikai szintűnek. (N.B.: a kezelőrendszerek is szétválasztják a logikai és fizikai tervet.) A korlát lehet biztonsági, integritási, hozzáférési jogi, földrajzi megosztási, hatékonysági stb., vagy akár csak ízlésbeli is.

Itt kell rámutatnom arra, hogy a színhármas az információs rendszernek a feldolgozásvetületére is értelmezendő.

P27 A dolgozó belépésekor végre kell hajtani a felvételi eljárást (fogalmi szint). A tervező összeállít egy programtervet a dolgozó-adatok karbantartására, ideértve a kilépési és egyéb adatkihatásokat is (logikai szint). A program különböző nyelvekben, a hozzáférési-elhelyezési tervtől függő módokon valósulhat meg (fizikai szint).

A SYDES-ban lehetőséget adok a fogalmi-logikai-fizikai színhármas tudatos alkalmazására. Az alábbiakban a 'tudatos' jelző tartalmát fogom kifejteni. Előbb azonban az eddigi összefüggéseket szemléltetem a 2.2 ábrával:

Adat vetület	Feldolgozás vetület
Fogalmi szint	Fogalmi szint
Logikai szint	Logikai szint
Fizikai szint	Fizikai szint

2.2 ábra: SYDES tervezési szintek

Az ábra vetületei egyben a szemléleti síkokat is jelentik. Az adat vetület a globális, a feldolgozás vetület az egyedi szemléletek helyett áll. Utóbbiakat az előbbire minden szinten le kell képezni (horizontális leképezés).

2.4.2 LEKÉPEZÉSI, KÖRNYEZETI ÉS RENDSZER SZINT

A hagyományos tervezési módszerekben a tervező a logikai tervet fizikaivá alakította át, figyelembe véve a technikai lehetőségeket és az ábrázolási igényeket. Eközben belenyúlt magába a logikai tervbe is.

Az átalakítás következtében a logikai terv megsemmisült, feloldódott fizikaiban és az 1.3.1 alpontban tárgyalt anomáliákat okozta.

A SYDES-módszerben a tervező eltérő szinteken eltérő kategóriákat alkalmaz. A kategóriákhoz a szintnek megfelelő jellemzőket köti az azokhoz tapadó értékészletekkel. A SYDES-tervező a fogalmi modellt nem alakítja át logikaivá, hanem arra leképezi. A logikai modellt pedig fizikaira képezi le.

Ez a vertikális leképezés azt jelenti, hogy a magasabb szintű kategória tényezőit alacsonyabb szintűeknek feleltetjük meg úgy, hogy az alacsonyabb szintű kategória jellemzőinek az értékeit definiáljuk.

Példa:

P28 A dolgozó egyedet (fogalmi szint) pénz- és munkaügyi rekord-típusba (logikai szint) képezzük le. Ezeket adott állományokban (fizikai szint) valósítjuk meg. Az egyed-rekord-állomány megfeleltetés tipikus tervezői feladat.

Itt kell kitérni a típustervek problémájára. Ilyenek nálunk nem csak azért nem születnek, mert mások a fizikai eszközök az azonos feladatú szervezetekben. Inkább az jellemző, hogy a logikai szintű felhasználói korlátok és igények mások. Mint arra másutt is rámutattam [H26], a típus-tervekkel nagyobb sikert lehetne elérni, ha azokat fogalmi szinten adnák meg. Majd az alkalmazó képezi le azt logikaira.

Mivel a felvázolt vertikális leképezés nem implikál tervátalakítást, a módszer biztosítja a vertikális adatfüggetlenséget. Van egy további előnye is a hagyományos módszerekkel szemben, ami magyarázatot igényel.

A vertikális leképezés történhet M:N-es módon is. Nem kizárt például, hogy a fogalmi szinten alkotott két egyedtypusból logikai szinten a tervező három rekordtypust határoz meg. Ez azt jelenti, hogy a leképezés olyan szintek közti kategóriaviszony, amelynek jellemzői értelemszerűen az összefüggésre vonatkoznak és nem köthetők a kapcsolt szintek egyikéhez sem. Így a leképezés módja, leírása és esetleges indokai nem tapadnak sem a fogalmi, sem a logikai szinthez a kettő között történő leképezés esetén.

Emiatt vezetem be a SYDES-ban egy mesterséges tervezési szintet, a **leképezésit**. Tudomásom szerint ilyesmi más módszerben nincsen.

A vertikális leképezés vonatkozik az adat- és feldolgozás-vetületre egyaránt. Az adatok és a feldolgozások fogalmi, logikai és fizikai szinteken megfogható **absztrakciók**. Velük szemben a környezeti vetület tényezői (felhasználók és erőforrások) konkrétan létező **valóságok**. Lehet is és kell is őket tipizálni, de nem lehet és nem kell őket megragadni fogalmi, logikai és fizikai szinteken. Ugyanakkor a környezetvetület tényezői mindhárom szinten összefüggenek a másik két vetület alkotóelemeivel (vö. pl. a hozzáférési joggal!).

Az említett összefüggések megfogalmazásának a támogatása és a tervezési tényezők homogén kezelése érdekében vezettem be a SYDES-ban a **környezeti** szint fogalmát. Ebben az esetben tehát a tényezők szintje és vetülete azonos. Itt is hangsúlyoznom kell, hogy a SYDES-szoftverben a vetületek implicitek, a kategóriákban adottak. A szoftverben a szint a fontos ismerv.

Végül nem feledkezhettem meg arról sem, hogy a korszerű módszereket eszközök támogatják. Ezekben szükségszerűen vannak olyan konstrukciók, amelyek nem a tervezendő, hanem a tervezési rendszer részei, illetve azt írják le. Két példát kell említenem: A normálforma fontos tervezési információ, de a működő rendszer tervének már nem része. A tervezési rendszerben le kell írni azt is, hogy mit jelent a végrehajtási mód.

A következetesség és az egynemű kezelés érdekében vezettem be a (tervezési) **rendszer** szintet a SYDES-ban. Ezen a szinten fogalmazok meg minden olyan ismeretet, amely a tervezési rendszerre vonatkozik, nem pedig az alkalmazási rendszer tervére. Esetemben ezt a szintet SYDES-szintnek is hívom.

A szintekre és vetületekre a következő pontban még visszatérek. Az eddigi összefüggéseket szemléltesse a 2.3 ábra!

KÖRNYEZETI	ADAT	FELDOLGOZÁS	LEKÉPEZÉS
	vetület	vetület	
szint	Fogalmi szint	Fogalmi szint	
&	Logikai szint	Logikai szint	Fogalmi-Logikai
vetület	Fizikai szint	Fizikai szint	Logikai-Fizikai
SYDES (rendszer) szint			

2.3 ábra: Tervezési szintek és vetületek a SYDES-ban

2.4.3 TERVEZÉSI ORIENTÁCIÓK

Az 1.4.5 alpontban utaltam arra, hogy minden tervezési módszernek nélkülözhetetlen eleme a **tervezési algoritmus**. Az a lépéssorozat-minta, amelyet követve készíti el a tervező az alkalmazási rendszer tervét. Az algoritmusokkal kapcsolatosan ki kell jelentenem:

T23 A széleskörűen használt tervezési algoritmusok befagyasztásra épülnek. Ezért 'orientáltak', merevek és így alkalmatlanok az integrált rendszerek tervezésére.

Ez a kijelentés kicsit általánosítotttnak tűnhet. Tény azonban, hogy az [I12] tárgyalása alkalmával is, ahol több tíz módszer képviseltette magát, határozottan kiálltak az egyetlen algoritmus és a befagyasztás mellett. A bemutatott módszerek szinte mind 'orientáltak' voltak.

Az orientáció következményeit tárgyalom alább. A következő alpontban mutatok rá arra, hogy létezik más út is.

A legtöbb módszer explicit vagy implicit módon épít a befagyasztásra. Annak érdekében, hogy a tervezés ne essen végtelen ciklusba, a terv valamilyen részletét megkötik és ahhoz igazítják a többi tervrészletet. Például: a hagyományos tervezésben előre megtervezték a kimeneteket (sőt: azok képeit is!) és azokból vezettek le minden tervelemet.

A kisebbik baj az, hogy ez a gondolatmenet végül az elkerülendőhöz, a ciklushoz vezet. Hiszen ha a kimenetnek egyetlen picinyke részlete is változik, akkor elméletileg - gyakorlatilag erre nem kerül sor – újra végig kellene futtatni a teljes tervezési folyamatot. Mivel arra nézve, hogy ilyenkor mi a teendő, kevés módszer ad eligazítást, az alkalmazó tervezők magukra maradnak és inkonzisztens rendszer születik.

A nagyobbik hiba az, hogy a tervben nemcsak adott elemeket kötnek meg, hanem a szemléletet is oly módon, hogy az egyik tervezési tényező mintegy uralja, meghatározza a másikat. Ilyen értelemben beszélünk tervezési orientációról. Nem az a baj, hogy a tervező valamilyen szemléletű. Hanem az, ha szemlélete torzító szemüvegén keresztül nézi a másik oldalt. A tipikus orientációk az alábbi következményekkel járnak:

- . A **feldolgozásorientált** módszerek a kimenetet kötik meg előre. Ha a módszer nem párosul elemzéssel, akkor eredménye logikai függőség (1.3.2 alpont) lesz. Az adatbázis szerkezete igen érzékeny lesz a változásokra, mivel közvetlenül a kimenetek alapján struktúrált.
- . Egyes tervezési módszerek abból indulnak ki, hogy a valóságot tükröző információegységeknek ún. életciklusa van. Jelenségek születnek, módosulnak, majd meghalnak. A valóság hű tükrözése érdekében ezt az életciklust kell pontosan követni. Az ilyen **eseményorientált** módszer sem oldja fel a logikai függőséget. Használhatatlan, ha a rendszer nem változás-, hanem lekérdezés-elemzés-, azaz feldolgozásérzékeny.
- . Az adatbázisok fejlődésével terjedtek el olyan módszerek, amelyek első és meghatározó lépésnek az adatmodell kialakítását tekintik. (Ezt az [I13] tárgyalásán is hangsúlyozták!) A tervező a modellt fagyasztja be a logikai függések felolása után. Az **adatorientált** módszerek jórésze elfeledi, hogy a feldolgozásmódot is elemezni kell. Jellemző például, hogy az [I12] és az [I13] teljesen elhanyagolja a bemenetnek (input) még a létét is!

Ismétlem: nem azt tartom hibának, ha egy tervező vagy egy módszer, mondjuk inkább adat-, mint eljárás szemléletű. Baj, ha a módszer implicálja azt, hogy az egyik tényező lerögzítésével a másik tervezési esélyei eleve beszűkülnek.

A bajok egy része elkerülhető lenne, ha a módszeralkotók először feladatokban gondolkodnának és csak azután lépésekben. Ezen az alapelven nyugszik a SYDES tervezési kerete.

2.4.4 A SYDES TERVEZÉSI KERET

Az 1.1.5 alpont az információs rendszer három vetületét mutatta be. Ezek közül a tervezőnek a környezetit le kell írnia (ritkább esetben át kell alakítania) és meg kell adnia a környezeti tényezőknek a másik két vetület elemeihez való viszonyait. A tervben, mint absztrakcióban a két vetület modelljét kell rögzíteni: az adat- és feldolgozásmódot. Meg kell határozni a kettő közötti összefüggéseket is.

Az adatok olyan részrendszerét, amelyet egy feldolgozás(csoport) kezel, vagyis az adatmodellnek a feldolgozással érintett alhalmazát **adatalmodell** írja le. A tervezőnek ki kell jelölnie a feldolgozások adatalmodelljeit is. (N.B.: Az adatalmodell nem tévesztendő össze a szemlélettel, a 'view'-val, ami egy másfajta alhalmaz. Az almodell nemcsak azokat a tényezőket tartalmazza, amiket a felhasználó látni óhajt, hanem azokat is, amik a szemlélet előállításához szükségesek. Kiválasztási ismérveket, referenciákat stb.)

Megtervezendő tehát két modell és azok kapcsolata, mégpedig a tervezés mindhárom absztrakt szintjén. Ezt mutatja a 2.4 ábra. a bevezetett információs rendszer



2.4 ábra: A SYDES tervezési kerete

Az ábra magyarázata a következő:

- . Az információs rendszer tervének két egymást támogató és egyenrangú oldala van: a viszonylag stabilabb adatmodell, illetve a viszonylag változékonyabb feldolgozásmódel. A két oldalt a 'tervezési létra' két szára jelképezi.
- . Az információs rendszer terve a valóság szilárd talaján áll és az alkalmazási rendszer adott részét fogja át.

- . A tervező a cél felé bármelyik száron megkezdheti az utat. Ha tevékenység vagy esemény szemléletű, úgy a feldolgozás, ha adat szemléletű, akkor az információs száron. Jó keretmódszer esetén mindkét módon ugyanoda kell eljutnia.
- . A létra fokai a tervezési szinteket képviselik. A tervező előbb fogalmi, azután logikai, majd fizikai modellt alkot. Ezek a modellek egyre szűkebbek, két ok miatt. Egyrészt az adott szintnek nem minden eleme kerül leképezésre a következő szinten. Másrészt így fogom jelezni, hogy a konkrét választásokon keresztül a terv egyre jobban beszűkül, több korlátnak van kitéve. A terv a másodlagos valóságot jelentő információs rendszerben csúcsosodik ki.
- . A tervező a létra egyik szárán sem léphet felfelé egyszerre kettőt. Amíg az adott szintű adatmodell el nem készül, addig nem térhetünk át más szintű feldolgozásmódra - és fordítva. (A felfelé haladás a vertikális leképezés.)
- . A létra stabilitását az adja, hogy a két szár minden szinten láncal összekapcsolt. A lánc az adott szintű almodelleket jelképezi. Vagyis a két modell minden szinten egyeztetésre kell, hogy kerüljön. (A lánc az egyedi és globális szemlélet közötti horizontális leképezés.)
- . Igen összetett rendszereknél célszerű előzetes, átfogó tervet is készíteni. A bevezetés tervezése külön feladat. Azt, hogy ez a keret miként kerül felhasználásra, nem vezet-e lazasághoz, a következő pont fogja megvilágítani.

2.5 A SYDES-SZOFTVER ÉS HASZNÁLATA

Ennek a pontnak nem az a célja, hogy programtechnikai szempontból mutassa be a **SYDES-szoftvert**. Korábban kifejtettem, hogy a szoftver a **SYDES-módszerre** épült és azt támogatja. Egyrészt ennek a módját fogom megvilágítani. Másrészt a módszer újabb vonásait fogom bemutatni, hiszen az eszközzel támogatott módszer (1.4.7 alpont) nyilván sajátos. A harmadik célom éppen az, hogy a tervezési segédeszközök kívánatos funkcióit összegezzem.

2.5.1 A TERVEZÉSI SEGÉDESZKÖZÖK FUNKCIÓI

Az utóbbi időben egyre jobban terjed a CASE (Computer Aided Systems Engineering, a számítógéppel támogatott rendszertervezés) gondolata. Nem alkalmazás- vagy programfejlesztési rendszerről, nem új-generációs nyelvről van szó. Azt, hogy mi a CASE, igen nehéz meghatározni, mivel igen eltérő filozófiájú és képességű eszközökről van szó. Csak annyi bennük a közös vonás, hogy teljes információs rendszerek fejlesztésére szolgáló módszereket támogatnak úgy, hogy főtermékük az információs rendszer adott részletezettségű terve.

Az ilyen eszközöknek, szerintem, négy alapvető funkcióval kell(ene) rendelkezniük. Ezek a következők:

- . **Dokumentálás.** Mint azt korábban kifejtettem, a terv-termék és a dokumentáció nem két külön lényeg. A tervezési segédeszközök minimális feladata a tervezési (átmeneti) és a terv (végleges) dokumentáció támogatása. A segédeszköznek tehát szótár-funkcióval kell rendelkeznie (vö. 1.4.6).

A SYDES-szoftvernél törekedtem arra, hogy a szoftver támogassa a szabványokat; biztosítsa a konzisztens terminológia-használatot és lehetővé tegye a jó kereszthivatkozást.

- . **Eligazítás.** A tervező a módszernek megfelelő struktúrákat kell, hogy alkalmazza és az azokat tükröző terminológiát kell, hogy használja (1.4.2). A tervezési-folyamatban célszerű algoritmusokat (1.4.5) kell követnie. Tisztában kell lennie a módszerben bevett megfogalmazási módokkal (1.4.3) és az eszköznek a használatával (1.4.7). A segédeszköznek irányítófunkcióval kell rendelkeznie.

A SYDES-szoftver a szokásos dokumentáción túl fogalommagyarázó segédlettel is ellátott. Ez eligazítást ad a tervezési információkra, algoritmusokra és az eszköz elemeire. Most a SYDES nem tartalmazza a projektirányítás direkt támogatását (implicit felhasználói lehetőséget ad erre). A jövőbeli kiegészítésnek/kutatásnak ez az egyik vonala.

- . **Optimalizálás.** A modell nemcsak terv, hanem elegáns megoldás is. Az információs rendszer tervének az eleganciája optimalitásában rejlik. Nem csak arról van szó, hogy a működő rendszer mennyire optimális. Itt elsősorban magának a tervnek az egyértelműségéről, teljességéről és minimalitásáról beszélek. A mai tervezőeszközök nagy része csak a legelemibb hibáktól óvja meg a tervezőt, Holott a segédeszköznek elemzőfunkciókat kellene biztosítania (1.4.4).

A SYDES kialakításakor komoly kutatásokat végeztem arra nézve, hogy a rendszertényezők összefüggései mikor optimálisak (4. fejezet).

- . **Generálás.** Az információs rendszerterv tényezői között vannak olyanok, amelyek másokból automatikusan adódnak, levezethetők. Lehet itt szó teljes típusokról vagy azok bizonyos jellemzőinek az értékéről. A generálási funkció azért is hasznos, mert az automata segít elkerülni a hibákat.

Arról azonban, hogy ezek milyen mélységűek legyenek, eltérőek a vélemények. Vannak CASE-ek, amelyek automatikusan prototípust készítenek. Ez azonban csak korlátozott feladatok esetében és szigorú fizikai megkötések mellett lehetséges. A SYDES rendelkezik generálási (rész)funkciókkal. Mivel kettős (általános és testreszabott) jellegét meg akarom őrizni, további alapos elemzésre van szükség ahhoz, hogy a generálási funkciókat kötöttségek nélkül kiterjesszem.

2.5.2 A SYDES TERVEZÉSI LOGIKÁJA

Tervezési logikának a tervezési algoritmus lényegét nevezem. A hagyományos tervezési logika szigorúan meghatározott lépéssorrendet feltételez. Szerintem ez a logika kudarcot vallott. Nem bírja el a változásokat. Nehezen tudja kezelni a fáziseltolódásokat, amikor a rendszer tervének az egyik része lényegesen megelőzi a többi részlet megalkotását. Rugalmatlan a tényezők meghatározási sorrendjét illetően. Nem viseli el azt, hogy azok átmenetileg (!) csak részlegesen meghatározottak. És ami igen fontos: nem képes nyomkövetni a terv optimum-kritériumait, ha a tervben változások következnek be. Pedig csak a 'csúszó' optimum lehet az integrált terv alapja.

A SYDES-konceptióban a hagyományostól teljesen eltérő tervezési logikát alkalmazok. Ennek először a lényegét foglalom össze, majd röviden ismertetem főbb lépéseit.

A SYDES-alkalmazó tetszőleges tervezési algoritmust használhat. A SYDES támogatja a fentről-lefelé lebontó és az alulról felfelé építkező feldolgozástervezést, a dekompozíciós és a szintézises adattervezést, valamint ezeknek az algoritmusoknak bármilyen kombinációját. (Erről a 3.fejezetben bővebben is lesz szó.) A tervező, szemléletének megfelelően, bármelyik tényezőtípussal elkezdheti a tervezési munkát.

A SYDES-szótárba kerülnek, kategóriájuknak megfelelően, a megtervezett tényezők, azok összefüggései, tervezési jellemzői és leírásai. Ezek eleget tesznek a legalapvetőbb tervezési szabályoknak (kritériumoknak). Időről-időre a tervező alkalmazza a SYDES megfelelő elemzőprogramjait és kiszűri segítségükkel a szuboptimális elemeket/összefüggéseket. Ha a tényezők adott csoportja nem mutat tervezési hibát, úgy sor kerülhet az adott tervrészlet véglegesítésére.

Az elmondott logika a következő, szoftver-szempontról bemutatott lépésekből áll:

- **Elembetöltés.** A tervező a meghatározott elemtípusokat (egyedeket, feldolgozásokat, rekordokat, programokat, mezőket, kimeneteket stb.) a szótárba viszi. A bevitel tetszőleges sorrendű. Emberi hatékonysági okokból célszerű egyszerre egyfajta tényező kötegeit kezelni. Amint a tervező valamilyen ismeret birtokába jut, azt azonnal első ellenőrzés alá vetheti.

Bevitelkor a SYDES bizonyos belsőösszefüggéseket érvényesít. A sorrendre vonatkozó megállapításomat pontosítja ez a kitétel. Példa: egy családfa egyed addig nem határozható meg, amíg alapegyede nem ismert. A bevitel során ellenőrzi a rendszer a kevés, de szigorú formai szabvány betartását. Az információk keresztellenőrzésére is sor kerül. Pl.: az átlagos előfordulásszám nem lehet nagyobb a maximálisnál.

A tervben lesznek olyan elemek, amelyek egymástól kölcsönösen függenek, ill. amelyek helyessége mások jóságától függ. Ezeket a kritériumokat csak azután lehet elemezni, miután a kapcsolt tényezők mindegyike szótárba került; a tényező egyedi bevétele után még nem.

Példa: nincs feldolgozás kapcsolódó esemény nélkül. Mivel a bevitel elemenként történik, a SYDES ilyenkor (szelektíven lekérdezhető) **jelzőket** állít be, amelyek mutatják az ideiglenes hiányosságot.

- . **Kiegészítés.** A 2.2.2 alpontban a tervezési információk kötelező és opcionális csoportjaira utaltam. A tényezőbetöltésnél az előbbieket feltétlenül meg kell adni. Az opcionális ismeretek a tényező-bevitelnél megadhatók. Az ilyen információk értéke egy változtató programmal bármikor bevihető és módosítható (a tartalmi szabályoknak megfelelően). Az előfordulásszám, például, az egyed opcionális jellemzője. Elembevitelkor vagy változtatással kezelhető.

Hasonlóan kezeli a SYDES a tényezőhöz tartozó szöveges leírásokat azzal az eltéréssel, hogy bevitelkor csak a definíció típusú leírás adható meg. (A SYDES többféle leírást ismer és leírástípusokat az alkalmazó is definiálhat. Korlátokat, feltételeket, algoritmusokat stb.)

- . **Viszonybetöltés.** Amennyiben a kapcsolódó két elemtípus, pl. egy feldolgozás és az azt kiváltó esemény, már a szótárban van, úgy azok viszonytípussal összeköthetők. A viszonytípusok megadása ugyanúgy történik, mint az elemtípusoké, de a kapcsolt tényezőkre utalni kell. A fenti két lépésben elmondottak a viszonyokra is vonatkoznak. Maguk a viszonyok is létesíthetnek összefüggéseket. A SYDES újabb változatában viszonyokat közvetlenül is be lehet tölteni (azaz: egy menetben a kapcsolt elemekkel).

- . **Kezelés.** Általános adatkezelésről van itt szó. Nyilvánvaló, hogy a

helytelenül bevitt tételeket módosítjuk vagy töröljük. Ehhez esetleg összefüggéseket kérdezzük le. Kiegészítő információkat töltünk be stb. Szemben az egyszerű szótárrendszerekkel, a SYDES törléskor és módosításkor ellenőrzi a terv integritását. A műveletet esetleg megtagadja, vagy beállítja a fentebb említett kritérium-jelzőket.

- . **Elemzés-2.** Elemzés-1 lépést nem említettem. Azért, mert az első elemzésekre betöltéskor, módosításkor és törléskor kerül sor. Ha a tervező a kiválasztott terület tervtényezőit a szótárba vitte, akkor van módja a második szintű elemzésre. A SYDES által beállított speciális jelzőket szelektíven le kell kérdezni. Ilyen módon főleg a modell teljességét és egyértelműségét ellenőrizhetjük.
- . **Elemzés-3.** Amennyiben az előző lépés teljességet mutat, akkor van lehetőség erre a lépésre. Futtatjuk a tényezőtípusoknak megfelelő (pl. normalizáló) és az általános (pl. redundancia-elemző) s peciális SYDES kötegelt programokat. Ezek az alkalmazási modell adott részletének az egyértelműségét, teljességét, főleg minimalitását vizsgálják. Nemcsak struktúrákat, hanem egyes korlátokat is elemeznek. Futásuk során hibajelzést adnak, letiltják a folytatást vagy belső jelzőket állítanak be.

Ezen a ponton derül ki, hogy a látszólag laza tervezési algoritmus milyen kemény tervezési logikán alapul. Ha a rendszer a koncepció által súlyosnak ítélt hibát fedez fel, akkor megtagadja a tervrészlet véglegesítését. A tervezőnek iterációval javítania kell (!) a hibát addig, ameddig a rendszer a tervet 'jóvá nem hagyja'.

- . **Generálás.** Egyes tényezőket, pl. egyedek közötti kapcsolatokat, a rendszer automatikusan határoz meg. Ennek az a feltétele, hogy a kiinduló tényezők helyesen megadottak legyenek. Ilyen szempontból 'generáló' programnak tekinthetők a modell-kiíró programok is. Valójában ezek újabb tervezési információt nem adnak, csak az elrendezettséget biztosítják a tervdokumentációnak megfelelő módon. Ezen kívül végső ellenőrzéseket is végrehajtanak a nem-kötelező részletek teljességére vonatkozóan. Tehát modell-kiírás után még mindig módja van a tervezőnek az iterációra.

A SYDES tervezési logika ismertetésének zárásaképpen utalnom kell a **szoftvertámogatás jellegére** is. A szótárrendszerek csak minimális beviteli ellenőrzéssel rendelkeznek. Vannak viszont olyan eszközök (pl. modell-normalizátorok), amelyek betöltéskor kereszt-ellenőrzéseket hajtanak végre (pl. nem engedik a rossz függés megadását).

Mivel itt keresztellenőrzésekről van szó, amelyekben adott esetben nagyszámú elem (az összes?) részt vesz, ez a támogatási megoldás nem hatékony. Elvileg sem helyes, mert a most 'rosszul' megadott tényező a következőn keresztül már jónak minősülhet. Ezért alkalmaz a SYDES külön elemzőprogramokat.

Ezeknek van egy további sajátosságuk. Szemben más eszközökkel (pl. normalizátorokkal), nem törlik automatikusan a matematikailag-logikailag helytelen tényezőt és összefüggést. A gyakorlati tapasztalat ui. azt mutatja, hogy a hibák oka igen sokszor emberi tökéletlenség és nem feltétlenül a matematikai értelemben rossz meghatározás az elvetendő. A SYDES ezért kijelzi és jelöli, de nem törli a hibás elemet.

A SYDES sokrétű elemzést ad. Súlyos hibák (pl. ciklusok) esetében ragaszkodik a korrekcióhoz. Több szintű üzenetrendszerrel szolgál és jelzőket állít be. A jelek vizsgálatakor megismétli a figyelmeztetést. Azonban végeredményben az emberre bízza a végső döntéseket. Adott esetben még a szuboptimális tervet is megengedi.

Ez az utolsó kitétel mutatja a SYDES-filozófia legmélyebb elvi gyökerét. Az információs rendszer tervezését alapvetően emberi, teljes mértékben nem automatizálható feladatnak tartom. A segédeszköz óvjon a hibáktól, figyelmeztessen a következményekre, generálja a triviális részleteket, dokumentáljon, esetleg irányítson. De ne vegye át az alkotó ember tervezési szerepét! Legyen neve szerint **segédeszköz!**

3. A SYDES RENDSZERMODELL FŐVONÁSAI

A 2.fejezetben bemutattam az információs rendszer tervének vetületeit és szintjeit (vö. a 2.3 ábrával). A vetületeknek megfelelően az alkalmazási rendszer modellje egymással bonyolult módon összefüggő három részmodellből áll: az adat-, a feldolgozás- és a környezet-modellből. Mint ugyanott jeleztem, ezek közül a környezetmodell csak érintőlegesen tárgya ennek a munkának.

A másik két vetület tekintetében különbséget teszünk a modellek fogalmi, logikai és fizikai szintje között. A fizikai szint a technikai környezet és az ábrázolási igények/szokások függvénye. A logikai viszont a felhasználói korlátok által meghatározott. A SYDES-szoftverben és -módszerben speciális és általánosított lehetőség van a logikai és fizikai modell összeállítására. Ezeknek a rendkívül sokféle korláttól függő modelleknek a bemutatása meghaladja e munka keretét. Ezért a továbbiakban a fogalmi modellezésre szorítkozom.

A SYDES rendszermodell bemutatása sokoldalú feladat. A fogalmi modellezés megjelölés ma még egyenként is viszonylag új; eltérő elméleti és gyakorlati gyökerekből táplálkozó; pontosan még nem ismert közös cél felé orientálódó; diszciplinává még nem érett elképzelés-együttest takar. Az elmúlt évtized első felében az eltérő adatmodell-koncepciók között még alapszinten volt a vita [I30]. Az évtized végére közeledtek egymáshoz az irányzatok [I31], sőt koordinációs kísérletekre is sor került [I32]. A 80-as évek elején találkoztak egymással az adat-, esemény- és tevékenységorientált koncepciók [I11]. Ma az egyeztetés szakaszában vagyunk [I12], de az [I13] körüli vita is mutatja, hogy még az alapokban sincs egyetértés.

Ezért nem elegendő, ha a SYDES-koncepciót felvázolom, hanem alapos indoklást is kell adnom. Mindezt tételszerűen fogalmazva a következő kijelentést kell tennem:

T3 A jelenlegi modellezési irányzatok alapos vizsgálata azt mutatja, hogy a koncepciókban elméleti és főleg alkalmazási rések mutatkoznak. Ezért volt szükség egy új modellezési koncepció kifejlesztésére. Ennek előnyei a vizsgált elméletekkel szemben egyértelműek.

Állításom igazolására a fejezetben az alábbi témákat fogom tárgyalni:

- . Először bemutatom a SYDES adatmodellt, indoklást adva arra nézve, hogy miért nem követek egy 'bevett' koncepciót.
- . A SYDES feldolgozásmódel hasonló tárgyalása következik úgy, hogy az adatmódellel való kapcsolatot is ismertetem.
- . A harmadik pontban a SYDES-ban alkalmazható fogalmi modellezési algoritmusok változatait ismertetem.

3.1 A SYDES FOGALMI ADATMODELLJE

A SYDES adatmodell-koncepciója egyik közismert irányzatnak sem felel meg. A felszínes szemlélő számára leginkább az egyed-kapcsolat [I33] módelre hasonlít. Ez a rokonság abból adódik, hogy még a nyílt viták idején olyan elképzelést fogalmaztam meg [H02-04], amely a relációs [I16] és a hálós [I34] módel **ötvözetének** tekinthető. Ebben a két irányzat pozitív vonásait próbáltam egyeztetni, saját gondolatokkal kiegészítve, elvetve azt, amit rossznak tartottam.

Az első megfogalmazásokat [H06-H07] követően nemcsak elméleti kutatások, hanem a gyakorlati munka [H31] eredményeképpen is finomodott módel-koncepcióm. Részletei a [H28] szoftverbe ágyazottak. Lényegét a 3.1.3-4 alpontok ismertetik. Előbb azonban a közkezdvelt modellezési irányzatokról fogok néhány szót ejteni, igazolva saját eltérő elképzeléseimnek a jogosságát (3.1.2 alpont). Mindez nem lenne kerek, ha az elméleti áttekintést nem a saját módel-felfogásom alapjainak a bemutatásával és igazolásával kezdeném (3.1.1 alpont).

3.1.1 AZ ADATMODELLEZÉS CÉLJAI ÉS KÖVETELMÉNYEI

A 70-es évek elején 'adatmódelnek' a meghatározott feldolgozási elképzelés szerint kezelt, az adott koncepciónak megfelelő struktúrájú adatbázisokra vonatkozó nézetek együttesét nevezték. Ilyen alapon különböztettek meg **hierarchikus, hálós és relációs** 'adatmódellet'. A megjelölés pontatlan, mert ezek a 'modellek' kezelési aspektusokat is implikálnak.

Ma is sokan összetévesztik a relációs adatmodellt a relációs kezelőrendszer egészét felölelő relációs modellel. Az osztályozás azért sem konzekvens, mert szemléletet kever össze struktúrákkal. Nevezetesen: a relációs modellben is kezelhető hálós (tehát: m:n-es) és hierarchikus (tehát: 1:n-es) **adatstruktúra**.

Az adatbázisok terjedésével egyre fontosabb hangsúlyt kapott az alkalmazás-független adatbázistervezés. Kiderült, hogy az 'adatmodellek' nincsenek ellentétben egymással. A hálós és a hierarchikus adatbázis is tervezhető relációsan [I35], vagyis az adattételek függésein alapuló normalizálással. Az adatmodell szó egyre inkább az **adatbázis-terv** jelentést vette fel. Kiderült, hogy ekképpen nem csak egy lényegre támaszkodik.

A fizikai függetlenség (1.3.1 alpont) miatt **szinteket** kell definiálni. A logikai függetlenség (1.3.2 alpont) viszont **szemléletek** elkülönítését követelte meg. A két igényt már a CODASYL [I34] is előrevetítette a séma-tárolási séma és a séma-alséma párossal. A relációs modell szemérmesen elhallgatta azt, hogy az adatokat tárolni is kell.

Kompromisszumként született meg az ANSI/SPARC [I17] fogalmi, külső és belső modellhármasa. Ennek kritikáját adta a 2.4.1 alpont. Az adatmodellt legalább két szemléletben (globális és egyedi) és mindkét szemléletet legalább két szinten (tartalmi és formai) kell meghatározni. Olyan adatmodellezési koncepciót, ami e kettős követelménynek eleget tenne, nem ismerem.

A szemléletek és a szintek száma fokozható a nagyobb függetlenség elérése érdekében. A [H08]-ban, például, a formai szintet hozzáférésire és ábrázolásra bontottam. A szemléleteket is lehetne általános-speciális-egyedi osztályokba sorolni, vagyis közbenső szemléletként határozni meg a csoportos felhasználókat. Azonban módszertanilag a túlsok szint/szemlélet nem kívánatos. Így maradt a SYDES-koncepció lényege a két szemlélet/három szint (vö. a 2.4 ábrával!).

Mindezek után az adatmodell fogalomnak adódik egy harmadik, szűkebb értelmezése is. Adatmodellnek az adatbázis **fogalmi szintű** és **globális szemléletű** tervét nevezik. Adatmodellezésnek pedig az ilyen terv kialakítására szolgáló tervezési tevékenységsort hívják. De vajon mi az adatmodell és az adatmodellezés **célja**? Ezen a téren különbözik lényegesen a SYDES-koncepció a többitől.

Az adatmodell **technikai célja** az ANSI/SPARC rendszer-struktúra megvalósulása esetén az, hogy kapcsolatot teremtsen a külső sémák és a fogalmi, a fogalmi séma és a belső sémák között. Ez a feladat azóta létezik, amióta van adatfeldolgozás. A CODASYL koncepcióban nyilvánult volna meg a legpregnánsabban, ha ott a séma valóban fogalmi lett volna.

Az adatmodell **tervezési célja** világos. Az adatmodell megalkotása jelenti az adatbázis-tervezés első lépését. Hiszen a tervezőnek a feladatról legelőször 'fogalmat kell alkotnia'. Az adatmodell a logikai (majd a fizikai) adattervezés alapja. Viszont a feldolgozás-tervezésben is az első szintet jelenti, amennyiben alapvető feltétel, hogy az egyedi, külső szemléletek megfeleljenek az egyetlen globális fogalminak.

Az adatmodellnek sokoldalúan kell tükröznie a valóságot. Az adatmodellezés kimondott célja a **szemantikai gazdagság** fokozása, ami az ún. 100%-os elvben testesül meg [I36]. Ez kimondja, hogy az adatprogram függetlenség érdekében minden információkkal kapcsolatos ismeretet a programtól elkülönítetten, az adatmodellben kell rögzíteni.

A SYDES-koncepció innentől tér el az általam ismert összes egyéb felfogástól. A T3 kiegészítésére, alátámasztására kell kijelentenem:

T31 A modellezési módszertanok jelentős része nem ismeri fel az adatmodell általános szervező hatását, kommunikációs célját és integráló erejét.

Elsősorban a relációs tervezők képzelik azt, hogy a normalizálás eredménye egyben az adatbázis-séma is lesz. Szinte minden munka ezt sugallja, ideértve az [I24]-et is. Ez tévedés! A fogalmi adatmodell az adatbázis-tervénél tágabb is, szűkebb is. **Tágabb**, mert olyan elemeket is felölel, amiket sem most, sem később nem kezelünk számítógépen. (Vö. az előző munkahely példával, 2.4.1 alpont!)

Szűkebb, mert nem tartalmazza a technikai kezelési információkat. A tágság adja az adatmodell szervező hatását. Az adatmodellező olyan tényezőkként is kapcsolatba hozza a tervét, amelyek információit nem számítógépen fogják kezelni. Így az információs rendszer gépi és kézi elemei együtt szervezhetők.

Az általam ismert modellezési koncepciók mindegyike arra a nézetre épült, hogy a fogalmi modellt **supertervezők** alkotják meg, akiket az egyszerűek és a felhasználók csak kiszolgálnak a megfelelő

információkkal. Az adatmodellnek a megfogalmazását sem értheti a mindennapi halandó.

Ez a nézet nevetséges és hamis. Nevetséges, mert mindent átlátni képes szupertervezők nem léteznek. Hamis, mert az adatmodellezés legfontosabb célja a **kommunikáció**, az, hogy a fejlesztés résztvevői tökéletesen megértsék egymást. Erre utaltam már a [H26]-ban is.

Lehetetlen nem látni, hogy az x és az y felhasználó érdekei adott esetben ütközhetnek. Ezért az őket kiszolgáló fejlesztők között is konfliktus fog támadni. Csak olyankor, ha egymás céljait nem értik meg és nem hajlamosak a kompromisszumra. Az adattárolási és kezelési ellentmondások feloldásának, az érdekegyeztetésnek, és ezzel a valóban **integrált** rendszer kialakításának a legcélszerűbb eszköze a fogalmi adatmodell. Ezt a következő érvek is alátámasztják:

- . Minden felhasználó szeretne a saját információs körénél **tágabb** képet kapni. Így láthatja a felé esetleg megjelenő követelményeket, ill. saját igényeinek bővítési lehetőségeit.

- . A 'szemlélet' koncepció hiányos. Egy alkalmazónak **többféle szemlélete** van. A hagyományos tervezésben látta ezek egységét, volt egyféle összképe a saját információiról. Ilyet a szupertervezőt feltételező módszerek nem kínálnak. Tehetnék, ha az adatmodell megfogalmazása érthető lenne.

- . A gyakorlati tapasztalatok szerint a végső felhasználó képes a fogalmi adatmodell megértésére, ha azt a megfelelő fogalmazásban találják neki. Ekkor **alkotó módon** járul hozzá a fejlesztéshez, saját feladatnak tekintve a jobbítást.

- . Amint azt a [H26]-ban kifejtettem, az adatmodell az oktatás, a begyakorlás eszköze is lehet, amennyiben az adatmodell a környezetre vonatkozó **tudásbázisnak** is tekinthető. A módszeralkotóknak át kell formálniuk gondolkodásmódjukat. Be kell látniuk, hogy az átlagos felhasználó a tisztán matematikai, logisztikai, szemantikai megfogalmazásokat nem érti. Bele kell törődniük, hogy nincsenek zseni tervezők, akik rövid idő alatt teljes rendszereket képesek átlátni. Egyetlen megoldás ezért csakis a közérthető modell-megfogalmazás.

3.1.2 OPTIMUMKRITÉRIUMOK ÉS MODELL-IRÁNYZATOK

A legtöbb modellezési szakember először hitet tesz a környezetének, szokásainak, lehetőségeinek megfelelő modell-konceptió mellett. Azután vizsgálja, hogy vajon a koncepció mennyiben felel meg egyes optimum-kritériumok teljesíthetőségének. Aligha vitás, hogy magának a koncepciónak a megválasztásával máris korlátoztuk az optimum-keresés lehetőségeit. Ennek bizonyítására - igen röviden – áttekintem a legelterjedtebb modellezési irányzatokat a fő optimumkritériumok szerint. Először megismétlem az 1.4.4 alpontban adott kritériumokat:

- érthetőség;
- egyértelműség;
- teljesség;
- minimalitás;
- megvalósíthatóság (itt nem vizsgálom).

A SYDES adatmodell több modellezési irányzat egyenként is alapos elemzésével, a különböző koncepciók hasznos konstrukcióinak az összegyűrásával, illetve átfogalmazásával, a negatív elemek elvetésével és az alapos átfontolást követően az egyéni kiegészítések bevezetésével készült.

- . **Hálós modell** [I34]. A CODASYL-javaslatról egyszerűen lefejtethők a fizikai függőséget okozó elemek. A rekord, az adattétel és a 'set' logikai fogalom könnyen fogalmi szintre vetíthető egyed, tulajdonság és kapcsolat formában. A modell szemantikailag viszonylag gazdag (sajátos kapcsolattípusok, opcionális stb.). Azáltal, hogy az elsődleges és az idegen kulcs nem kötelező, a modell sokat veszít az egyértelműségből, az integritásból. A hálósan megfogalmazott modell érthető, de a másik három kritériumra nincsenek megköttései.

A SYDES innen örökölte a fizikai jellegétől megfosztott kapcsolat-fogalmat (annak specifikumaival együtt) és azt, hogy az m:n-es összefüggést egyedként modellezem.

- . **Bináris modell** [I37]. Két ok miatt eleve tarthatatlan. Egyrészt a jelenségek nemcsak párosával függnek össze. Ha a koncepcióhoz ragaszkodnánk, akkor sértenénk a minimalitás elvét azáltal, hogy számos mesterséges konstrukciót vezetünk be. (Példa: adott projektekre bizonyos cikkeket szállítanak meghatározott partnerek. Miképpen fejezhető ki ez a tény binárisan?) Másrészt: a felhasználónak - nekem sem! - érthető, hogy például nincs személy koncepció, csak személyi-szám/nem, személyi-szám/foglalkozás reláció van. A koncepcióban a modellezendő 'objektum' veszik el.

A modell ellentmond a minimalitás elvének. Ha a személyeknek négy jellemzője van (összesen öt fogalom), akkor ezt a bináris modell négy relációval (nyolc fogalompár) tükrözi. A 'binaritás' gondján csak névlegesen segítenek az ún. lebonthatatlan (irreducible) relációk [I38].

A SYDES innen örökölte a leírt (egyed) és a leíró (tulajdonság) relativitását. Azt, hogy a 'szín' mindkét módon felfogható. A bináris koncepció sejteti a tényezők egyenrangúságát is. Innen adódik, hogy lehet modellezni tulajdonság nélküli egyedet - és fordítva!

- . **Relációs modell** [I16]. Az n-fokú relációk modellje tulajdonképpen felöleli a bináris és lebonthatatlan relációkét. Gyakorlatilag azonban azoktól lényegesen különbözik. Elsősorban azáltal, hogy a felhasználó számára is jól érthető és nem implikál felesleges redundanciákat. De pusztán adatmodell-struktúra szempontjából a fizikai terhektől megtisztított hálós-modellhez képest szegényes. Elsősorban a kapcsolat fogalmát nélkülözi.

A szemantikai gazdagságot fokozza a kiterjesztett relációs modell (amit a B.függelékben elemzek). Azonban a kapcsolat itt is csak korlát marad és nem lesz struktúrális tényező. Ugyanakkor aligha kétséges, hogy az összes modellezési koncepció közül egyedül a relációs tett komoly lépéseket a matematikai értelemben vett minimalitás felé, ami implikálja egyben a szemantikai egyértelműséget is. A teljes 4.fejezetet ennek a témakörnek szentelem, éppen egyedisége miatt.

A SYDES-koncepcióba átvettem a minimalizálás (normalizálás) relációs koncepcióját. Azt a 4.fejezetben kifejtett indokok miatt az ott megadott módokon bővítettem ki. A valóságghűség kedvéért kell

közölnöm, hogy a kiterjesztett relációs modell (RM/T) munkámra már nem volt hatással. Értékelését mégis fontosnak tartom (B.függelék).

- . **Chen-modell** [I33]. Az egyed-kapcsolat modell feltételezhetően (?) a CODASYL javaslat átgondolásával, fogalmi szintre való emelésével született. Eerre utal a kapcsolat kategória használata. Mivel magam is ezt az értékelést végeztem el, a Chen-féle alapmodellből nem merítettem gondolatot. Erről könnyen meg lehet győződni, ha valaki a két modellkoncepció definícióit gondosan tanulmányozza.

A Chen-féle modellben egyednek nevezik az információval leírandó jelenséget és kapcsolatnak az egyedek közötti összefüggéseket. A definíciós kör tautologikusan bezárul akkor, amikor kijelentik, hogy a kapcsolatok információval leírandó jelenségek... A SYDES-ban nincs ilyen súlyos definíciózavar: az egyed és kapcsolat fogalma egyértelmű.

A Chen-modellből mégis vehető hasznos gondolat. Az összetettebb információs rendszerek esetében a modellezésnél nem célszerű rögtön belevágni a teljes egyed-tulajdonság-kapcsolat modell kifejtésébe. Célszerű készíteni előbb egy átfogó egyed-kapcsolat modell-keretet, amely csak a lényeges egyedeket és összefüggéseiket öleli fel, tulajdonságok nélkül. (Vö. a BSP adatosztály-elemzésével [I06]!)

Amellett, hogy mindegyik modellkoncepció hasznos konstrukciókat tartalmaz, három negatív vonásuk nem kérdéses:

- . Inkább leíró jellegűek - kevés kritériumot tartalmaznak.
- . Főleg alapszerkezeteket adnak - szemantikailag szegények.
- . Korlátos szemléletűek - nem kötődnek feldolgozásmóddalhoz.

3.1.3 A SYDES ADATMODELL ALAPKATEGÓRIÁI

A SYDES alkalmazási adatmodell véges számú egyed-, tulajdonság és kapcsolattípus szervezett egysége. A SYDES adatmodellje tehát EAR típusú [I31]. A továbbiakban ezt a három kategóriát, azok összefüggéseit, ill. sajátosságait mutatom be. Az ismertetés célja az általános fogalmak SYDES-sajátosságainak a felvázolása.

Egyed. (Mivel alább mindig típusokról lesz szó, a kifejezések mellől elhagyom a 'típus' szót.) A SYDES-ban az információkkal leírni kívánt

jelenségek (objektumok) absztrahált, tipizált tükörképét nevezem egyedtípusnak. A konkrét jelenség reprezentánsa az egyedelőfordulás. Az egyedtípus az előfordulások adott halmazával rendelkezik.

Szemantikailag különbséget teszünk alap, egyedi, specializált és családfa **típusú** egyed között. Az adatmodell lényegét az alapegyedek (cikk, vevő, rendelés stb.) adják. Az egyedi egyedek kumulált információk tervezésére szolgálnak. (Az összes-vevő egyed rögzíti vevőink összes tartozását.) A specializált egyed alosztályt tükröz (export rendelés). Végül a családfa egyed egy adott egyedtípus előfordulásai közötti belső m:n-es összefüggést írja le. Nem vitás, hogy az alap és egyéb egyedek között számos korlát határozható meg. Ezek többségét a SYDES érvényesíti. (Pl.: minden alapegyedhez egy egyedi tartozhat és fordítva. A specializált egyed hivatkozása az alapegyed szerepneve. A családfáé az alapegyedének a hivatkozásából összetett.

Más szempontból megkülönböztetünk alap, egyedi, leíró, kapcsoló, referencia, osztályozó stb. egyedeket. Az első kettő fentebb magyarázott. A leíró egyed hierarchikusan kapcsolódik az alaphoz és annak tárgyról mond bővebb információt (pl.: dolgozó előző munkahelye). A kapcsoló két alapot köt össze (pl.: a rendeléstétel a rendeltetést és a cikket). A referencia egyed nem más, mint egy kód-név páros. A kényelmes hivatkozás kedvéért az alapegyedek kódhivatkozásokat tartalmaznak. A referencia-egyed a nevet adja meg. (Pl.: mindenütt a pótlékkódot alkalmazzuk hivatkozásként. Van egy pótlékkód/-név egyedünk.) Az osztályozó egyed egy másik előfordulását sorolja be (pl.: gépkocsi-típus). Az egyedek **fajtája** - ahogyan ezt az ismérvet hívom - sok ellenőrzésre ad alapot, főleg a hivatkozási integritás terén.

A modellezési koncepciók közül egyedül az [I32] próbált meg hasonló elképzelést megfogalmazni, erőfeszítéseimmel párhuzamosan. A kiterjesztett relációs modell azonban még mindig nem elég gazdag és kimutathatóan inkonzisztens (B.függelék).

Koncepciómban alaposan elemezzük az egyed **normálformáját** és a feldolgozásokkal való összefüggését. Utóbbi esetben azt, hogy vonatkozik-e rá minden, az 'életciklus' alapján és az egyed stabilitása alapján elvárható **feldolgozásmód**.

Tulajdonság. A SYDES-ban az információkkal leírni akart jelenségek (objektumok) sajátosságainak absztrahált, tipikus tükörképét hívom

tulajdonságtípusnak. A konkrét sajátosság reprezentánsa az érték. A tulajdonságtípus az előfordulások (értékek) adott halmazával bír.

Szemantikai szempontból különbséget teszünk alap, konstans, származtatott és szöveg **típusú** tulajdonság között. Az egyedeket az alaptulajdonságok (cikknév, vevőcím, rendelésszám stb.) írják le. Ezek adott értékhalommal rendelkeznek a kérdéses egyed vonatkozásában. Értékük közvetlen megismerés, ill. közlés eredménye. A konstans hasonló tulajdonság, de egyedtől független az értéke és így egyedhez nem is kapcsolódik.

A származtatott tulajdonság értékhalomjának elemeit felhasználói (matematikai/logikai) műveletekkel nyerjük. Ezek közül egyesek közvetlenül egyed-előfordulásokhoz kötődnek. Tehát azoknál kell modellezni, de a hatékonyság dönt ottani kezelésükről (a származtatott tulajdonság tárolása, pl. a rendelésértéké, ui. speciális redundanciát jelent). A szöveges tulajdonságot technikai célokra adjuk meg - nincs egyedben. A tulajdonság típusa kereszt-ellenőrzésekre nyújt lehetőséget. Konstans és szöveges tulajdonság nem lehet egyedben. Nem célszerű, ha az egyed kulcsa vagy annak része származtatott stb.

Egy másik szempontból megkülönböztetünk kód, dátum, név, érték, mennyiség, szöveg stb. tulajdonságot. Ezeknek az értelmére nem térek itt ki. Nyilvánvaló azonban, hogy a kódokat, neveket, dátumokat érték/ábrázolási szabályok kötik. Ezeknek a megfogalmazására ad módot ez az osztályozás, amit a tulajdonság **fajtájának** neveztem.

A modellezési koncepciók hasonló osztályozást nem tartalmaznak. Nem törődnek az egyedhez nem kapcsolt tulajdonságokkal és elhanyagolják a származtatott adatok modellezésének a problémáját. A koncepciómban alapos elemzésre kerül a tulajdonság ún. szerepe, vagyis az egyedhez való kapcsolat minősége. Külön erőfeszítésekkel követem nyomon a SYDES-ban a származtatott tulajdonságok modellezését (ld. a tulajdonság-struktúráról!).

3.1.4 A SYDES ADATMODELL ÖSSZEFÜGGÉSKATEGÓRIÁI

Szóhasználati, kezelési, szerkesztési, leírási és egyéb szempontokból a SYDES-koncepcióban különbséget teszünk viszonyoknak és struktúráknak nevezett összefüggések között. Ez is a szemantikai és

elemzési gazdagságra való törekvés tényezője. **Viszonynak** az inhomogén összefüggést nevezem, amely eltérő kategóriák közötti m:n-es kapcsolat. **Struktúrának** a homogén összefüggést hívom, amely egyazon kategória előfordulásai közötti m:n-es kapcsolat. A különbségtétel nem lényegtelen. Pl.: csak struktúrák esetében beszélhetünk előzményről, szintről és a legtöbb esetben ciklusról (ez alól akad kivétel is).

Az adatmodell legfontosabb összefüggése az **egyed-tulajdonság-viszony**. A viszonykategória előfordulásai adott jelenség specifikus vonásait írják le úgy, hogy az egyed-reprezentánsokat a tulajdonság-reprezentánsokhoz kötik. A két tükörkép m:n-es viszonyáról van szó. Ezt azért kell hangsúlyozni, mert egyes tervezési eljárásokon a tulajdonságokat csak hierarchikusan, az egyedekhez kötötten lehet megadni. Ez redundanciára és inkonzisztenciára vezet, amint azt a T21 tételben kifejtettem.

E viszonyokkal kapcsolatosan rögzítjük a **függéserőt**. Ez mutatja, hogy a tulajdonság az egyed minden előfordulására értelmezhető-e (erős) vagy sem (gyenge). Az azonosító és minden része értelemszerűen erős. A függéserő koncepciója lehetővé teszi a normálforma 'álproblémák' kiküszöbölését (vö. a 4.4.2 alponttal) és megalapozzák a kapcsolati szemantikát is.

A másik osztályozási ismértv a **stabilitás**, ami azt mutatja, hogy az egyed előfordulásainak 'életciklusában' az adott tulajdonság értéke módosulhat-e. Az azonosítók stabilak. Az eseményjellegű egyedeknek a leíró tulajdonságai is azok. Ennek az ismértvnek a feldolgozások tervezésekor lesz szerepe.

A SYDES-módszerben igen sokoldalú ellenőrzésre kerül, hogy mikor kapcsolható össze egy egyed és egy tulajdonság. Külön elemzési információ rögzíti a tulajdonságnak az egyedekkel kapcsolatos általános és speciális feladatát, ún. abszolút és relatív **szerepét**. (Vigyázat! Ez nem a szerepnév!) Alapvetően egy tulajdonság egy egyedben lehet azonosító, ill. leíró szerepű (pl.: rendelésszám, ill. - dátum). Mivel az összetett azonosító (tételkód) nem kizárt, külön szerep lesz az azonosítórész (rendelésszám és cikkszám). Ha a tulajdonságot csak egyetlen egyed viszonyában vizsgáljuk, akkor ezek az alapvető szerepek (a SYDES egyéb lehetőségeit, pl. az alternatív kulcsot, nem érintem). A relatív szerep a viszony jellemzője.

A tulajdonságok nemcsak egy egyedhez kötődhetnek. Az abszolút szerep a legfontosabb feladatot adja meg (a SYDES-ban szigorúan kötött szabályok szerint). A fontossági sorrend ez: azonosító, azonosítórész, leíró. (A kulcsot alkotó elemek kevésbé rugalmasan változtathatók.) Ha a tulajdonság több egyedben leíró, egyben pedig azonosító, akkor abszolút szerepe az utóbbi. Mindez módosítja a relatív szerepet. Ha a tulajdonság egy egyedben kulcs, akkor más egyedekben kapcsoló (leíró, illetve kulcsrész).

A SYDES szerepkonceptiója a modellezésben egyedülálló. Lehetővé teszi az egyedek közötti összefüggések automatikus meghatározását, optimalizálását, a normálforma-elemzést, a kereszthivatkozási korlátok struktúrális meghatározását. Ezzel szemben a kiterjesztett relációs modell [I32] 'hivatkozási integritás' koncepciója rosszul megalapozott (B.függelék). Megjegyzendő, hogy a szerepeket a SYDES automatikusan generálja és jelzőben állítja be.

A szerepen kívül a viszonyt jellemzi a **normálforma elemzésjel** is. A SYDES beállítja az egyed normálformáját. Itt azt mutatja, hogy a konkrét viszony okoz-e problémát vagy sem.

A másik fontos kategóriaösszefüggést **kapcsolatnak** nevezzük. A kapcsolattípus egyed típusok összefüggését mutatja, tehát voltaképpen **egyedstruktúra**. A SYDES-ban a kategóriát kétféleképp használok.

Saját értelmezésem szerint két egyed között kapcsolat van akkor és csak akkor, ha az egyik egyed kulcsa a másikban leíró vagy azonosítórész relatív szerepű, ill. ott van a szerepneve (ld.: alább). Ilyen alapon megkülönböztetek 'van-neki' ('has a') **birtoklási**, illetve és 'az-is-az' ('is a') **osztályozási** kapcsolatot. Ennek jelét a SYDES automatikusan állítja be. Az is egyértelmű, hogy ilyen alapon csak 1:N és 1:1 fokú kapcsolatok értelmezhetők. Ezeket az egyértelmű kapcsolatokat a SYDES automatikusan generálja. Meghatározza azok **fokát** is.

A SYDES támogatja a Chen-féle modellt is azért, hogy nagy rendszereknél előzetes modellt lehessen készíteni. A tervező tehát maga is kijelölhet kapcsolatokat. Ilyenkor maga adja meg annak jellemzőit, így a fokot is. M:N fokú kapcsolat meghatározására is van lehetőség. A végső adatmodellben azonban csak a SYDES által generált kapcsolatok szerepelnek.

A kapcsolatok fontos tulajdonsága az **opcionális**. Ez azt mutatja, hogy az egyik kapcsolt egyedtípus minden előfordulásának konkrét

kapcsolatelőfordulást kell-e létesítenie a másik egyed típus legalább egy előfordulásával, és megfordítva.

A kapcsolat fokát és opcionálitását a legtöbb koncepcióban [pl. I31 és I35] egyetlen jellemzőnek tartják. Könnyű kimutatni, hogy ez a felfogás a SYDES-módszerénél szemantikailag szegényebb és egy különös hibára ad alapot.

A jelzett irodalmakban a kapcsolatfajta mutató 'x:y' jelölésben az 'x' és az 'y' egyaránt három értéket vehet fel. Ezeket: 0, 1 és N. Evidens, hogy a 0 jelöli az opcionálitást. Világos, hogy ekképpen $3 \times 3 = 9$ kapcsolatfajta adható meg. A SYDES-ban az 'x' és az 'y' értéke csak 1 és N lehet. Tehát 4-féle fok létezik. Ugyanakkor a kapcsolat mindkét oldalán lehet opcionális vagy kötelező (4-féle változat). Együttesen tehát 16 fajta kapcsolat különböztethető meg a SYDES-ban.

A különbség lényege egyszerű. A hivatkozott módszerek kifejezik azt, hogy az előfordulás adott oldalon nem feltétlenül kapcsolt (0). Ez az érték azonban elfedi a megengedett maximumot. Vagyis azt, hogy a többi (!) előfordulás közül csak egy (1) vagy több is (N) kapcsolódhat-e a másik oldalhoz.

A kapcsolatnak van egy további, fontos osztályozó ismérve a SYDES-ban, a szemantikai gazdagság fokozására. Az alapegyed és a családfe, specializált, ill. aggregált egyed között hasonló **típusú** a kapcsolat. Ezeket és az alapkapsolatokon kívül megkülönböztetnek visszamutató és többszörös kapcsolatot. Az előbbi egyazon egyed típus előfordulásai közötti M:1-es vagy 1:1-es összefüggés tükre. Az utóbbi a két egyed közötti, adattal le nem írható minősített viszonyok reprezentánsa (pl.: a személy lakik/dolgozik a településen).

A kategória tárgyalásának lezárása előtt ki kell jelentenem:

T32 A SYDES-koncepció az ismert modellezési irányzatoknál egyértelműbb és szemantikailag gazdagabb lehetőséget kínál az egyed-összefüggések modellezésére.

A hálós modellben a 'set' fizikai fogalom. Nem épül a kötelező 'set item' fogalmára, ami lehetne szerepnév is. A kapcsolatok foka, opcionálitása és típusa korlátozott. (Ez az utóbbi kitétel minden egyéb modellirányzatra is érvényes.) A relációs modellek ezt a kategóriát nem is ismerik, holott felfedezték a 'foreign key' fontosságát. A modell megengedi az m:n-es kapcsolódást is, amiből a 4.5.4-ben kifejtett

egyértelműségi problémák fakadnak. Jellemző, hogy a szemantikai gazdagságra törekvő RM/T modell (B.függelék) is csak korlátok névvel nem hivatkozható laza szövevényének tartja a kapcsolatot és nem kategóriának. Magában a Chen-modellben a három alapkategória ellentmondásosan épül egymásra.

A SYDES-et sem tartom szemantikailag eléggé gazdagnak a sokoldalú összefüggések kifejezésére. A szemantika a viszonyoknak igen sok válfaját ismeri. Ezeket még vizsgálnom kell.

Amint az egyedek, úgy a tulajdonságok között is meghatározhatunk homogén összefüggést, ún. tulajdonság-struktúrát.

T33 Mivel a közismert modellezési koncepciók hasonló kategóriát nem tartalmaznak, sőt, azt eleve kizárják, a SYDES koncepciót a többinél szemantikailag gazdagabbnak és egyértelműbbnek szabad tartanom.

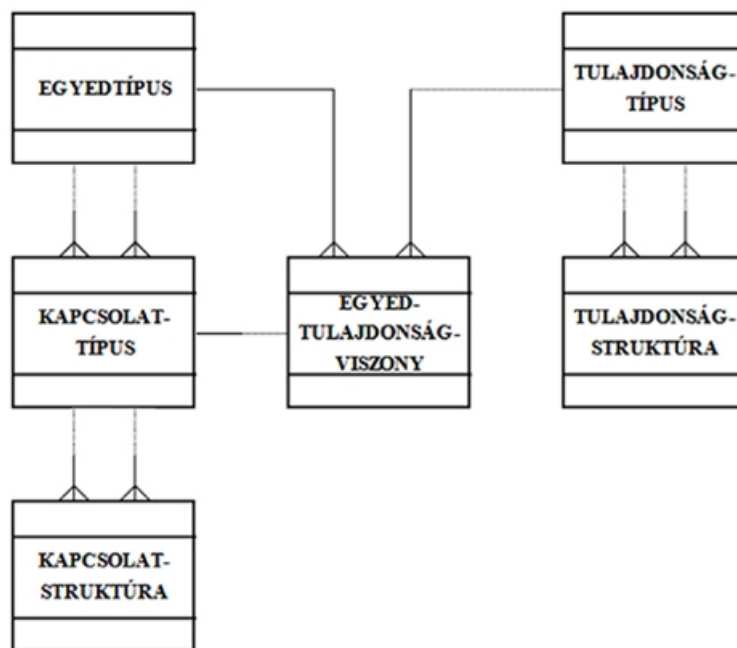
Csoport és **származtatási** tulajdonság-struktúrát különböztetek meg, de a felhasználó-adta struktúra-típust sem zárom ki.

A **csoportot** a relációs modell - alapgondolata miatt - elveti. A 4.2 pontban alaposan elemzem ennek súlyos következményeit. A hálós modell a csoportot külön kategóriának tartja, de nem kezeli struktúraként. Ennek több negatív hatása van:

- . Ha a csoport és a tulajdonság két külön kategória, akkor külön egyed-csoport viszonykategóriát is fel kell állítani. Aktuálisan minden tulajdonság-viszonyt meg kell duplázni.
- . Ilyen módon lehetetlen az egyed normálformájának a megállapítása (amit a csoport egyébként nem zárna ki!).
- . Ha egy fogalomból csoportos tulajdonság lesz vagy fordítva, akkor a két kategória közti átvitel miatt extra adatkezelésekre kerül sor.
- . Nehézkessé válik a cikluselemzés (tartalmazza-e a csoport saját magát beépülőként?). Egyáltalán: elveszik a nem lényegtelen szintfogalom (hányadik szinten van a tétel?).
- . A csoport hierarchikus konstrukciónak tűnik, holott valójában hálós. A csoport eleme másik csoportban is lehet tag.

A SYDES a csoportokat is tulajdonságokként kezeli. Számos, itt nem részletezhető szabály vonatkozik arra, hogy egyes tulajdonságok mikor alkothatnak egymással struktúrát. A normálforma elemzés során a

SYDES a csoportokat automatikusan felbontja. Megfelelő átfedés-elemzés után teljes normálforma-elemzést hajt végre. A csoportokat redundancia és ciklus szempontjából külön is lehet elemezni.



3.1 ábra: A SYDES adatmodell-kategóriái

A SYDES további alapvető konstrukciója az ún. **előzményháló**. Megadhatjuk egy származtatott tulajdonság tényezőit, ha azok közül valamelyik származtatott, annak a tényezőit stb. Ez így származtatási családfa. Mivel azonban egy tétel több másiknak is lehet tényezője, az összefüggés hálós. Itt dinamikai viszonyról van szó: a tényezőt ismernünk kell, mielőtt megismerhetjük a belőle származtatott tételt. Ezért használjuk az előzmény (precedencia) megnevezést.

A többi koncepciótól eltérően nálam a precedencia-struktúra is az adatmodell része. Ez alkalmat ad ismét a redundancia és a ciklus elemzésére (egy tétel se származzon önmagából!). A származtatott tulajdonságok struktúraként kötődhetnek az eljárásokhoz és azok algoritmusaival keresztellenőrizhetők. Most vizsgálom a származtatott

tulajdonságok hivatkozási referenciájának automatizálását. Az ilyen tulajdonságok egyedére is hivatkozni kell. A tényezőkből és az algoritmus jellegéből ez a referencia levezethetőnek látszik.

Ezzel az adatmodell vázának az ismertetését befejeztem, bár a SYDES-kategóriák még sok jellegzetességgel bírnak. A kategóriák összefüggéseit szemlélteti a 3.1 ábra az előző oldalon.

3.2 A SYDES FOGALMI FELDOLGOZÁSMODELLJE

A SYDES feldolgozásmo­dell-kon­cep­cióját nem lehet olyan világosan és egzak­tu­l összevetni más koncepciókkal, mint az az adatmodell esetében történt. Ennek az az oka, hogy a fogalmi szintű feldolgozásmo­dell gondolata az adatmodelléből nőtt ki és saját gyökerű elvei nincsenek. A hagyományos feldolgozástervezés gondolatait próbálják meg sokan felvetíteni a fogalmi szintre. Az első sikeres kísérletek után [I11] a kutatás felgyorsult és az [I12] már önálló gondolatokról számol be.

A feldolgozásmo­dell fontosságát, adatorientációjú lévén, később fedeztem fel és ezen a téren kutatásaim nem kiterjedtek. Mindenesetre a [H19] szoftver már tartalmazta az 'eljárásmodellezést' is. Modell-kon­cep­cióm lényegét a 3.2.2 alpontban mutatom be. Ezt szervesen egészíti ki az adatalmodell tárgyalása a 3.2.3 alpontban, ahol az adat- és feldolgozásmo­del­lek összefüggéseit tárgyalom. Legelőször azonban néhány elvi megfontolást teszek közzé a feldolgozásmo­del­lezés egészével kapcsolatosan (3.2.1 alpont).

3.2.1 A FELDOLGOZÁSMODELLEZÉS CÉLJAI, IRÁNYZATAI

Az 1.1.5 alpontban szóltam az információs rendszer feldolgozás vetületéről, mint olyanról, amely az információs eseményeket és tevékenységeket öleli fel. Ennek megfelelően feldolgozásmo­del­lnek a fogalmi szintű esemény/tevékenységtervet nevezem; feldolgozásmo­del­lezésnek pedig az erre irányuló erőfeszítések együttesét hívom.

A fogalmi szintű feldolgozásmo­del­lezésnek több célja van:

- . Kiinduló célja a felhasználói **feldolgozások rendszerezése**. Vagyis az események és tevékenységek logikai és dinamikai elrendezése. Ehhez adhat gondolkodási/megvalósítási keretet a következő alponthoz tárgyalt funkcionális hierarchia, ill. a 3.3 pontban bemutatott funkcionális háló.
- . A SYDES-koncepcióban kimondott cél a meghatározott feldolgozás-modell **optimalizálása**. Nem elég egy működőképes tervet összeállítani. A tervnek érthetőnek, egyértelműnek, teljesnek, minimálisnak és megvalósíthatónak kell lennie.
- . A fogalmi modell a **logikai feldolgozástervezés** alapja. A végrehajtandó információs tevékenységek tartalmának és összefüggéseinek az ismeretében, az adatmodellel való kapcsolatot elemezve dönt a tervező a megvalósítási módokról.
- . Az előzőhöz kapcsolódóan a feldolgozásmodellezésnek kimondott célja a **fogalmi adatmodellel való egyeztetés**.

A feldolgozási modellezési koncepciókat e célok alapján illenék értékelni. Ennek van egy igen komoly akadálya. A feldolgozás-modellezésben nem alakultak ki olyan iskolák és irányzatok, mint az adatmodellezésben. Így az általam itt használt osztályozás és elemzés a szakirodalomban nem lelhető fel:

- . **Petri-háló.** Ez az irányzat az esemény-tevékenység-feltétel hármason alapul [I39]. Néha esemény-művelet-üzenet hármast alkalmaznak [I40]. Az utóbbi jól mutatja a koncepció alapvető gyengéjét, amit az [I12] vitáján is felvetettek: a fogalmi tervezéshez képest mikroszinten fogja meg ez az elképzelés a tényezőket. A feltételekkel ellátott előzményháló, ha azt nagyobb egységekre vetítjük fel, feltétlenül hasznos. A SYDES-ban így alkalmazom.
- . **Életciklus-elv.** Az információs egységek életciklusáról már korábban szoltam (2.4.3 alpont). Az esemény-orientált módszerek egy részében [I41] ezt az elvet kombinálják a struktúrált tervezésben használt hierarchikus lebontás elvével. Például:

változnak a cikkek és a vevők is (1.szint). Cikkek születnek, módosulnak és meghalnak (2.szint). A cikkek születésének van a, b és c módja (3.szint).

Ez a koncepció, mint minden hierarchikus elv, magában hordozza a redundancia, ill. a helytelen kapcsolódás veszélyét. Példa: a számla kiküldése értékesítési vagy pénzügyi esemény-e? Nincsenek szabályai a hálósan összefüggő események célszerű hierarchikus lebontására.

Ugyanakkor a SYDES-ban is felhasználom az életciklus-elvet az adat-és a feldolgozásmódel kapcsolata teljességének az ellenőrzésére.

- . **Adatfolyam.** Az újítás jegyében egyesek hajlamosak elvetni ezt a hagyományos konstrukciót (mint az [I12] vitáján is). Az adatfolyam felhasználók, tevékenységek és adategységek viszonyát írja le, dinamikus összefüggésben. Szemben az előző két elvvel, a bemenet és kimenet fontos szerepet játszik. Külön egységek az 'adattárak'.

Kétségtelenül fennáll a veszély, hogy az adatfolyam fizikailag meghatározott lehet. Nem csekélyebb a veszélye a logikai függőségnek sem, vagyis annak, hogy az adatfolyam egyéni szemlélettől befolyásolt.

Megfelelő megfogalmazás és az adatmódelrel való keresztellenőrzés esetén ezek a hibák kiküszöbölhetők. A SYDES-ban nagymértékben építék az IPO - bemenet-feldolgozás-kimenet - logikájára. Azt kiegészítem az adatfolyamból hiányzó esemény-konstrukcióval.

Léteznek még egyéb részelméletek is, amelyek közül a tevékenységek hierarchikus lebontásán alapuló a tárgyalt második koncepcióhoz hasonlóan értékelhetők. Függetlenül az elméletek gyökereitől és struktúráitól, egy igen fontos közös vonást felfedezhetünk bennük, ami a SYDES-ra is jellemző. Ez egy sajnálatos, de természetes negatívum, aminek a kifejtése a következő alpontban történik meg.

3.2.2 A SYDES FELDOLGOZÁSMÓDELLEZÉSI KATEGÓRIÁI

Az információs rendszer célja az, hogy a felhasználók számára információvá értelmezhető adatösszefüggéseket produkáljon. Az adatok ilyen célból történő feldolgozását nevezem információs **tevékenységnek** (1.1.2 alpont). Módelről lévén szó, pontosabb

kifejezéssel tevékenységtípust kellene mondani. Abban az értelemben, hogy a tervező a tevékenységek közös, lényeges és tartós vonásai alapján alkotja meg ezeket a típusokat. A tevékenység egy-egy végrehajtását, a konkrét aktust tevékenységelőfordulásnak is hívom.

Adott környezetben, adott időszakban a típus meghatározott előforduláshalmazzal rendelkezik (végrehajtási gyakoriság).

P31 Szabó és Kovács belépésekor végrehajtjuk a 'felvétel' nevű tevékenységet. Ugyanazt a tevékenységtípust végezzük. A két személy konkrét alkalmazása (felvétel-előfordulás) mégsem teljesen azonos. Kovácsnak, pl. nincs családi pótléka. Egy évben 50 dolgozót veszünk fel. Ez az előforduláshalmaz mérete.

(N.B.: Fogalmi szinten a felhasználói eljárásokat tekintem alap-egységnek. Ezek nem tévesztendőek össze a programrészként említett eljárásokkal ('procedure'). Ezért másutt az eljárásmodell/-modellezés kifejezést használok a feldolgozásmodell/-modellezés helyett.)

Ezen a ponton lép fel a feldolgozásmodellezés legnagyobb dilemmája, ami a tevékenység, ill. eljárás szó tartalmának a viszonylagosságából fakad. Egy szorzás éppen úgy tevékenység, mint az inputelőkészítés, a program vagy a számlázási osztály információs működése. Felmerül tehát a kérdés, hogy mi az információs **tevékenységi egység**?

Erre a kérdésre választ az irodalomban nem találtam. (Az [I13] vitáján Nijssen úr közölte, hogy nem is fogok.) Így a SYDES-konceptióban kénytelen voltam saját elméleti utat keresni. Ezt majd az alpont közepén mutatom be. Itt egyszerűen leszögezem, hogy információs tevékenységi egységnek azt a feldolgozást nevezem, amely a bemeneti eseménytől a kapcsolt kimeneti eseményekig tartó műveleteket öleli fel.

A meghatározás lazaságából is, de a feldolgozás-tervezés ősi gyakorlatából is következik, hogy az előbb meghatározott egységhez képest a tevékenységek különböző **absztrakciós szintjeit** célszerű alkalmazni. A SYDES-konceptióban a konkrétan végrehajtandó eljárásokat eljáráscsoporthoz, azokat funkciókba, alrendszerbe, majd rendszerekbe sorolom; illetve modulokra, majd rutinokra bontom le. Ezek a szintek lazán meghatározottak; értelmezésük az alkalmazó feladata. Nem kötelező mindegyiket használni, de a hierarchiát be kell tartani. A lebontást szemléltesse egy triviális példa!

P32 Vállalatunkban (rendszer) az értékesítés (alrendszer) számláz (funkció). Külön kezelik az export-számlákat (eljáráscsoport). A számla kiállításakor (eljárás) külön ellenőrzik a kedvezményeket (modul). A valutaelszámolás algoritmusa mindig azonos (rutin).

Ez a 'hierarchia' a SYDES-konceptióban valójában háló, mivel az alacsonyabb szintű egység több magasabb szintűhöz tartozhat. (Aktuális esetben kérdés, hogy ez célszerű-e. Lásd a struktúrák redundanciáját alább!)

A tevékenységekhez meg kell adni azok **feldolgozási célját** (lekérdezés, karbantartás stb.). Az adatmodellhez kapcsolódóan ez segít az életciklus teljességének a megállapításában. Az eljárásoknál a SYDES két elemzőjelet kezel. Az egyik azt mutatja, hogy megadták-e az eljárás bemenetét és kimenetét. A másik az ún. navigációjel. Lásd a következő alpontot!

A SYDES-konceptióban a változási jelenségeket események tükrözik. Pontosabban eseménytípusok, amelyeknek a bekövetkezése jelenti az eseményelőfordulást. Adott időszakban, adott környezetben az eseménytípus meghatározott előforduláshalmazzal rendelkezik. Példaként gondoljunk a dolgozó felvételekre!

Mint azt korábban kifejtettem, az információkezelési igényt is változásként értelmezem és eseményként modellezem csakúgy, mint az információs tevékenység eredményeként beálló változásokat.

Az eseményeket a vizsgált rendszer szempontjából **külsőkre** és **belsőkre** osztályozom. Az előbbiek kívülről érik a rendszert vagy oda hatnak (pl.: rendelés, számlázás). Az utóbbiak nem lépnek ki a belső környezetből (pl.: belső megrendelés). Más szempontból különbséget teszek **egyed- és állapotszerű** események között. Az előbbiek tulajdonságokkal írandók le és kapcsolataik vannak az adatmodellhez is. A SYDES két elemzésjellel mutatja, hogy az esemény kapcsolatai az egyedekhez, illetve tevékenységekhez optimálisan meghatározottak-e.

Itt kell visszatérnem a tevékenység-fogalom viszonylagosságára. Mondanivalómat egy rendelésfelvételi példa szemlélteti.

P33 Az 'A' vállalatban a beérkező rendeléseket rögzítik és nyugtázó levelet küldenek a megrendelőnek. A rendeléseket valamikor később feldolgozzák.

A 'B' vállalatban azonnal megállapítják, hogy a rendelés kielégíthető-e. Ennek értelmében értesítik a vevőket. A rendelt tételeket le is foglalják.

A 'C' vállalatban a lefoglalt tételekre szállítási diszpozíciót állítanak ki. A hiányzó tételekre belső rendelést adnak fel.

A 'rendelésfelvétel' fogalma a három vállalatban alapvetően más. Ez az eljárás, mint egység, más-más terjedelmű. Sem elméleti, sem gyakorlati elvet nem lehet lerögzíteni arra nézve, hogy melyik eljárás a helyesen meghatározott. Az eljárás és az esemény tehát nem olyan objektívan behatárolható egységek, mint az adatmodell tényezői. E viszonylagosságból fakad változékonyságuk is. Még alapos kutatást igényel az, hogy objektívabb tevékenység egységeket alkalmazzunk. A SYDES-konceptióban - egyelőre - csak egyetlen irányelvet tartottam hasznosnak megfogalmazni.

Az adatok feldolgozása során meglévő adatösszefüggésekből újakat, majd azokból ismét újabbakat állítanak elő. (Vö. a 3.1.4 alpontban tárgyalt adat-precedenciával!) Nevezzük azt az adatösszefüggést, amit a felhasználó már maga kíván információvá értelmezni **végsőnek**, a többit **átmenetinek**. Együtt ezeket az 'állomásokat' **adatfeldolgozási láncnak** neveztem [H17]. Számos feltétel függvénye, hogy a felhasználó e láncba hol kapcsolódik be és abból mikor lép ki.

Az, hogy milyen (fokú) adatösszefüggést tart a felhasználó általa értelmezhetőnek, döntései megalapozójának, rendkívül szubjektív dolog. Ugyanakkor erre a tényezőre hatnak szakmai ismeretek, szokások, rendszeren belüli és kívüli szabályzók is. A tervezőnek mindezt mérlegelnie kell. Nem tévedhetek nagyot, ha kijelentem: az információs tevékenység azoknak a műveleteknek az összessége, amelyek az alkalmazó által kiváltónak tekintett eseménytől az általa értelmezhetőnek vélt kimeneti adatösszefüggések összeállításáig tart.

Az események és tevékenységek **esemény-tevékenység-viszonyok** láncával kapcsolódnak egymáshoz. Egy esemény, amely egy adott tevékenységet lezár/elindít, újabb tevékenységet válthat ki/fejezhet be, hacsak az esemény nem külső. Ezek a viszonyok együtt adják ki a bemenet-feldolgozás-kimenet feldolgozási hálót, az **IPO-hálót**.

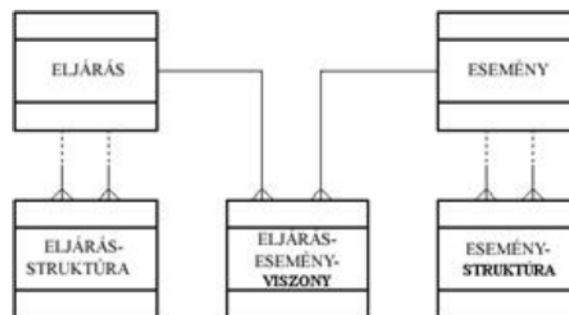
E viszonyokban rögzítjük, hogy az adott feldolgozást az esemény **elindítja** vagy **lezárja**. Külön utalunk arra, hogy a tétel kötődik-e

feltételhez. (Vö. a Petri-hálóval!) Elemzésjelek mutatják, hogy struktúráisan (statikusan) megfelel-e a tétel az IPO szabályoknak, ill. dinamikusán a navigációnak. (Lásd a következő alpontot!)

A SYDES-koncepcióban definiálható a tevékenységek összefüggése, a **tevékenység-struktúra**. Kétféle ilyen struktúrát javaslok. A (kvázi)-hierarchikus szerkezetben a rendszer, alrendszer, ..., eljárás stb. absztrakt felépítés adható meg. Erről tudni kell, hogy valójában háló és ekképpen elemezhető a SYDES redundanciakereső programjával.

(Ez a magasabb szintű egységek közös alárendeltjeit mutatja ki.) Az is világos, hogy ez a struktúra igen szubjektív.

A tervező statikus és dinamikus struktúrákat jelölhet ki ún. eljárás-hálóban. Ebben azonos absztrakciós szintű tevékenység-egységek összefüggései kerülnek megfogalmazásra. Így követhető a struktúrált tervezés elve is. Azonban a SYDES-ban ez a struktúra elsősorban a tevékenység-precedencia meghatározására szolgál. Erre a célra két szintet javaslok.



3.2 ábra: A SYDES feldolgozásmodell-kategóriái

A tényleges eljárások szintjén lehet megadni, hogy a végrehajtandó tevékenységek miként követik egymást. Magasabb tevékenység-egységek szintjén javasolt a funkcióháló. Összetett rendszerek esetén ez az előzetes tervezés eszköze. Pl.: ez írja le, hogy az értékesítés-tervezés funkciójának a bemenete a rendelésállomány-kezelés és a darabjegyzék-feldolgozás.

A fentihez hasonlóan kezelem az ***eseménystruktúrákat***, mint előzményviszonyokat. Ezek is elemezhetők redundanciára. Egy másik SYDES-program a ciklusokat szűri ki.

Ezzel a feldolgozásmódel vázának az ismertetését befejeztem. A SYDES feldolgozásmódel véges számú feldolgozás (nevezzük itt eljárásnak!) és esemény szervezett együttese. A nevezett kategóriák összefüggéseit szemlélteti a fenti 3.2 ábra.

3.2.3 A SYDES ADATALMODELL-KATEGÓRIÁI

Ez az első pont, amikor expliciten rá kell mutatnom a modellezési módszer kialakításának hihetetlen elméleti és gyakorlati nehézségeire.

Abból kiindulva, hogy a tervezőnek nem csak a fogalmi adat- és feldolgozásmódel kell megterveznie, hanem a kettő harmonikus összefüggését is. A feldolgozás adatokat használ. Ez a trivialitás a metasíkon úgy fogalmazódik meg, hogy a feldolgozási és az adat kategóriák összefüggnek.

Szemben más módszertanokkal [pl. I13], a SYDES tervezésekor ugyancsak ügyeltem arra, hogy a lehetséges minimumon tartsam a kategóriák számát. A SYDES-ban a csoport nem külön kategória, amint hogy a rendszer vagy a modul sem az. Annak ellenére, hogy a fogalmi kategóriákat valóban a lehetséges minimumon tartom, mégis gondjaim vannak a modellek kapcsolásával.

A tevékenységek egyedekre, azok tulajdonságaira és kapcsolataira vonatkoznak. Ezen belül is érintenek tulajdonságstruktúrákat, illetve egyed-tulajdonság-viszonyokat. Eddig ötféle kapcsolódást említettem. Mindezt megduplázhatom azzal, hogy az eseményekre ugyanez vonatkozik. Még csak a háromszorozásnál tartok, ha felvetem: miért nem adott tevékenység adott eseménye szempontjából modellezzünk?

Az észszerű minimumon tartás jegyében csak két újabb kategóriát vezetek be, a modellezés két eltérő síkján. Összetettebb rendszernél célszerű készíteni előzetes modellkeretet (nem tévesztendő össze a keret-módellel!), vagyis egy átfogó módel-vázlatot. Ez az alapvető egyedeket, azok kapcsolatait és az egyedekre vonatkozó legfontosabb tevékenységcsoportokat öleli fel. (Ld. a BSP-re vonatkozó kitétel a 3.1.2 alpontban!) Az ilyen 'keret' (frame) alkalmat adhat a tevékenységek előzetes hierarchikus lebontására éppen úgy, mint az

adatmodellezési feladat ésszerű megszervezésére. Mindenképpen kapunk egy gondolkodási keretet (innen a név).

Ennek a keretnek a támogatására használom a **tevékenység-egyed-viszonyt**. A viszonyt jellemzi a **kezelésmód**. Az, hogy a tevékenység lekérdezi, karbantartja stb. az egyedet. Leírja a **hozzáférejmód**, ami azt tükrözi, hogy a tevékenység az egyedtípus egyetlen, szelektíven néhány, vagy összes előfordulását érinti-e. Elemzésjel mutatja az egyed helyét az adatkezelési-gráfban (navigációban, [H23, H24]).

Végeredményben azonban a feldolgozás nem egyedre, hanem az egyed tulajdonságaira, illetve összefüggéseire vonatkozik. Mint azt a korábbiakban említettem, az egyedet és a kapcsolatot is mindig egyed-tulajdonság-viszony reprezentálja. Így a SYDES-konceptióban az eseményeket mindig tulajdonságokhoz kötöm. Ez a megoldás kettős konstrukcióra ad alkalmat.

Rögzíthető az a tény, hogy az esemény is lehet információval leírandó (tulajdonsághoz kapcsolt) jelenség tükörképe. A rendelés eseménynek felsorolhatók a (nem-normalizált) tulajdonságai. Egyidejűleg ez az összefüggés a SYDES-ban metaviszony, vagyis viszonyt kapcsol más tényezőhöz. A kategóriával kifejezhető az is, hogy az esemény tulajdonságát melyik egyedben kell kezelni. A rendeléskarbantartás adott esetben vonatkozhat a rendelés és a vevő egyedben lévő vevőkódra egyaránt.

Az **esemény-tulajdonság-egyed** viszonyt a normálforma és a navigációs elemzési kódok jellemzik. Az előbbi a 4.fejezetben részletesen tárgyalom. A navigációs koncepcióját igen régóta (1976) használom. Maga a navigáció azt írja le, hogy egy feldolgozás melyik egyedek milyen tulajdonságait milyen sorrendben kezeli és azokon milyen műveleteket hajt végre. Még dolgozom azon, hogy a különböző kategóriáknál lévő elemzésjelek alapján miként lehet a navigáció teljességét és helyességét bizonyítani. U. i. ez adhat alapot arra, hogy a feldolgozási tevékenységsort pusztán a jelzett viszonyok megadásával automatikusan generáljuk.

Ezt alátámasztandó, a nevezett viszonyokat két további jellemzővel írom le. Az egyik az egyed tulajdonságán végrehajtandó műveletet jelöli ki. A másik az esemény tulajdonságának bemeneti, ill. kimeneti szerepét mutatja. Együttesen azt, hogy a kimenetre a tulajdonság modellből, bemenetből, származtatásból vagy programból kerül-e.

3.3 ALTERNATÍV TERVEZÉSI ALGORITMUSOK

Az algoritmusok bemutatása előtt utalnom kell arra, hogy a SYDES, mint **testreszabott** szoftver tartalmazza az előzőekben (3.3 ábra) ismertetett kategóriákat. **Általánosított** szoftverként viszont lehetővé teszi, hogy a tervező tetszőleges számú saját kategóriát jelöljön ki; azokat a fenti kategóriákhoz kapcsolja; jellemzőkkel írja le és ezeknek értéket adjon. Akit tehát az alább bemutatott algoritmus-változatok nem elégítenek ki, az újabbakat kreálhat illetve alkalmazhat.

Lényegüket tekintve az algoritmusokat egy három-dimenziós ábrán lehetne összegezni. Az alábbi 3.3 ábrához képzeljük hozzá a magasságot, mint harmadik dimenziót!

	Adat	Feldolgozás
<i>fentről-lefelé</i>	Adatmodell dekompozíció	Feldolgozás levezetés
<i>alulról-felfelé</i>	Adatmodell szintézis	Feldolgozás integráció

3.3 ábra: Alternatív algoritmusok

A harmadik dimenzióban azt tükrözném, hogy mind az adat-, mind a feldolgozásmodellezésben létezik globális (keret) és részletes tervezés. Most pedig lássuk, kategóriákhoz kötötten, azok sorrendje szerint az algoritmusokat!

Globális és részletes adatmodell tervezés. Előbbinél a sorrend: egyed/kapcsolat/tulajdonság/tulajdonság-struktúra. Az utóbbinál: egyed/tulajdonság/tulajdonság-struktúra/kapcsolat.

Adatmodell dekompozíció és szintézis. Előbbinél a sorrend: egyed/tulajdonság/tulajdonság-struktúra/kapcsolat. Az utóbbinál tulajdonság/tulajdonság-struktúra/egyed/kapcsolat.

Globális és részletes feldolgozás tervezés. Előbbinél a sorrend eljárás/(hierarchikus)eljárás-struktúra/esemény/esemény-struktúra.

Utóbbinál eljárás/(hálós)eljárás-struktúra/esemény/esemény-struktúra /(hierarchikus)eljárás-struktúra.

Feldolgozás levezetés és integráció. Az előbbi esetben először a globális adatmodellt állítjuk össze és abból vezetjük le az eljárásokat (életciklus-szemlélet). Utóbbiban először a feldolgozásokat tervezzük meg. Az egyedi szemléletekből integráljuk a globális adatmodellt. Amint látjuk, a SYDES rendszermodellje a rugalmas kereten keresztül számos ismert és sok kevésbé használt tervezési eljárást támogat. lényegesen többet, mint az általam ismert tervezési módszerek bármelyike. Ezt tételként is kijelenthetném.

4. RELÁCIÓS NORMÁLFORMÁK A SYDES ADATMODELLEZÉSBEN

A SYDES adatmodellezés célja az, hogy segítse a tervezőt egyértelmű, teljes, minimális, valóságű, áttekinthető és az adott környezetben kezelhető adatbázis-struktúra kialakításában. Ezt a célt a tervező két nagy lépésben éri el. Először fogalmi adatmodellt alkot és azt elemzi. Ezt követően pedig a fogalmat logikai adatmodellre képezi le úgy, hogy közben tekintetbe veszi a leendő adatbáziskezelő legalapvetőbb struktúrálási és kezelési sajátosságait.

A SYDES-módszer kialakítása során kellett figyelni a relációs modellre [I16] általában is, és különösen az azzal kapcsolatos normálformákra és normalizálási eljárásokra. Ezek ui. explicit vagy implicit módon közelebb vihetnek az optimális adatszerkezetekhez. Nem véletlenül szánok komoly szerepet a SYDES-módszerben ezeknek a konstrukcióknak.

A szokásos normálformák és tulajdonságfüggések alapvető ismeretét feltételezem [I19-23]. Így feleslegesnek tartom meghatározásaik itteni ismertetését. E koncepciók többoldalú vizsgálata során a következőket állapítottam meg:

T4 A relációs normálformák adatmodellezési szerepét a szakemberek egy része félreérti. Vannak, akik ezt a lehetőséget alulbecsülik, míg mások túlértékelik. Szükség van tehát a normálformák fontosságának reális megítélésére.

Vizsgálatom az utóbbi feladatra irányult. Másodlagos eredményeként meg merem kockáztatni annak a kijelentését is, hogy a hagyományos relációs adatmodell maga is revízióra szorul. (Az ún. kiterjesztett relációs modell bírálatára a B.függelékben térek ki.) Céлом érdekében a normálformákat és a normalizálási eljárásokat több vetületben is értékelem:

- Vizsgálom, hogy teljesek-e az alapvető normálformák.
- Elemzem a sikállomány hátrányait és a csoportokat.

- Értékelem a normalizálási eljárások következményeit.
- Komoly figyelmet szentelek a modell-szemantikának.
- Kifejtem a normálformák hatását a modell optimumára.
- Végül összefoglalom az elemzéssel elért eredményeket.

Mondanivalómat sok példával fogom szemléltetni. Ezekben az alábbi jelölési konvenciókat alkalmazom:

- A relációk nevei nagybetűsek.
- Az attribútum-nevek nagybetűvel kezdődnek.
- A reláció attribútumai zárójelben felsoroltak.
- Az elsődleges kulcsot $p:x$ jelöli, ahol x a név.
- Az idegen kulcsot $f:y$ jelöli, ahol y a név.
- A csoport tagjait '+' jel köti össze {}-n belül.
- A $C\{X\}$ ill. $C\{X+Y\}$ azt mutatja, hogy $C=X$ ill. $C=X+Y$.
- Az érdektelen attribútumokat '...' mutatja.
- A funkcionális függés jele: $---->$
a függetlensége: $--/->$
a kölcsönös függése: $<--->$
a többértékű függése: $--->>$

4.1 A RELÁCIÓS NORMÁLFORMÁK TELJESSÉGÉRŐL

Az ÁDÁM & ÉVA [H19, H21, H27] segédeszköz kialakítása során (is) kritériumként követeltem meg a relációk kulcsainak az egyediségét. Két relációnak nem lehet ugyanaz az attribútum a kulcsa. Ez a korlát nehézségekhez vezetett, amikor nem zártam ki, hogy a kulcs több attribútumból összetett legyen.

Az egyediség vizsgálata ekképpen nem merülhet ki egyszerű, néven alapuló átfedéselemzésben. Azt is értékelni kellett, hogy két kulcs – a tagok sorrendjétől függetlenül - nem azonos összetételű-e. El kellett kerülni az ilyen helyzeteket:

$$R1x(p:X\{A+B\}) \text{ és } R2y(p:Y\{A+B\}) \text{ vagy } R2z(p:Z\{B+A\})$$

Az összetett kulcsok léte nem-triviális modellezési gondokat vet fel. Nem vizsgálhatók egyszerűen a közöttük lévő funkcionális vagy egyéb függések. Ezek közül sajátosan kell kezelni a kulcs részein alapulókat.

Nem beszélve az ún. **triviális** függésekről, amelyekben az összetett kulcs saját részeit határozza meg így: $X\{A+B\} \text{ ----> } A$ és $X\{A+B\} \text{ ----> } B$.

A modellbeli egyértelműség és könnyű hivatkozás kedvéért az összetett kulcsokat külön attribútumokként határoztam meg és struktúrákban adtam meg azok részeit. Azután az alkalmazások során [H31] meglepő helyzetek váratlan elméleti következtetésekre juttattak. Ezeket mutatom be az alábbi pontokban.

4.1.1 A TELJESSÉGVIZSGÁLAT ALAPJA ÉS MÓDJA

A szakírók jelentős része azt feltételezi, hogy az alapvető normálformák (1-3NF, BCNF, 4-5NF) sora teljes és ezért logikailag konzisztens normalizálást biztosít [I24, 239.o.]. A [H15] kialakításakor végzett vizsgálatom [H18] a feltételezés ellenkezőjét bizonyítja.

T41 Az alapvető normálformák sorozata belülről nem teljes. Emiatt az azokra épülő normalizálási eljárások sem konzisztensek. Be kell látni, hogy a normalizálás főleg szemantikai feladat és nem csak a kivetítés/összekapcsolás matematikai műveletein alapul.

A '**belülről** nem teljes' kifejezés mutatja, hogy nem vitatom, miszerint a normálformák sora az 1NF-fel kezdődik és az 5NF a végső normálforma (projection-join szempontból). Elsősorban a 2NF, 3NF és BCNF sorozat körül csomósodnak problémáim.

A tétel megfogalmazásához az a felismerés vezetett, hogy a 2NF és a 3NF között nincs matematikai értelemben vett lényeges különbség. Ezt jól szemlélteti az alábbi példa:

P401 RENDELÉSTÉTEL ($p:\{f:\text{Rszám}+f:\text{Cszám}\}$, ..., Cikknév)

A függések: $\text{Rszám}+\text{Cszám} \text{ ----> } \text{Cszám}$ (triviális)
 $\text{Cszám} \text{ ----> } \text{Cikknév}$
 $\text{Rszám}+\text{Cszám} \text{ ----> } \text{Cikknév}$

Fogalmazhatunk úgy is, hogy a részleges függés is tranzitív függés. Annak sajátos esete, amelyben a kiinduló függések egyike triviális. A példa alapján logikusan csak az alábbi két gondolatsort követhetjük:

1. A példa harmadik függését részlegesnek nevezzük ugyan, de hívhatnánk azt egyszerűen tranzitívnak is. A részleges és a tranzitív függések által okozott anomáliák és azok feloldási módja is – kivetítés külön relációba - azonos. Tehát nincs alapunk arra, hogy a 2NF és 3NF alakokat megkülönböztessük.

2. A másik lehetőség az, hogy fontosnak tartjuk a 'normális' és a triviális függés közötti különbséget. Ekkor a 2NF és 3NF alakokat eltérőeknek kell tekintenünk. Ami azzal jár, hogy a normálformákat belülről nem-teljesként, inkonzisztens sorként kell szemlélnünk. A normálformák belső teljességének az ellenőrzéséhez célszerű felhasználni a következő döntési táblázatot [H18]:

	1	2	3	4	5	6	7	8
A ----> B	P	P	P	P	D	D	D	D
B ----> C	P	P	D	D	P	P	D	D
A ----> C	P	D	P	D	P	D	P	D
tranzitív								X
részleges				X				
kulcstörő							X	

A táblázatban három tulajdonság (A, B, C) három függése szerepel a feltételtörzsben úgy, hogy a harmadik függés tranzitívnak látszik. Azonban a feltételbejegyzések két értéke eloszlatja ezt a látszatot. A 'P' kulcsrészt, 'D' leírót jelent. A 'P' azt mutatja, hogy a meghatározó saját részeként, vagyis triviális függéssel határozza meg a függőt. A 'D' bejegyzés a leíró 'normális' függését jelzi.

A táblázat tevékenység-területe mutatja, hogy a három tulajdonság adott mintájú függéssora (oszlop) esetén milyen típusú függésről van szó. Ezt a területet részben üresen hagytam: csak az ismert eseteket jelöltem be. Vizsgálatomnak éppen az a célja, hogy a többi feltétel-együttest elemezzem és a következmények magyarázatát megadjam.

Az oszlopokat mondanivalómnak megfelelő sorrendben tárgyalom az alábbi pontban. Előbb azonban az ismert eseteket mutatom be.

8.oszlop: A **tranzitív függés** klasszikus esete.

$A \rightarrow B \rightarrow C$

Rendelésszám $\rightarrow$ Vevőkód $\rightarrow$ Vevőnév

4.oszlop: A **részleges függés** felismerhető esete.

$A\{X+Y\} \rightarrow B\{X\} \rightarrow C$

Tkód{Rszám+Cszám} $\rightarrow$ Cszám $\rightarrow$ Cikknév

7.oszlop: A **kulcstörő függés** alapvető esete.

$A\{X+Y\} \rightarrow B \rightarrow C\{X\}$

Utcaid{Kerületszám+Utcaszám} $\rightarrow$ Irányítószám

Irányítószám $\rightarrow$ Kerületszám

4.1.2 SZOKATLAN FÜGGÉSKOMBINÁCIÓK

A fenti táblázatból levezethető tehát az 1-3NF és a BCNF. A táblázat előállításának idején azonban az irodalom nem foglalkozott a másik öt esettel. Ezek közül kettő egyszerű.

1.oszlop: Nevezzük ezt **triviális hierarchiának**.

$A\{X+Y+Z\} \rightarrow B\{X+Y\} \rightarrow C\{X\}$

Szállításid{Cszám+Szszám+Pszám} $\rightarrow$ Cikkszál-
lítóid{Cszám+Szszám} $\rightarrow$ Cszám

Ez az eset azért érdekes, mert mindhárom függés triviális lévén a tranzitívitas nem küszöbölhető ki. Erre a tervezési segéd-eszközzel is fel kell készülni! Külön kérdés az, hogy vajon az A és a C által azonosított relációk összekapcsolandók-e vagy sem (ld. a 4.2 pontot!).

2.oszlop: **Kizárt**. Ha az $A \rightarrow B$ és a $B \rightarrow C$ függés triviális, akkor az $A \rightarrow C$ is csak az lehet.

A többi három eset komoly elvi problémákat vet fel: megkérdőjelezi a normalizálás tisztán matematikai jellegét.

3.oszlop: Ezt az esetet **hierarchikus kulcsnak** hívom, mert az összetett kulcs egyik részétől függ a másik.

$$A\{X+Y\} \text{ ----> } B\{X\} \text{ ----> } C\{Y\}$$

$$\text{Tételid}\{Rszám+Cszám\} \text{ ----> } Rszám \text{ ----> } Cszám$$

Egyes szakemberek erre azt mondhatnák, hogy ez a függéssorozat ellentmondásos, nem fordulhat elő. Modellezési gyakorlatomban konkrétan felbukkant ez a probléma. Mindig is meg fog jelenni akkor, ha többen időben és térben elhatároltan dolgoznak egyazon modellen.

A sorozat ellentmondásos, de előfordulhat. Oka többnyire az, hogy homonim nevet használnak. Itt: Rszám-nak nevezték a többtétéles és az egytétéles rendelés kulcsát egyaránt. Az utóbbi esetben a Cszám valóban függ az Rszámtól, míg az előbbi esetben nem.

Mások úgy vélhetnék, hogy ez a kulcstörő függés sajátos esete. Tőlük azt kérdezném: miért nem a részlegesé? Ez az eset a nevezett kettő kombinációja. Mindkettőtől eltér annyiban, hogy normalizálással nem oldható fel. A normálforma probléma hívja fel a figyelmet egy tervezési hibára, amit szemantikai módon küszöbölünk ki (új nevet adunk az egyik Rszám-nak).

5.oszlop: Ezt az esetet **metszet**nek neveztem, mivel két összetett kulcsnak van közös része.

$$A\{X+Y\} \text{ ----> } B\{X+Z\} \text{ ----> } C\{X\}$$

$$Dtételid\{Cszám+Dszám\} \text{ ----> } Rtételid\{Cszám+Rszám\}$$

$$Rtételid\{Cszám+Rszám\} \text{ ----> } Cszám$$

Az ilyen teljesen 'tudathasadásos' eset a gyakorlatban sokszor előfordulhat. Példánkban a DISZPOTÉTEL kulcsa meghatározza a RENDELÉSTÉTELét. Az eset skizofrén, mivel hasonlít a kulcstörésre, adott esetben a **pszeudo-tranzitívításra** (ld. 4.1.3) is, de nem oldható fel azok mintájára. A problémát ugyanaz az összefüggés okozza, mint a hierarchikus kulcs esetében. Nevezetesen az, hogy a harmadik, tranzitívnak látszó függés triviális, tehát nem szüntethető meg. Itt én két alesetet vélek felfedezni. Megértésükhöz azonban ki kell lépni a függéshármasból.

Ha a rendelések és a diszpozíciók viszonya hierarchikus, akkor fennáll az $Y \text{ ----> } Z$ függés (Dszám ----> Rszám). Az $A\{X+Y\} \text{ ----> } B\{X+Z\}$ kiküszöbölhető, mert vonatkozik rá az additívítási szabály [I21].

Ha viszont a rendelkezések és a diszpozíciók hálós viszonyúak, viszont egy DTÉTEL mindig csak egy RTÉTELhez tartozik, úgy $Y \rightarrow Z$, de $A\{X+Y\} \rightarrow B\{X+Z\}$ és a három függés közül egyiket sem lehet megszüntetni, vagyis nincs normalizálás.

6.oszlop: Ezt az esetet **csoportfüggés**nek hívom, mert az összetett tétel része külön is függ a kulcstól.

$A \rightarrow B\{X+Y\} \rightarrow C\{X\}$
 Balesetid $\rightarrow$ Bdátum $\rightarrow$ Hónap

A relációs modell elvileg atomi fogalmakkal dolgozik. Azonban elméletileg nem kellőképpen tisztázott - szerintem - az, hogy mit jelent az 'atomis' fogalom. Ismét szemantikáról lévén szó, véleményem szerint ez a tisztázás nem is a relációs koncepció feladata. Tény az, hogy az irodalomban igen sok a nem-csoportos adatként tükrözött(!), de valójában összetettnek tekintendő fogalom (név, telefonszám, dátum, cím; továbbá különböző - adott esetben nem elemi - kódok stb.). Ezért azután a gyakorlatban előfordulhat, hogy egy relációban rögzítik az összetett fogalmat is és annak (elemi) részét is. Ha valós részről van szó (a Hónap tényleg a baleset hónapja), akkor nincs teendő. A csoportfüggés ui. triviális. Egyébként viszont a megoldás ismét szemantikai; megfelelő elnevezést kell találni. A harmadik függés ekkor nem-valós.

Ezt a pontot a következőkkel zárhatom le:

- A hagyományos normalizálási koncepció nem foglalkozik a triviális hierarchia kezelésével. A csoportok tárgyalásánál (4.2) rámutatok ennek a következményeire. Nem tud mit kezdeni a hierarchikus kulccsal, a metszettel és a csoportfüggéssel. Ezeket a helyzeteket BCNF vagy egyéb problémaként félreértelmezi; lehetetlennek, illetve triviálisnak tételezi fel.

- Tehát a normálformák sora belülről hiányos. Az elképzelhető hét függési eset közül csak hármat fed le megnyugtató módon. Ezért a hagyományos normalizálási eljárások maguk is tökéletlenek, mivel csak látszólag alkalmazhatók a négy sajátos esetre.

Az elméleti szakemberek egy része a fenti elemzésen megütközik mondván: a normálformák arra valók, hogy a relációkat kivétítéssel

ésszerűen lebontsuk, feltételezve, hogy az adott függések megállják a helyüket.

A szakirodalomban mindig olyan példákat mutatnak be, amelyek elvileg kizárják azt, hogy a függésmegadás téves is lehet. Nekem, mint gyakorlat-elméleti szakembernek más a megközelítési mód. Feltételezem, hogy a tervezők - optimumra törekedve is - megadhatnak olyan függéseket, amelyek önmagukban helyeseknek látszanak, együttesük azonban helytelen. Ezek elméleti és gyakorlati kiszűrése a feladatom.

4.1.3 EGYÉB NORMÁLFORMA MEGFONTOLÁSOK

A fentiekben bizonyítottam, hogy a hagyományos normálformák sora belülről nem teljes. Abban az értelemben nem az, hogy nem ad kellő eligazítást az optimális adatszerkezet meghatározására. Pedig csak három attribútum funkcionális függéseit vizsgáltam és a többértékű függésről még szó se volt!

Az alábbiakban gondolat kísérletemet kiterjesztem háromnál több attribútum funkcionális függési viszonyaira. Megvizsgálom néhány sajátos esetet. Az alpontot egy, a normalizálási módszerre (4.3 pont) átvetető következtetéssel zárom.

Összetett tranzitívitas. Kezdetben a normálforma elemzés ún. dekompozíciós (kivetítések) eljárása jelentette 'a' normalizálást. Az ilyen elemzés tárgya az individuálisan, önmagában szemlélt reláció volt. E korai koncepció korlátai ma már világosak és ezzel a generikus példával szemléltethetők:

P402 Ra (p:A, f:B, E)
 Rb (p:B, f:C)
 Rc (p:C, f:D)
 Rd (p:D, E)

A ----> B, E és B ----> C ----> D ----> E

Tételezzük fel, hogy A, B, C és D - esetlegesen E is - eltérő relációk kulcsai úgy, hogy az Rx reláció az X kulcsot és a tőle közvetlenül függő tulajdonságokat tartalmazza csak!

A sorozat egy általános, nemcsak két relációt érintő, vagyis összetett tranzitivitást jelez. Hagyományos lebontása, amely szerint az Ra-ból kiemelt E tulajdonság előbb az Rb, majd az Rc, végül az utolsó, Rd relációba vándorol, nemcsak hosszadalmas. Az összefüggés összetett és a tervező esetleg fel sem fedezi az Ra relációban, hogy az A ----> E függés tranzitív.

Az összetett tranzitivitas három elméleti problémát vet fel. Az első az, hogy amennyiben a tervező nem ismeri fel közvetlenül az A ----> E függést, úgy egy segédeszköz miképpen támogathatja a tranzitivitas felszámolásában. E téren kísérlet történt az ún. univerzális reláció fogalmának a bevezetésére [I25]. Már többen utaltak arra, hogy ennek a felállítása és lebontása valódi képtelenség [I26]. A témára a 4.3 pontban térek vissza. Itt annyit jegyzek meg, hogy a SYDES egyetlen univerzális reláció helyett az alulról felépíthető összes ösvényt elemzi. Erről is a 4.3 pontban lesz szó.

A második elvi problémát még senki sem vizsgálta részleteiben és erre én sem teszek itt kísérletet. Az előző alpontban láthattuk, hogy hét olyan esettel kell számolnunk, amikor a három attribútum között meghatározott függés tranzitívnek **látszik**. A hét közül egyetlen esetben 'tisztá' a tranzitivitas. Pedig annak jellegétől függ a megoldás.

A kérdés tehát az, hogy amennyiben egy összetett függéssorban (P402) (triviális) és leíró (normális) függések egyaránt előfordulnak, úgy a végső, kedvezőtlen A ----> E függést milyen módon kell megítélnünk és feloldanunk. A SYDES - más alap hiányában - az (A,B), (D,E) és (A,E) függések alapján következtet.

Ciklusok. A harmadik elméleti kérdés arra vonatkozik, hogy az NF-elemzések feltételezik a ciklusok előzetes kiszűrését, különben végre sem hajthatók. Konceptiómban függési ciklust akkor említek, ha egy attribútum közvetve meghatározza önmagát. A ciklust egyszerűnek nevezem **kölcsönös** függés esetén. Egyébként a ciklus összetett. Az előbbire példa az alábbi:

P403 VÁLLALAT (p:Vállalatid, Vnév, ..., f:Egyszámlasz)
EGYSZÁMLA (p:Egyszámlasz, Összeg, ..., f:Vállalatid)

Valamikor a magyar vállalatok egyetlen számlaszámmal rendelkeztek a Magyar Nemzeti Banknál. Így ténylegesen létezett a

Vállalatid <--->Egyszámlasz kölcsönös függés. Hamis nyomot követ az, aki csak kulcsjelölteknek tekinti a jelzett attribútumokat és össze kívánja vonni a P403 két relációját. Így ui. szemantikai szegénységet vezet be (nem beszélve az opcionálitásról, ld. a 4.4 pontot!). Ugyanakkor a P403 változatlanul hagyása lehetetlenné teszi a normálforma elemzést, hiszen a két reláció minden leíró tulajdonsága (Vnév, Összeg) a kulcsokon keresztül tranzitívnak fog tetszeni. Az ilyen helyzetet automatikus segédeszköz magától nem oldhatja fel.

Az efféle kölcsönös függést a SYDES eleve kizárja és megköveteli, hogy szemantikus(!) megfontolások alapján azt mesterségesen egyirányúvá tegyék. A P403-ban egyértelmű, hogy előbb van vállalat, csak azután egyszámla. Ezért a VÁLLALAT relációból fogjuk kiemelni az Egyszámlasz tulajdonságot, mert úgy tükrözhető a vállalat-egyszámla viszony mesterséges hierarchiaként. A ciklusok hármas és többes összetételű eseteit egyébként is csak szemantikai úton tudjuk feloldani:

P404 Helységid ----> Járásid ----> Megyeid ----> Helységid

Ebben a példában az utolsó függés rosszul megadott, mert valójában kölcsönös: a székhelyre kíván utalni. (A problémát a SYDES-ban virtuális tranzitivitással oldjuk fel, de ezt nem mutatjuk be itt.)

Valójában az a feladat, hogy a normálforma elemzések előtt megszüntessük az alább általános módon kifejezett ciklusokat, különben a normalizálás lehetetlen lesz:

A ----> B -...-> D ----> E ----> A

A ciklus tipikus szemantikus kötöttségű konstrukció. Normálforma-elemzéssel kimutathatjuk, de feloldása csak (újra)értelmezéssel történhet, mint a P404 példában. N.B.: A SYDES-szoftver a P403-féle elemi ciklust eleve kizárja. A P404-féle összetett ciklusokat pedig tetszőleges összetételen keresztül is felismeri. Ehhez ösvényelemzést használ csakúgy, mint az összetett tranzitivitásnál (P402).

Gyémántalakzat. Mivel ezt a kifejezést magam vezettem be, ezért az különös magyarázatra szorul. Szemléltetésére egy hétköznapi példát alkalmazok:

P405 VEVŐ (p:Vevőkód, ...)
 SZERZŐDÉS (p:Szszám, ..., f:Vevőkód)
 TELEPHELY (p:Telepid, ..., f:Vevőkód)
 RENDELÉS (p:Rszám, ..., f:Szszám, f:Telepid)

A szerződések és telephelyek között nincs viszony. Gondolkodjunk egyedekben, mint fogalmakban! A rendelés alárendeltként egyaránt tartozik a telephelyhez és a szerződéshez, amik a vevő alárendeltjei. A felfelé való elágazás, majd összezáródás adja a 'gyémánt' alakot.

Ez az alakzat egyes relációs dogmatikusok számára, akik univerzális relációban és ún. minimális **lefedésben** (cover -[127]) gondolkodnak, elképzelhetetlen, mert az Rszám közvetve kétféle úton is meghatározza a Vevőkódot. Velük szemben én a P405 megoldását korrektnek tartom, mert szemantikai szempontból helyesen tükrözi a valóságot.

Pszeudo-tranzitívitas. A fent bemutatott döntési táblázat az általános tranzitívitas olyan különböző eseteit vázolta fel, amelyekben a függések tisztán, vagyis járulékos attribútumok nélkül mutatkoztak.

Velük szemben gyakori a pszeudo-tranzitívitas is, amelynek lényegét a következő példa szemlélteti:

P406 DISZPO (p:Dszám, ..., f:Rszám)
 RTÉTEL (p:{f:Rszám+f:Cszám}, ..., Rmennyiség)
 DTÉTEL (p:{f:Dszám+f:Cszám}, ..., Rmennyiség)

$A \rightarrow B$ és $X\{B+C\} \rightarrow D$ és $Y\{A+C\} \rightarrow D$

Nyilvánvaló, hogy a diszpotétel relációban a rendelt mennyiség a rendeltétellel szemben redundáns és kiküszöbölendő. Nem tiszta tranzitívitasról van szó. Külön vizsgálatot érdemelne a jövőben az, hogy a döntési táblázat egyes esetei miként alakulnának, ha az alapképlet pszeudo-tranzitívitas és nem egyszerű tranzitívitas lenne.

4.1.4 NORMÁLFORMA-TELJESSÉG ÉS A SYDES

Egyelőre nem szóltunk a 4-5NF alakokat megalapozó függésekről. Mégis máris kiderült, hogy az alapvető (1-3NF, BCNF) normálformák sora belülről nem teljes (4.1.1). Léteznek külön, xNF módon nem jelölt

összefüggések, amelyek hagyományos normalizációs elveken nem vezethetők le (4.1.2). A helyzetet tovább bonyolítja, hogy a függések több reláción is végigvezetendő függéssorozatokként értelmezendők (4.1.3) és az összetett tranzitivitás és ciklus, a gyémántforma és a pszeudo-tranzitivitás elméleti és gyakorlati problémái még nem teljesen feltártak. Az univerzális reláció és a lefedés konstrukciói csak árutakat jelentettek ezen a téren.

A SYDES szoftver feltárja a relációkhoz kötött attribútumok összes lehetséges normálforma hibáját úgy, hogy a természetes esetet (triviális hierarchia) nem tekinti bajnak. A SYDES az ismertett döntési tábla (4.1.1) alapján dolgozik. Szemben az általam ismert normalizálási segédeszközökkel és a SYDES elődeivel, ez a szoftver nem törli automatikusan a *matematikailag* rossznak hitt(!) függést.

Úgy vélem, hogy a 4.1.2 kellőképpen bizonyította: a normálforma-hiba csak jelzi a lehetséges problémát. A megoldás lehet szemantikai. A SYDES tetszőleges hálóban felfedezi az összetett függésproblémákat (4.1.3), de nem univerzális reláció alapon, hanem a relációk közötti függések analízisével. Ennek lényegét a 4.3 pont fogja felvázolni.

4.2 TULAJDONSÁGCSOPORTOK AZ ADATMODELLBEN

A relációs modell elvileg megengedi azt, hogy egy relációban az elsődleges kulcs elemi attribútumokból összetett legyen. Elméletileg nem zárja ki, hogy egy függésben - tekintettel az additivitási szabályra - ne csak a meghatározó, hanem a meghatározott attribútum is összetett lehessen. Azonban gyakorlatilag ezeket az összetételeket nem lehet külön szerkezeti tényezőkként meghatározni. Magyarul: a csoportokra nem lehet külön névvel hivatkozni. Ha ezt megengednék, úgy a táblázat nem felelne meg a reláció egzakt fogalmának.

Ebben a pontban nem az ismétlődéssel foglalkozom, hanem a tulajdonságcsoportok - kulcsok és leírók - általános modellezési problémáival. Be fogom bizonyítani az alábbi tételt:

T42 A relációs adatmodell nélkülözi a csoport struktúrális fogalmát. Emiatt a relációsan meghatározott adatmodellszerkezet, adott esetben,

hiányos vagy inkonzisztens is lehet. A csoport hiánya megnehezíti az NF-elemzést.

A csoport tárgyalásának a szükségességéhez a következő felismerés vezetett: Mivel az atomi attribútum fogalma nem egzakt, a relációs tervező így kezelheti pl. a Dátumot. Ennek az az ára, hogy a tételt procedurálisan kell lebontania pl. az ellenőrzéshez, és az nem kapcsolható az Évhez, ami a saját része. Megfordítva: kényelmetlen, ha a tulajdonságot mindig DÁTUM{Év, Hó, Nap} alakban kell megadni.

4.2.1 CSOPORTOS AZONOSÍTÓ-TULAJDONSÁGOK

A 4.1.2 alpontban sajátos függési esetként vázoltam fel az ún. **triviális hierarchiát**. Szemléltesse ezt most egy példa:

P407 SZÁLLÍTÓ (p:Szszám, ...)
SZÁLLÍTOTT-CIKK (p:{f:Szszám+f:Cszám}, ...)
SZÁLLÍTÁS-PROJEKTRÉ (p:{f:Szszám+f:Cszám+f:Pszám}, ...)

Mivel a példában minden elemi attribútum idegen kulcs, rájuk vonatkoznak a hivatkozási integritás szabályai. Ennek megfelelően a harmadik relációba csak létező szállító-, cikk- és projektszám hivatkozás illeszthető. A példa kapcsán többféle relációs elemzési nehézségre kell felhívnom a figyelmet:

- A harmadik relációnál a **hivatkozási korlátok** nem elegendők. Ide csak olyan {Szszám+Cszám} páros illeszthető, ami a második relációban létezik. E lenne a helyes jelölés: f:{Szszám+Cszám}. Itt nem érvényes az Armstrong-szabály (ld. 4.6 pont)!
- Sajnálatos, de az előző feltétel nem fogalmazható meg hivatkozási korlátként, mivel csoportot nem lehet kijelölni. A 'referential integrity' alapszabályai itt nem érvényesíthetők [I24, 89.o.]. A hivatkozási integritást ui. elsődleges doméjnen definiálták úgy, hogy az **elsődleges** doméjn egyetlen-attribútumú!
- Ha ezt a korlátot megadhatnánk, akkor viszont **korlát-szintű redundancia** (korlát-tranzitívitas) mutatkozna. Hiszen ha a

harmadik reláció kulcsa hivatkozik a másodikéra, akkor redundáns hivatkoznia az elsőre is.

- Tegyük fel, hogy a második és a harmadik reláció is tartalmaz egy X leíró tulajdonságot! Ez nyilvánvalóan részleges függést okoz a harmadik relációban. Amit aligha fedezünk fel, mivel nem tudjuk, hogy ennek a relációnak a kulcsa meghatározza a másodikét.

Az utóbbi problémát általánosan is megfogalmazhatjuk. A csoportok meghatározásának a hiányában a függési viszonyok elemzése és normalizálása kétségesse válik. Gondoljunk pl. egy olyan hierarchikus kulcsra, ahol $p:\{A+B+C+D\}$ és létezik az $\{A+B\} \text{ ----> } \{C+D\}$ függés! Vagy olyan pszeudo-tranzitivitásra, ahol $\{A+B+C\} \text{ ----> } D$, $\{A+E+F\} \text{ ----> } D$, és amellet $\{B+C\} \text{ ----> } \{E+F\}$. Aligha találjuk meg ezeket a függéseket!

A SYDES-módszer a modellezésben és a támogató normalizálásban vitathatatlanul előremutatóbb megoldást alkalmaz. Nem hűnyva szemet a valós összefüggések felett, a SYDES-ban a tervező meghatározhat csoportokat is. A P407 példa a pontosabb SYDES megfogalmazásban így néz ki:

P408 SZÁLLÍTÓ (p:Szszám, ...)
SZÁLLÍTOTT-CIKK (p:Szcikkid{f:Szszám+f:Cszám}, ...)
SZÁLLÍTÁS-PROJEKTRE (p:Szprojid{f:Szszám+f:Cszám+f:Pszám}, ...)

A SYDES-ban az összetett kulcsok közötti függéseket is egyszerűen lehet megadni. Így kijelölhető ez a függés: Szprojid ----> Szcikkid.

Mindez az alábbi előnyökkel jár:

- Vizsgálható, hogy egyedi-e az összetétel. Ha létezik egy $R_x(p:\{A+B\})$ reláció, akkor a SYDES nem engedi meg, hogy létezzen az $R_y(p:\{A+B\})$ vagy az $R_z(p:\{B+A\})$.
- Vizsgálható, hogy az összetett kulcs összetett részei maguk is kulcsok-e, ergo hivatkozási integritást jelentő idegen-kulcsok. Az $R_x(p:\{A+B\})$ reláció kulcsa része az $R_y(p:\{A+B+C\})$ kulcsának.

- Megfordítva: elemezhető, hogy egy összetett kulcs része-e egy másik összetételnek. Az is elemezhető, hogy közvetlen, nem-tranzitív részről van-e szó.

A SYDES a P407/P408 esetén az első és a harmadik reláció között nem keres hivatkozási összefüggést, mert annak alapjait megtalálja az első és a második, ill. a második és a harmadik közötti hivatkozásban. Amint látható, a SYDES kiküszöböli a fentebb felsorolt, a csoportok hiányából fakadó elemzési problémák jelentős részét.

A feltételezett X leíró tulajdonság részleges függését pl. a következő, relációsan nem megengedett sorozattal tárjuk fel:

Szprojid ----> Szcikkid (triviális, rész)
 Szcikkid ----> X
 Szprojid ----> X

4.2.2 CSOPORTOS LEÍRÓ-TULAJDONSÁGOK

Az alapvető probléma alig különbözik az előző alpontban felvázolttól. A hivatkozási integritás lehetőségét és redundanciamentességét vizsgálom. Most úgy, hogy a **külső-kulcs** leíró tulajdonság és nem kulcsrész, mint fent. Szemléltetésül a P406 példára hivatkozok vissza.

A DTÉTEL (p:{f:Dszám+f:Cszám}, ..., Rmennyiség) relációban leíró tulajdonságként kellene megadni az RTÉTEL reláció {Rszám+Cszám} kulcsát. Ugyanis a diszpotételek rendelőstételekre vonatkoznak, azok hierarchikus lebontásai. Tehát elvileg létezik a Dtételid ----> Rtételid függés. E példán ütköznek ki az ortodox relációs modellezés hibái:

- Az Rtételid{Rszám+Cszám} csoport nem adható meg a DTÉTEL relációban a rendelőstétel hivatkozásaként, mert csoportot nem lehet megnevezni. Pedig evidens, hogy ez lenne az egyetlen természetes nem-redundáns, nem-tranzitív hivatkozás. Ui. nem bármelyik rendelés akármelyik cikkéről van szó, hanem egy meghatározott párosról, ami rendelőstétel azonosító!
- Külön-külön tulajdonságként még annyira sem vihető a DTÉTEL relációba az Rszám és a Cszám tulajdonság. Egy relációban nem fordulhat elő kétszer ugyanaz az attribútum (Cszám). Ha fennáll

a Dszám ----> Rszám függés, akkor az Rszám részlegesen függ az összetett kulcstól. Ezen kívül nem a megfelelő korlát vonatkozik rá, mert csak egyedi attribútumként validálható.

- Mellékes, de nem lényegtelen hatás az, hogy nehezen lehet feloldani a pszeudo-tranzitivitást, mert nincsenek felismerhető, felbontható és átfedő csoportok. A pszeudo-tranzitivitás pedig a csoportátfedésen - közös rész - alapul (vö. 4.1.3).
- A pszeudo-tranzitivitás a 4.1.2 alpontban ismertetett metszet sajátos esete, amikor is két összetett kulcs nem-közös részei között van függés. Ámde sem a metszet-, sem a csoportfüggés megoldására nem kapunk eligazítást a relációs koncepcióban.

A fenti második megjegyzésre - kétszer nem vihető ua. a tulajdonság egy relációba - egyesek azt mondhatnák, hogy ez természetes. Számukra készült az alábbi, életből vett példa:

P409 MUNKAFAZIS (p:Fszám, ..., f:Projektkód, f:Erőforráskód)

Mindkét idegen kulcs összetett: Projektkód{Szervezet+Pszám} és Erőforráskód{Szervezet+Eszám}. Feltételezésünk szerint az erőforrást nem mindig a projektért felelős szervezet adja.

Itt a relációs koncepció csapdába esik. Ha ezekkel az összetételekkel hivatkozik, akkor procedurálisan kell kikeresni a szervezetet. Esetleg kedvezőtlen függés (pl. pszeudo-tranzitivitás) marad feltáratlan. Ha lebontja az összetételeket, akkor nem tud két szervezetet megadni, ill. elveszíti a hivatkozást az összetett kulcsú projekt és erőforrás felé.

4.2.3 CSOPORTOK MODELLEZÉSE A SYDES-BAN

A SYDES-módszer előnyei a relációs modellezéssel szemben világosak. A diszpozíciós példa (P406) megoldása egyszerű. A DTÉTEL relációba csoportként beillesztjük az RTÉTEL Rtételid{Rszám+Cszám} összetett kulcsát. Így megszűnik a hivatkozási integritás problémája. Mi több, a pszeudotranzitíven függő Rmennyiség attribútum ezt követően már egyszerűen tranzitívként tárható fel.

A csoportként való meghatározás szabályai a SYDES-ban szigorúak. A csoportosnak szándékozott tulajdonságokhoz előre ki kell jelölni a várt tényezők számát. Attribútum-struktúrában adjuk meg, hogy egy X tulajdonság az A{X+Y} csoport része. A csoport hálós szerkezet; azaz a csoport egy tagja más csoportban is lehet beépülő. Nem kizárt, hogy a csoport hierarchikusan több szintű legyen (lásd az alábbi korlátot!).

Kimondottan technikai és nem elvi okokból, a kulcsokat másként kezeljük a SYDES-ban, mint a leíró tulajdonságokat. A belső, hierarchikus felépítést csak azoknál a tulajdonságoknál engedjük meg, amelyek bizonyosan leírók maradnak. Nem lesznek elsődleges kulcsok és így idegen kulccsá sem válnak. Az elsődleges kulcs csoportként meghatározható, de részeinek a száma (8-ra) korlátozott és részei nem lehetnek csoportok.

Az összetett tulajdonság, mint csoport, külön attribútum. Ha kulcsként adjuk meg, akkor a SYDES ellenőrzi azt is (a tényezők számán és elemi voltán kívül), hogy minden tényezőjét megadták-e és az összetétel - sorrendtől függetlenül - egyedi-e. Az elemzés során a SYDES megállapítja, hogy egy kulcs bármelyik része maga is kulcs-e, illetve a kulcs része-e másik összetételnek. Ilyenkor automatikusan létrehozza a hivatkozási összefüggést, beillesztve a hivatkozott attribútumot a hivatkozó relációba, aminek a kulcsától függ.

A SYDES a leíró tulajdonságok részeit ténylegesen nem tölti be a relációhoz és így a redundanciát elkerüli. Az elemzés során azonban lebontja a csoportokat, hogy a tagok esetleges kedvezőtlen függéseit feltárja. A csoportot tehát **virtuális attribútumként** kezeli. A csoport létezik is, meg nem is.

Ezzel szemben a relációs koncepció mereven ragaszkodik az **egysíkú** struktúrához. Mivel fogalomrendszerünk eleve feltételezi a csoportokat mint önálló, esetleg több szinten át egymásbaágyazott konstrukciókat, a fenti elemzés helyett egyszerűen és jogosan kijelenthettem volna, hogy a relációs modell nem alkalmas a fogalmi modellezésre. Azonban nem az volt a célom, hogy a relációs koncepciót elmarasztaljam.

Arra szerettem volna rámutatni, hogy a normálforma elemzés, a kulcskijelölés, a hivatkozási integritás megadása, vagyis mindaz, amit a relációs adatmodellezés lényegének tekintünk, sokat veszít azáltal, hogy nélkülözi a csoport konstrukciót.

A SYDES léte és működési módja bizonyítja, hogy a csoport és a relációs gondolatok egymást kiegészítik és nem kizárják. A fogalmi modell elemzésében a csoport nélkülözhetetlen. Talán célszerű lenne a relációs modellt is ekképpen átalakítani! Ennél figyelmet érdemel a virtuális attribútum konstrukció.

4.3 NORMALIZÁLÁSI ELJÁRÁSOK ÉS A SYDES-KATEGÓRIÁK

Az optimális normálformák kialakítására két eljárást javasol a szakirodalom. A normálforma dekompozíciót [I20] és a normálforma szintézist [I28]. Ebben a pontban nagyító alá kerül a két algoritmus.

Bizonyítani fogom az alábbi tételt:

T43 Egy reális mértékben összetett adatmodell optimuma nem érhető el a nevezett algoritmusok egyikével sem. A két eljárás egyébként egymást nem kizárja, hanem feltételezi. Együttes alkalmazásukkal az optimum jobban közelíthető, mint a két egyedi módszer bármelyikével.

A 'reális mértékben összetett' jelzőt nem fogom pontosítani, viszont alkalmaznom kellett a szélsőségek elkerülésére. Két tulajdonságból lehet optimális modellt alkotni. Az irodalmi mintapéldákból is. A gyakorlat-elméletben azonban attribútumok (tíz)ezreiről van szó!

A pont további részeiben feltárok egy szemantikai problémát, ami az eltérő módszerhasználatból ered. Bemutatom a SYDES normalizálási algoritmusát. Végül utalok a SYDES-kategóriák relációs felfogására.

4.3.1 ADATMODELL DEKOMPOZÍCIÓ ÉS SZINTÉZIS

Az eredetileg javasolt **dekompozíciós** eljárásban a tervező először a célszerűnek tartott relációkat definiálja, kijelölve attribútumaikat. Majd a relációkat egyenként sorra veszi. Megvizsgálja a bennük lévő attribútumok különböző (funkcionális, többértékű) függéseit, az egyszerűeket és az összetetteket egyaránt. Kiszűri a kedvezőtlen

helyzeteket; a részleges, tranzitív, kulcstörő stb. függéseket úgy, hogy kivetíti a megfelelő tulajdonságo(ka)t külön reláció(k)ba.

Ha már létezik reláció a kivetítés alapjául szolgáló kulccsal, akkor az optimalitás jegyében [I19] az új reláció abba beleolvad. Az is előfordulhat, hogy a kivetítéssel új elem nem keletkezik, mert a kulcs is, a tőle függő tulajdonság is - párként - már megtalálható egy másik relációban. A P401 példa esetén létezhet már a CIKK (Cikkszám, Cikknev, ...) reláció. Így a RENDELÉSTÉTELből való kivetítés új elemet nem létesít.

Könnyű belátni, hogy egy ilyen módon optimalizált relációkból álló adatmodell-terv nem szükségszerűen optimális. Ezen megállapítás szemléltetésére elegendő egy hétköznapi példa:

P410 VEVŐ (p:Vevőkód, Vevőcím)
RENDELÉS (p:Rszám, Vevőnév)

A két reláció egyenként nem is lehet tökéletesebb normálformában. Együtt azonban nem optimálisak. Hiányzik ez a függés: Rszám ----> Vevőkód. Ennek alapján a Vevőkódot a RENDELÉSbe illesztve, a tranzitív Vevőnevet a VEVŐbe áthelyezve már optimumhoz jutunk. A hiányzó függést azonban a tiszta dekompozícióval, ami **egyedi** relációkra irányul, nem lehet felfedni.

A triviális kapcsolati hiányok elkerülésére használják a normálforma **szintézis** eljárását. Első modellezési eszközm, a SZIAM [H15, H16] is ezt alkalmazta. Az eljárás lényege a következő: Először meghatározzák az attribútumokat. Majd kijelölik az ezek közötti közvetlen - nem-részleges, nem-tranzitív stb. - függéseket. Ezután annyi relációt definiálnak, ahány meghatározó tulajdonságot találtak. A relációnak a meghatározó a kulcsa, a meghatározottak pedig a leíró tulajdonságai.

Az ilyen leegyszerűsített formalizmuson alapuló optimumkeresés egy valamennyire is összetett rendszer esetén nem célravezető. Ezer tulajdonság esetén félmillió(!) tulajdonságpár funkcionális függését kellene megvizsgálni, nem szólva az összetett tulajdonságokról és a többértékű függésről. A feladat gyakorlatilag kivitelezhetetlen.

Itt kell visszatérnünk az analízis módszerére. A kezdeti hiányosságot kiküszöbölendő (ti. azt, hogy függés maradt feltáratlan) vezették be az **univerzális reláció** [I25] fogalmát. Ez a reláció az alkalmazás összes attribútumát tartalmazza. A köztük lévő függések elemzése reményt

nyújthatna arra, hogy nem marad függés feltáratlan. A konstrukció létjogosultsága vitatható. Az analízis hívei átvették és módszerük lényegévé tették a szintézist, hiszen az dolgozott univerzális relációval. Csak nem így, hanem attribútumkészletnek nevezték a koncepciót.

Leszögezhetjük tehát, hogy a normálforma dekompozíció elvileg nem tökéletes módszer, mert figyelmen kívül hagy függéseket. A szintézis pedig gyakorlatilag kénytelen ugyanezt tenni.

A normálforma dekompozíció és szintézis algoritmusai nem vezet mindig ugyanarra az eredményre. Ezt az alábbi példa mutatja:

P411 VEVŐ (p:Vevőkód, ...)
TELEP (p:Telepkód, ..., f:Vevőkód)
RENDELÉS (p:Rszám, ..., f:Telepkód, f:Vevőkód)

Vállalatunk vevőitől úgy érkeznek hozzánk a rendelések, hogy azokat hol maga a vevő, hol annak egyik telepe adja fel.

A szintézist alkalmazó tervező ezt a megoldást nem találja meg; azt elvileg rossznak tartja. Ő a következő relációfüggetlen függésekben gondolkodik: Rszám ----> Telepkód ----> Vevőkód. Számára az Rszám ----> Vevőkód függés tranzitív, és ezért a RENDELÉS reláció csak második normálformájú.

Az analízist használó tervező reláción belül elemzi a függéseket. Számára a Telepkód csak a TELEP relációban határozza meg a Vevőkódot. A Telepkód ----> Vevőkód függés a RENDELÉS relációban nem áll fenn, mivel nem minden rendelésben van Telepkód érték. A megoldás tehát jó, nincs tranzitív függés.

A szintézis és a dekompozíció ellentmondása feloldhatatlan. Az előbbi szemantikai értelmezés hiányában visz tökéletlen megoldásra. Az utóbbinak meg a lényegében van a teljesség hiánya. A P411 példa azt is szemlélteti, hogy az univerzális reláció értelmetlen konstrukció, mert ugyanolyan szemantikai félreértelmezésre vezet, mint a szintézis.

4.3.2 A SYDES ELEMZÉSI MÓDSZERE

Az előző alpontban rámutattam, hogy a két alapvető normalizálási módszer egyike sem állja meg a helyét önmagában. Az analízist ki kell egészíteni a teljességet biztosító részmódszerrel. Ez nem lehet az univerzális reláció, mert ezen szintetikus konstrukció elemzése

gyakorlatilag lehetetlen. A szintézist meg kell fejteni szemantikai elemzéssel, különben nem-teljes eredményre vezethet olyankor, ha valamelyik jó függést formai alapon kedvezőtlennek ítéli.

A SYDES-koncepcióban a következő algoritmust javaslom:

- Először adjuk meg célszerűnek tartott relációinkat úgy, mint egyedeket, egyelőre attribútumok nélkül. A relációk ui. nemcsak a normalizálás tárgyai. Tervezési keretet is jelentenek, eljárásokhoz kapcsolódnak, rájuk események vonatkoznak stb.
- Jelöljük ki a relációk kulcsait és célszerűnek tartott leíró tulajdonságait. Az ismétlődést kerüljük el. Ha tudjuk, akkor azonnal szűrjük ki a tranzitív, részleges és egyéb függések kedvezőtlen jelenségeit.
- Vizsgáljuk meg - célszerűen egy mátrix segítségével - a kulcsokat párosával, hogy van-e köztük feltáratlan függés. Ha új függést fedezünk fel, akkor egészítsük ki a megfelelő relációt a meghatározott kulccsal.
- Most elemezzük újra az így kiegészített relációk normálformáját. Iteráljunk addig, ameddig az azonosítók között új függéseket fedezünk fel. A leírt eljárás tetszőleges pontjain alkalmazzuk a SYDES alábbi elemzőprogramjait a rossz függések kiszűrésére.

A SYDES algoritmus a heurisztikus. Nem ígéri az eleve elérhetetlen elméleti optimumot. Reális mértékben összetett modellezési feladatnál a tiszta analízisnél teljesebb, a tiszta szintézisnél szemantikailag helyesebb eredményt ad, mert algoritmusunk az analízis és a szintézis ötvöze. Az előbbinél elvileg tökéletesebb, az utóbbival szemben viszont gyakorlatilag végrehajtható. Ezt külön is magyarázom.

A SYDES-ban a leíró tulajdonságokat adott relációba soroljuk és a reláció első normalizálása után csak akkor foglalkozunk velük újra, ha a relációba új tulajdonság kerül, ami függési viszonyban állhat ezekkel a tulajdonságokkal. Vagy még akkor sem. Például: előre leszögezzük, hogy a Vevőnév csakis a Vevőkóddal azonosított VEVŐ

relációban lehet. Így a Vevőnév tulajdonság további függéseit nem kell ellenőriznünk.

Mivel alkalmazásfüggő, nem lehet megmondani, hogy egy elméletileg feltételezett(!) univerzális relációban mennyi a leíró tulajdonság és ezek közül hányat lehet triviálisan jól tervezettnek tekinteni. Saját tapasztalataim szerint az átlagos alkalmazásban az attribútumok 10-15%-a a kulcs és a leírók között még ennél is kisebb arányban vannak olyanok, amelyekről esetlegesen kiderülhet, hogy valójában kulcsok. A tipikusan leíró tulajdonságok modellezése tehát nem gond.

Ezért választottam azt a megoldást, hogy csak a kulcsokra végzem el a teljes elemzést, egy **univerzális kulcsrelációval**. Egy régebbi tanulmányban [H18] vizsgáltam azt, hogy milyen eséllyel fedezhetők fel a rejtett függések akkor, ha csak a kulcsok között tételezünk fel univerzális relációt, és csak azok függésvizsgálatait iteráljuk. A lehetséges kombinatorikus esetek 87%-ban a rejtett függés feltárul. Ez a százalék - durva becslés szerint - mutatja azt is, hogy elvileg milyen mértékben közelíthető meg az optimális adatmodell. Mert ugyan jóval több a leíró mint a kulcs, de ugyanakkor kicsi a valószínűsége annak, hogy köztük meglepő függésre bukkanunk.

A SYDES elemzőprogramjai az alábbi módon támogatják a fenti normálforma-elemzési módszert:

- Amikor a tervező megadja az egyedek attribútumait, a SYDES karbantartja azok **abszolút és relatív szerepeit**. A relatív szerep azt mutatja, hogy az attribútum a relációban kulcs, kulcsrész vagy egyszerű leíró, és ugyanakkor van-e idegen-kulcs feladata (kapcsoló tétel). Az abszolút szerep az attribútum legfontosabb relatív szerepével egyezik meg. Ha egy attribútum egy relációban kulcs, egy másikban pedig leíró, akkor abszolút szerepe kulcs. Az egyik relációban kulcs, a többiben pedig kapcsoló tétel a relatív szerepe.

- A SYDES először a kulcsok közötti függéseket vizsgálja. Az elemzés során ui. legelőször a **ciklusokat** szűri ki és ha talál ilyen, akkor az elemzést megszakítja. Majd a 4.1.1 alpontban megadott táblázat alapján a hat esetnek megfelelően kijelzi a **kedvezőtlen függéseket**.

- Ezután a SYDES összeállítja a függések **szuperhálóját**. Ezen olyan funkcionális függéssorozatokot kell érteni, amelyek kezdő tagjai a mások által meg nem határozott kulcsok. Felül ezek az **ösvények** egybefuthatnak. Ösvényeket említek, mivel ezek egyben a relációk (egyedek) közötti - mindig csak felmutató - navigációs utaknak is megfelelnek.

- A SYDES karbantartja a normálformákat. Majd **konnektivitás-elemzést** végez. Kijelzi az egymással kapcsolt relációk alhalmazait. Az így kiírt alhalmazok relációi között nincs semmiféle kulcskapcsolat (ld. a hivatkozási integritást).

- Ezután a leíró tulajdonságok normálforma hibáinak a feltárására kerül sor, szintén döntési táblázatunk alapján. Itt az elemzést már könnyíti az előbbieken felépített háló.

- A tervező az elemzést bármelyik ponton megszakíthatja. Megadhat új összefüggéseket és törölhet régieket. Esetleges iterációk után a szerepek alapján automatikusan generáltathatja a **kapcsolat-típusokat**, a relációk összefüggéseit.

Tudom, hogy ha a technikai adatok, illetve a feladat, vagyis a modell jellegzetességei nem ismertek, akkor az alábbi információk nem mondanak sokat. Mégis szeretném érzékelteni a SYDES képességeit. 300 egyedtípus 1100 doméjnen meghatározott 1800 attribútumát elemeztük. A kulcsok közti viszonyok (500) elemzése 17 percig tartott és 33 hibát mutatott ki. A leíró-viszonyok (1300) elemzése 22 perc alatt nem jelzett újabb hibát. (A gép: IBM AT, 512K, 20MB.)

4.3.3 NORMALIZÁLÁS ÉS A SYDES-KATEGÓRIÁK

Az előző fejezetben a SYDES adatmodelljét egyed-tulajdonság-kapcsolat (EAR) modellként mutattam be. Ebben a fejezetben viszont relációkról, doméjnekről, attribútumokról, normalizálásról, hivatkozás integritásról beszélek. A fejezet elején tételként jelentettem ki, hogy a relációs adattervezés nem vezet optimális adatmodellre, itt pedig

elárultam, hogy a SYDES is végez normalizálást. Nincsen mindebben logikai ellentmondás?

Nincs. Tételtem nem azt fejtegeti, hogy a normalizálás elvetendő, hanem azt, hogy tökéletesítendő. Bár a normalizálást szemantikai kötöttségűnek mondják a legtöbb műben (így az [I24]-ben is), valójában azt a túlzott formalizmus jellemzi. Nem vitás, hogy az EAR modellek szemantikailag gazdagabbak a relációsnál. A gyakorlat-elméleti szakember feladata tehát 'egyszerű': ötvöznie kell a kétféle modellt. Az egyikből át kell vennie az erőteljes formalizmust, a másiktól a kevésbé formális szemantikát. A SYDES-nak éppen az a lényege, hogy egyesíti a különböző oldalakat. Ezt normalizálási eljárása is mutatja, amely az 'ellentmondó' algoritmusokat ötvözi.

Azt, hogy a SYDES-ban miképpen értelmezzük és alkalmazzuk a normalizálást, a következő két pontban még bővebben is tárgyalom.

Itt be kell mutatnom, hogy a SYDES-kategóriák milyen módon felelnek meg a relációsaknak. Korábban nem lett volna célszerű erre a magyarázatra kitérnem, mivel a SYDES-kategóriák kapcsolódnak a függések és a normalizálás fogalmához.

A relációs modellben három alapkategória van: a doméjn, mint értékhalmoz; a relációba leképezett doméjn, az attribútum és a reláció, ami - durván fogalmazva - az attribútumok sora. A kategóriák formálisak. Ebben a struktúrában az egyetlen szemantikai lényeg az, hogy az egyazon doméjnből eltérő relációkba leképezett attribútumok nevei különbözőek lehetnek. A koncepcióban nincs külön kategória a tükrözendő jelenségre.

A SYDES modelljében az egyed kategória szolgál a jelenségek tükrözésére. Ennek megfeleltethető az attribútum nélküli reláció fogalma. Nincs itt ellentmondás: egy relációnak megadhatjuk a nevét, előforduláshalmazát (range), felhasználási módját stb. anélkül is, hogy ismertetnénk attribútumait.

A tulajdonság a SYDES-ban reláció nélküli attribútum. Nem teljesen azonos a doméjn kategóriájával. Nemcsak értékhalmoz, mivel van szemantikai tartalma is. A tervező a SYDES-ban nem alkalmazhat 'Név' vagy hasonló általános nevű tulajdonságot. Meg kell mondania, hogy 'személy' vagy 'vállalat' nevére van-e szó. Vagyis mindig áll egyed a tulajdonságok mögött. A relációba leképezett tulajdonságot, az attribútumot, egyed-tulajdonság-viszonynak nevezzük.

A SYDES-kategóriák és a normalizálás összefüggenek. Az optimális modell annyi egyedet ölel fel, ahány funkcionális meghatározó van. A tulajdonság minden olyan egyedben szerepel, amelynek kulcsa őt meghatározza. Két egyed között kapcsolattípus létezik, ha a kulcsaik között funkcionális függés van. A SYDES-ban a kapcsolat szerkezeti, névvel ellátott tényező és nem pusztán hivatkozás-integritási korlát.

4.4 NORMÁLFORMÁK ÉS EGYÉRTELMŰSÉG

A normalizálással foglalkozó szakirodalomban utalás történik arra, hogy a normálforma szemantikai fogalom [I24, 237.o.]. Azonban a szemléltetésül alkalmazott példákban szinte kivétel nélkül feltételezik, hogy az attribútumok és a relációk helyesen meghatározottak. Úgy tűnik, hogy a normalizálás egyetlen tévedhetetlen szakember feladata, akit a felhasználók értelmezési tekintetben helyesen eligazítottak.

A valóság viszont az, hogy az adatmodellezést nem egy szakember végzi. Az eltérő szemléletű felhasználók tucatjaival kommunikálva az attribútumok közötti összefüggéseket és így a relációkat is, térben is elkülönítve, időben széthúzva határozza meg több modellező. Ha külön-külön ki is zárnánk a tévedés lehetőségét (ezt sem tehetjük), akkor is számolnunk kell az elkülönültség miatti ellentmondásokkal.

A gyakorlat-elméleti szakembernek erre a helyzetre kell felkészülnie. A Bayerische Motoren Werke vállalatnál, ahol megvették az ÁDÁM & ÉVA segédeszközümet, más normalizátorokat is kipróbáltak. Őt attribútum és négy reláció egyszerű példájánál ezek az eszközök megbuktak, amikor tévesen adtak meg függéseket. Az én eszközüm kimutatta a szemantikai hibát.

T44 A normálformák nemcsak matematikai formai konstrukciók. A normalizálás jelentős mértékben hozzájárul az adatmodell szemantikai alapú tökéletesítéséhez is.

4.4.1 A NORMÁLFORMÁK ÉS ATTRIBÚTUM-HIVATKOZÁSOK

Az eddigiekben éles különbséget tettem az azonosítók és az egyszerű leíró tulajdonságok között. Ezt a megkülönböztetést célszerű

fenntartanom az alábbiakban is. A tulajdonsághivatkozások kétféle hibája esetén ui. a kulcsok és a leírók másképpen viselkednek. Ezért a következőkben külön tárgyalom a homonim és szinonim kulcsok, ill. leíró tulajdonságok esetét.

Homonim azonosítók. Tegyük fel, hogy a tervező azonos Rszám névvel illeti a VEVŐRENDELÉS és a SZÁLLÍTÓRENDELÉS kulcsát! Ezt a megoldást a SYDES kétszer utasítaná el. Egyazon névvel nem adható meg két eltérő tulajdonság. Két különböző egyednek meg nem lehet azonos (nevű) az elsődleges kulcsa. Ilyenkor célszerű bevezetni a Vrszám és az Szrszám kulcsneveket. Elvileg azonban nem kizárt, hogy az egyik relációban Rszám marad a kulcs neve. És éppen ez okozhat további gondokat.

A tervezők hajlamosak elfeledni, hogy átneveztek fogalmakat. Az Szrszám minősített név bevezetése ellenére a leíró tulajdonságok megadásánál a Szállítókódot a VEVŐRENDELÉS Rszám kulcsához kötik, tévedésből. Szintézisnél igen gyakori a homonimák ilyen jellegű 'visszacsempészése'. Ha a szintézist dekompozícióval párosítjuk, akkor a VEVŐRENDELÉS relációjának a vizsgálatokor fény derül a hibára, hiszen oda nem illő attribútumra fogunk bukkanni. A tévedések egy része automatával is kimutatható, mert pl. tranzitivitást okoz.

A homonimákat ab ovo el akarom kerülni. Ehhez két résztechnikát alkalmazok. A félreérthető megnevezéseknél eleve bevezetem a minősítést (Vevőrszám, és nem csak Rszám). A tulajdonságokat (is) minősítő kódokkal osztályozom. Így a Nem/kód és a Nem/karaktersor tulajdonságok elkülöníthetők. Arra vigyázni kell, hogy ezt a minősítést ne tévesszük össze a szerepneveket és altípusokat megalapozó minősítéssel! A Vevőrszám és a Szállítórszám nem egyazon általános Rszám szerepnevei, hiszen értékeik nem egy Rszám értékthalmazának a tényleges alhalmazai.

A homonim kulcsokat azért is kerülni kell, mert hamis képet mutatnak a kapcsolhatóság szempontjából. Az alábbi példa első relációjában a Tulajdonoskód azonosító és ezért minden tulajdonosnál más értékű. A másik relációban kétértékű és magán, ill. állami tulajdonost mutat. A két reláció tehát csak látszólag kapcsolható!

P412 TULAJDONOS (p:Tulajdonoskód, ...)
GÉPKOCSI (p:Rendszám, ..., f(?):Tulajdonoskód)

Szinonim azonosítók. Igen gyakran előfordul, hogy a tervezők ugyanazt az attribútumot, feledékenységből vagy lazaságból, hol így, hol úgy nevezik. Igen sokszor nem tényleges szinonimáról van szó, hanem csak írásmódbeli különbségről. Éppen elég egy automata megzavarására, ha felváltva alkalmazzuk a Cikk-Típuskód, a Cikk-Típus-Kód, vagy a Cikktípus-Kód nevet. Vagy hol Vevőkódnak, hol Vevőkulcsnak nevezzük a VEVŐ egyed azonosító tulajdonságát. A virtuális szinonimák éppen olyan gondokat okoznak az optimalitás elemzésében, mint a valósak.

A virtuális szinonimák nagyrésze elkerülhető, ha betartjuk saját anyanyelvünk írási szabályait és egyezményeket vezetünk be az írásmódra (rövidítés, tagok elválasztása, kulcs hivatkozása stb.). A nevek karaktersorok alapján történő vizsgálata és a minősítőkódok szerint történő szelektív lekérdezés is segíti a szinonimák feltárását.

A szinonim azonosítók azonban egyéb módokon is kimutathatók. Meg kell próbálni betartani a **teljességszabályt**. Ennek értelmében ha egy tulajdonság függ egy kulcstól, úgy a kulcs által azonosított relációban van a helye. Ha a tervező Vevőkód és Vevőid kulcsokkal (szinonimák) két relációt tervez, akkor mindkettőben lesz Vevőnév. (Ha a két reláció neve azonos, akkor a szinonima azonnal feltárul.)

Ha a leírók nem árulják el a szinonim kulcsot, akkor is azt teszik a kapcsoló tulajdonságok. A vevőrendelésben utalunk a vevő kulcsára. Ha a VRENDELÉS relációban a két szinonima - Vevőkód és Vevőid - együtt szerepel, akkor a dekompozíciós (relációnkénti) vizsgálatnál ez kiderül. Ha valamelyik itt nem fordul elő, akkor viszont éppen a kapcsolat hiánya fogja jelezni a problémát (inkonnektivitás).

Végül vissza kell utalnom arra, hogy a SYDES-ban külön vizsgáljuk a kulcsok közötti függéseket. Legkésőbb ezen a ponton fog kiderülni, hogy a Vevőkód és a Vevőid által azonosított, eltérő nevű(!) relációk valójában azonos jelenséget takarnak, mert elsődleges kulcsaik között kölcsönös függés fedezhető fel, ami nem-valós; szinonimán alapul.

Homonim leírók. Ez a jelenség igen gyakran előfordul a nem kellő minősítés miatt. 'Mennyiség'. Ilyen nevű tulajdonság akár több tucat relációban is felléphet. Az efféle homonimát azért kell kiküszöbölni, mert rontja az egyértelműséget. Itt mutatkozik meg a relációs és a SYDES gondolkodás egyik alapvető különbsége. A relációs modell

sokszor normálforma-hibát említ ott, ahol a SYDES egyértelműség-hiányt mutat ki. Példa:

P413 CIKK (p: Cikkszám, ..., Ár)
RENDELÉSTÉTEL (p: Tételid{Rszám+f:Cikkszám}, ..., Ár)

A relációs tervező annyit lát, hogy a második reláció csak első normálformájú a részleges függés miatt. Ez tévedés. A normálforma-hiba itt egy szemantikai problémára, egy homonimára hívja fel a figyelmet. Egészen más fogalom a cikkhez mindig tapadó 'egységár' és az adott rendeléstől függő, az előzőtől esetlegesen eltérő 'szerződéses' vagy 'rendelési' ár. Vagyis nem pusztán a normálforma hibát, a jelzést, kell felszámolni, hanem a szemantikai problémát kell kiküszöbölni.

A SYDES elemző-programjai a feltárt normálforma hibákat nem szüntetik meg automatikusan (megtehetnék), hanem csak jelzik őket. Tapasztalataim szerint ugyanis a normálforma hiba az esetek több mint felében nem technikai problémát (redundancia) jelez, hanem szemantikai tévedést. Sokszor nem azt a függést kell megszüntetni, ami matematikailag látszik feleslegesnek. A SYDES külön is kijelzi az olyan redundanciát, amelynél két reláció közös leíró tulajdonsága nem okoz normálforma gondot. A közös név ui. takarhat homonimát is.

A kérdéskört lezárandó utalnom kell egy általános félreértésre. A relációk átalakításával kapcsolatosan sokszor esik szó az ún. **attribútum-migrációról**, tulajdonságvándorlásról, amikor is egy tulajdonság az egyik relációból átkerül a másikba. A kifejezés teljes szemantikai félreértést tanúsít.

Tegyük fel, hogy a P413 példában az Ár eddig csak a CIKK relációt jellemezte, mostantól pedig csak a RENDELÉSTÉTELt írja le! Itt szó sincs 'migrációról'! A két Ár szemantikai tartalma ui. teljesen más. Ezért a SYDES modellező nem ismer tulajdonságváltoztatást, csak törlést és új bevitelt.

Szinonim leírók. A szinonim leírók egy részét éppen a normalizálás fogja feltárni. Lássunk erre egy példát:

P414 GÉPKOCSI (p:Rendszám, ..., Tulajdonoskód)
TULAJDONOS (p:Törzsszám, ..., Tulajdonosfajta)

A kulcsok közötti elemzéssel (szintézis) a Törzsszám a GÉPKOCSI relációba kerül. A reláció (újra)elemzésével (dekompozíció) fény derül a Rendszám ----> Törzsszám ----> Tulajdonoskód tranzitivitásra. Az utóbbit a TULAJDONOS relációba visszük át. Az újraelemzés ki fogja deríteni, hogy a Tulajdonoskód és a Tulajdonosfajta szinonimák.

A szinonimák egy része azonban így sem tárulkozik fel. Különösen az igen általános nevű tulajdonságoknál nem. Ilyen lehet pl. a 'Név' - 'Megnevezés' vagy a 'Mennyiségi egység' - 'Mértékegység' pár. Igen nagyméretű modellekben a több ezer attribútum között elemzés után is maradhatnak szinonimák. Ezek azonban leírók és komoly gondot nem okoznak.

4.4.2 VIRTUÁLIS ÉS VALÓS NORMÁLFORMA-HIBÁK

Az előző pontban a homonimákról és szinonimákról, valamint ezek kiszűrésének lehetőségeiről volt szó. A SYDES nemcsak a homonimák és szinonimák feltárásában ad támogatást. Segítségével a tervező szét tudja választani a valós normálforma problémát a látszólagostól. Erről és a szerepnév szemantikai konstrukciójáról, sajátos SYDES-felfogásáról szól ez a pont.

A 4.3.2 alpontban rámutattam a normálforma dekompozíció és szintézis ellentmondására (P411), amit a szakirodalom általában feloldhatatlannak tart. Kijelenthetem, hogy:

S441 A normalizálási eljárások ellentmondásai látszólagosak. Szemantikai és matematikai kettős alapon feloldhatóak.

A SYDES-ban az egyed-tulajdonság-viszony, az attribútum, kötelezően megadandó fontos jellemzőjének tartom a **függéserőt**. Ez a jellemző a **nem-alkalmazható** sajátos tulajdonságértéken alapul. Jobb hiányában jelöljük itt ezt az értéket '@'-vel!

Ez az érték a 'nulla' illetve az 'üres' három esetének az egyike. Megértéséhez vegyük alapul a Szüléseinek-száma nevű tulajdonságot a SZEMÉLY relációban! Ha egy nő nem szült, akkor az érték valós nulla (szám). Ha nem tudjuk, hogy hányszor szült, akkor az érték 'ismeretlen'. Ez nem-valós nulla. Ha pedig a kérdéses személy férfi, akkor a tulajdonságnak nincs sem valós, sem nem-valós nulla értéke. A tulajdonság a férfiakra értelmetlen és értéke nem-alkalmazható (@).

A fentiek alapján a SYDES-ban az A ----> B funkcionális függést **erősnek** nevezem akkor, ha a B tulajdonság az A minden lehetséges értékéhez felvesz a nem-alkalmazhatótól különböző értéket (ez lehet nem-valós nulla is). Egyébként a nevezett függés **gyenge** és a B attribútum értékhalmozának eleme a '@'.

A függéserő jellegének nagy szerepe van a normalizálásban. Térjünk vissza a P411 ominózus, ellentmondásos relációjára:

P415 RENDELÉS (p:Rszám, ..., f:Telepkód, f:Vevőkód)

Normálforma dekompozíciónál a megoldás elfogadott. Szintézis esetén viszont nem. Kérdés az, hogy miképpen oldhatja fel az ellentmondást egy SYDES-féle automata, amely a két eljárást ötvözi. Miként lehet, hogy az Rszám ----> Vevőkód függés a Telepkódon keresztül tranzitív is (szintézis), meg nem is az (dekompozíció). A megoldás a függéserőben rejlik. A RENDELÉS reláción belül a Telepkód gyengén függ az Rszám-tól, mivel a vevő által feladott rendelésekben nincs értéke. (Ez alternatívan áll a Vevőkódra is, de annak értéke mindig megadható és ezért nem tekintjük gyengének.)

A SYDES-koncepcióban a tranzitív függést **virtuálisnak** nevezem akkor, ha a függéssorozat (hármass) egy vagy több tagja gyenge. Egyébként a tranzitivitás **valós**. Következésképpen a SYDES az Rszám ----> Vevőkód függést áltranzitívnak mutatja ki és megszüntetését nem javasolja. (N.B.: ez vonatkozik a többi függésfajtára, valamint az összetett függésekre is.)

Azáltal, hogy a függéseket relációtól függetlenül is és reláción belül is vizsgálja a SYDES kettős algoritmus, az említett ellentmondást egyszerű módon oldja fel. Másféle, szemantikainak nevezett megoldást nem alkalmazok. Lássunk csak két egyéb 'javaslatot' a P415 probléma megszüntetésére!

Mivel a rendelőt kizárólagosan vagy a telep, vagy maga a vevő adja fel, egyesek egyszerű trükkhöz folyamodnak: kijelentik, hogy a vevő maga is egy telep. Vagyis nem kell külön Vevőkód a RENDELÉS relációba: a vevőkódok a Telepkód sajátos értékei. Ez a megoldás tökéletesen rossz. Szemantikailag zavaros. Külön Telepkód értékek kijelölését követeli meg a vevőnek, mint áltelepnek. Ezt vagy procedurálisan kell ellenőrizni, vagy be kell vezetni egy minősítést a relációba, ami mutatja, hogy a telep valójában vevő. Mindenekfelett

azonban a megoldás lehetetlenné teszi a hivatkozási integritást. Ezt a rossz módszert gyakorlati tervezők ajánlották a számomra!

Mások azt javasolnák, hogy a RENDELÉS relációban nevezzük át Rendelést-feladó-telepkódnak a Telepkódot. Az utóbbi meghatározza a Vevőkódot, az új nevű tulajdonság már nem, és így feloldódik az ellentmondás. Ez a javaslat az összefüggéseket félreértelmezi. Viszont alkalmas arra, hogy bemutassam a SYDES szerepnév-használatát.

A félreértés kettős. Ha nincs előre rögzítve, hogy melyik telepek adhatnak fel közvetlenül is rendelést, akkor nincs értelme az új névnek, az szemantikailag üres (ld. az alábbi érvelést), ill. a szintézis alapján a Rendelést-feladó-telepkód is meghatározza a Vevőkódot, hiszen a rendelést feladó telep is csak egyetlen vevőhöz tartozik.

A relációs koncepcióban a doméjn nevét előlről minősítik és úgy alkalmazzák attribútum névként, amikor az attribútum eltérő szerepeit akarják hangsúlyozni (I13, 86.o.). Az így minősített nevet nevezik szerepnévnek. Ezzel az eredeti értelmezéssel szemben a szakirodalom példái a minősítést általánosan is kiterjesztik. Mondjuk Testsúly névvel képezik le a SZEMÉLY relációba a Súly doméjnt, holott a Súly név sem okoz félreértést. A kiterjesztett relációs modell egyenesen kötelez a szerepnevek általános használatára (B.függelék).

A SYDES-ban a neveket másként alkalmazom. Egyáltalán nem látom értelmét az olyan általános nevű doméjnek bevezetésének, amelyek ráadásul szemantikai zavart is okoznak. Itt van példaként a Magasság doméjn. Az első kérdés az, hogy helyes-e itt az implicit generalizáció foka? Miért nem Dimenzió vagy akár Mérték nevű egységet alkalmazunk? Csak a tervező szubjektív döntésén múlik a doméjn neve és tartalma? A második kérdés az, hogy a Magasság szemantikailag egységes doméjn-e? Azonos fogalom-e a személy, a hegy és a híd magassága, és a 'személy' az csak minősítő szerep (prefix)? Bizonyára nem!

A SYDES-ban a tulajdonságok (nevei) mindig egyedhez kötötten meghatározottak. A Személy-magassága nevet doméjn- és attribútum-névként közösen használom. Csak akkor vezetek be tényleges szerepnevet, ha a doméjn valós alkalmazásáról van szó, ill. a minősítés az értelmezéshez elkerülhetetlen. A szerepnév a SYDES-ban tehát nem kvázi-szinonima, mint a relációs modellben. A kérdéskörre a 4.4.5 alpontban még visszatérek.

4.4.3 A '@'-REDUNDANCIA ÉS A FELTÉTELES FÜGGÉS

A relációs koncepció nem szentel elegendő figyelmet a redundancia egyik sajátos esetének, holott a normalizáció ismert célja a fizikai redundancia és az azzal járó tárolási anomália elkerülése. Az alább bemutatott példában a redundancia nem az ismert anomáliák értelmében okoz karbantartási problémát. Talán emiatt nem fordult felé a figyelem.

P416 TULAJDONOS (p:Törzsszám, ..., Foglalkozás, Jövedelem, ..., Ágazat, Felügyelet, ...)

A példában a kulcs gyenge függéssel határozza meg a gépkocsi tulajdonosának valamennyi mutatott attribútumát. Ez azt jelenti, hogy a Foglalkozás és a Jövedelem @-értékű lesz, ha a tulajdonos nem magánszemély. Egyébként a másik két attribútumra igaz ugyanez. A leendő adatbázis minden tételénél előfordul a 'nem-alkalmazható' érték. Ez óriási redundancia.

Minden adatbázisban lehetnek @-értékek. A kerek alkatrésznél a hosszak, a szögletesnél az átmérőnek nincs értelme. A saját gyártású alkatrésznek nem lesz eladási ára stb, stb. Mindez még nem meríti ki a fizikai redundancia tényét. Fizikai átfedésről akkor beszélek (ld. a 4.5.2 alpontot!), ha egy reláció soraiban (tuple) egy B tulajdonság mindig ugyanazt az értéket veszi fel, ha egy A, nem feltétlenül explicit tulajdonság adott értékű.

A kedvezőtlen függés mindig fizikai redundanciát jelent, de nem minden fizikai redundancia indoka a nemkívánatos függés. A P416-ban az implicit Tulajdonostípus (magán, állami) explicitté tétele esetén sem találunk rossz funkcionális függést. Erőltetetten többértékű függést feltárhatnánk, de akkor nevetséges kulcsokat kapnánk. (Kétségtelen ui., hogy fennáll a Foglalkozás+Jövedelem --->> Törzsszám.)

A P416 reláció tehát nem bontható le hagyományos módon. Viszont a SYDES-ban a relációt kedvezőtlennek tartom, mivel magasfokú benne a **@-redundancia**. Azaz tulajdonságcsoporthoz veszik fel mindig ugyanazon @-értéket, ha az (implicit) Tulajdonostípus adott értékű. A helyzet feloldására egy új fogalmat vezettem be: a **feltételes függés** koncepcióját [H18].

Egy adott R (TULAJDONOS) relációban az Ad (Jövedelem) leíró attribútum feltételesen függ a reláció Ak (Törzsszám) kulcsától, ha a relációban létezik olyan - nem szükségszerűen explicit - Af (Tulajdonostípus) feltétel, amelynek Afi értéke (vállalat) esetén az Ad mindig a @ (nem-alkalmazható) értéket veszi fel.

Belátható, hogy a feltételes függés a tranzitivitás sajátos eseteként is felfogható. Ha ui. az Af (Tulajdonostípus) értéke adott (vállalat), akkor az Ad (Jövedelem) értéke is eleve meghatározott (@). Ez a függés @-redundanciát okoz és ezért ki kell küszöbölni. Az erre vonatkozó sajátos normalizálási eljárást és annak elvi alapjait a következő alpont fejt ki. Itt arra kell még utalnom, hogy a feltételes függés egy korlátsort is megfogalmaz. (Magán tulajdonoshoz nem adható meg (pl. Ágazat.) A relációs koncepció nem képes ezt kifejezni.

4.4.4. SPECIALIZÁCIÓ ÉS EGYEDALTÍPUSOK

A P416 reláció nagymértékű fizikai redundanciát implicál és ezért valamilyen módon fel kell bontani. A relációk nemcsak **horizontális** projekcióval, az attribútumok mentén, azok számát - a reláció fokát - csökkentve bonthatók le. Mód van a **vertikális**, szelekciót alkalmazó lebontásra is a relációs sorok (tuple) mentén úgy, hogy a reláció terjedelmét (range) csökkentjük. Elvileg a P416 felbontható így is:

P417 MAGÁN (p:Törzsszám, ..., Foglalkozás, Jövedelem)
ÁLLAMI (p:Törzsszám, ..., Ágazat, Felügyelet)

Ez a megoldás azonban több problémát vet fel:

- Mivel két alaprelációnak nem lehet azonos a kulcsa, új, nem-kapcsolódó neveket kell bevezetni a Törzsszám helyett.

- A ... módon jelölt tulajdonságok között lehetnek a kulcstól eredetileg erősen függők is (Tulajdonosnév). Ezeket is minősíteni kellene a lebontás során. Mindez gond, ha:

- Két relációval kell dolgozni akkor, amikor a közös (erősen függő) tételeket kezelik, pl. keresik egy balesetet okozó gépkocsi tulajdonosát. Az ilyen kezelés - procedurális.

- Ráadásul a kapcsolódó relációkban is minősítést kellene bevezetni. Pl. a GÉPKOCSI-ban kétféle Törzsszám adatot illene tartani. Amivel pedig ott vezetnénk be gyenge függést.

A P416 által mutatott problémát, a @-redundanciát, a SYDES-ban sajátos normalizálási eljárással oldjuk fel. Ez az eljárás a feltételes függésen alapul és lényege a következő:

A feltételes függést tartalmazó relációból vetítsük ki a feltételesen függő tulajdonságokat, mint leírókat. Az eltérő feltételértéktől függő tulajdonságok külön relációkba kerülnek. Az ilyen reláció(k) kulcsa(i) az eredeti reláció kulcsának a szerepneve(i), a feltételértékkel minősítve. Az eredeti relációban megmaradnak a kulcstól erősen függő tulajdonságok. A P416 relációjának a felbontása tehát a következő:

P418 TULAJDONOS (p:Törzsszám, Tnév, ..., Tulajdonostípus)
MAGÁN (p:M-Törzsszám, ..., Foglalkozás, Jövedelem)
ÁLLAMI (p:Á-Törzsszám, ..., Ágazat, Felügyelet)

A feltételes függésen alapuló, fent leírt normalizálási eljárást specializációnak nevezem. A specializáció valódi altípusokat eredményezett esetünkben, mivel az M-Törzsszám és az Á-Törzsszám a Törzsszám valós alhalmazait jelölő szerepnevek (ld. a következő alpontot!).

Könnyű belátni, hogy a javasolt normalizálási eljárással kapott P418 eredmény megoldja a P416-ban mutatkozó fizikai redundancia gondjait. Ugyanakkor viszont nem implikálja a P417 fentebb felsorolt anomáliáit. Normalizálási eljárásunk tehát kedvező. Komplikáltabb eseteire a következő alpont tér ki.

4.4.5 SZEREPNÉVEN ALAPULÓ SYDES-STRUKTÚRÁK

Az előbbi alpont rávilágított arra, hogy a normalizálás segít egy alapvető modellezési probléma megoldásában. Arról van szó, hogy a tervezők néha csak nehezen tudják eldönteni, hogy egy adott jelenséget egyedként vagy tulajdonságként modellezzenek-e. Ha csak megérzéseikre hagyatkoznának, akkor a tervezés elveszítené tudományos megalapozottságát.

A P416-os példa feloldása jól szemlélteti, hogy a modellezés során a tulajdonságot felválthatja az egyed, mint tükrözési kategória. Belátható, hogy a normalizálás általában is a kategóriák ilyesfajta szétválasztása felé hat. Hiszen a függési meghatározóknak mindig külön reláció, azaz egyed felel meg. Ezt további három példával is szemléltetem, mielőtt a szerepnevek struktúraformáló funkciójára áttérnék.

P419VEVŐ (p:Vkód, Vnév, ..., Egyszámlasz, Tartozás, ...)

Tegyük fel, hogy a vevők és az egyszámlák közötti viszony 1:1 fokú! Tételezzük fel, hogy vannak magán és vállalati vevőink és természetesen csak az utóbbiaknak van egyszámlájuk! Ekkor az Egyszámlasz, Tartozás, ... sor @-redundáns és létezik a Vkód ----> (Vtípus) --m-> Egyszámlasz, Tartozás, ... feltételes függés. (Az implicit tulajdonság zárójelben van. A --m-> jelölés mutatja, hogy a függés a meghatározó m=magán értéke esetén áll csak fenn.)

Az előző alpontnak megfelelően itt is felmerülhet a feltételesen függő attribútumok külön relációba való leválasztása. Vagyis az, hogy egyedet készítünk tulajdonságból. Az a tény, hogy itt a specializáció 'féloldalas', mivel a magán vevőkhöz nem készül külön reláció, nem zavaró. Csupán azt mutatja, hogy a feltételes függés **nem alternatív** [H18]. A lebontás módja azonban sajátos. Az Egyszámlasz meghatározza a {Tartozás, ...} sort és fennáll a Vkód <---> Egyszámlasz **kölcsönös függés**. A leválasztásnál nem szükséges, hogy az új reláció kulcsa a régiének a szerepneve (E-Vkód) legyen. Sőt, az új egyed leendő neve (EGYSZÁMLA) sem utal a régre.

Amikor egy relációt a feltételes függésre alapozva úgy bontunk meg, hogy az új reláció kulcsa az eredeti relációnak az attribútuma és azzal

kölcsönös függésben áll, **virtuális specializációnak** nevezem. Az ilyenkor keletkezett egyed ui. az eredeti egyednek nem valós altípusa.

Az alábbi példa keményen szemlélteti a hagyományos relációs normalizálás hiányosságait a feltételes függés területén:

P420 ERŐFORRÁS (p:Ekód, Személyisz, Személynév, ...,
Géptörzssz, Amortizáció, ...,
Cikksz, Anyag-megnevezés, ...)

Ez a példa nem kitalálás: a gyakorlat során fedeztem fel. Próbáltam normalizálni: sikertelenül. A személyek, gépek és anyagok ui. nem egyszerre jelennek meg egy relációs sorban és így a reláció join-függéssel sem bontható le. A P420 reláció ötödik normálformájú. Mégis evidens, hogy igen rosszul tervezett, mert benne hatalmas fokú a @-redundancia.

A P420 esetében a SYDES-módszer a feltételes függésen alapuló **totális specializációt** alkalmazza. Ez a nevét onnan kapta, hogy az eredeti reláció, mint típus teljesen feloldódik a keletkező altípusokban. Ui. a SZEMÉLY, GÉP és ANYAG relációk megalkotása után nincs értelme megtartani az eredetit.

Végül bemutatok egy nem-alternatív, de valós specializációt:

P421 RENDELÉS-2 (p:Rszám, Rdátum, ...)
EXPORT-RENDELÉS (prf:E-Rszám, Valutanem, Váltás, ...)

Az export-rendelés is egy rendelés, annak (valós) altípusa. A rendelések közül csak az export-rendeléseket jellemzi a Valutanem, Váltás, ... attribútumsor. A többi rendelésben ezek @-értékűek.

Ahhoz, hogy az összes rendelés ne hordozza a hatalmas @-redundanciát, az eredeti RENDELÉS-1 itt nem mutatott relációját lebontottuk a P421 két relációjára.

N.B.: Az EXPORT-RENDELÉS kulcsa előtt új jelölés látható. Az E-Rszám nemcsak elsődleges kulcs (p). A másik reláció kulcsának a szerepneve (r) és így kapcsolatot teremt a RENDELÉS-2 felé, tehát tulajdonképpen a szerepnév egyben idegen-kulcs (f) is. A SYDES koncepcióban ez az egyetlen eset, amikor a kapcsolatot két elsődleges kulcs (és nem egy elsődleges és egy idegen kulcs) alapozza meg, és az 1:1 függés megengedett.

Az ilyen lebontással kapott egyedet hívjuk **specializált egyednek**, másképpen egyedaltípusnak a SYDES-ban. A ilyen egyed kulcsa az eredeti egyedének a szerepneve, vagyis értékeit ugyanabból a doméjnból veszi, mégpedig valós alhalmazként. Azaz minden E-Rszám is egy Rszám érték, de ez fordítva nem igaz. A specializált egyed **explicit** leíró tulajdonságai az eredeti reláció feltételesen függő attribútumai. Az altípusnak vannak **implicit** leíró tulajdonságai is; az EXPORT-RENDELÉS-t jellemzi a RENDELÉS-2 összes leíró tulajdonsága. Az eredeti reláció megbontásával kapott két egyed explicit leíró tulajdonságai nem fednek át. A két egyed között 1:1 fokú, felfelé kötelező kapcsolat van (vö. létezésfüggéssel).

Az egyedaltípusok, mint fontos fogalmi konstrukciók, nemcsak tisztán **extenzionálisan**, vagyis az attribútumsor felbontásával, tagolásával fejezhetők ki a SYDES-ban. Az altípus megadásának létezik egy elsősorban **intenzionális**, tehát a relációs sorok (tuple, előfordulás) besorolásán alapuló módja is. A relációs koncepcióban ismeretlen, de egyébként széleskörűen használt struktúrális elemről, a rekurzív vagy visszamutató kapcsolatról van szó.

P422 DOLGOZÓ (p:Törzsszám, Dnév, ..., rf:Vezetőkód)

Dolgozóink egy része vezető. Minden dolgozónak legfeljebb egy vezetője van. A Törzsszám tehát funkcionálisan meghatározza a Vezetőkódot. Ezen túlmenően azonban a vezető maga is egy dolgozó (altípus), akit jelen esetben nem kívánunk külön attribútumokkal leírni. Ahhoz, hogy rámutassunk a dolgozó vezetőjére, a Vezetőkódot a Törzsszám szerepneveként (r) határozzuk meg úgy, hogy ez a sajátos tulajdonság a reláció bizonyos sorait egy másikhoz kapcsolja (f) visszamutatással. Az előfordulásokat osztályozzuk, intenzionálisan.

Itt kívánom megjegyezni, hogy a SYDES szerepnév fogalma nem azonos a relációséval. Egyrészt nálam nem kikötés, hogy a szerepnév a doméjn nevének előlről való minősítésével szülessen. A Vezetőkód a Törzsszám szerepneve és nem kell a Vezető-törzsszám nevet viselnie (bár tehetné). Másrészt a SYDES-ban megengedett a szerepnév-hierarchia. Pl.: a Törzs-szám lehet a Személyszám szerepneve. A SYDES nem zárja ki a szerepnevek hálós viszonyát. A Motorcsónak-id lehet egyszerre a Vizijármű-id és a Motorosjármű-id szerepneve is. Éppen a fenti két ok miatt a szerepnév nem csupán előtét (prefix).

Az előzőekben kifejtettem azt, hogy vannak visszamutatással modellezhető altípusok. Ezzel nem azt mondtam, hogy a rekurzió csak altípus meghatározásra szolgál. Az involúció kifejezheti az egynemű dolgok együvé tartozását is. Azonban mindig csak olyankor, ha az összetartozó dolgok kapcsolata 1:N fokú (ill. ennek a eseteként 1:1 fokú). Viszont be kell látni, hogy vannak esetek, amikor egy jelenség előfordulásainak viszonya hálós (M:N fokú). Lássunk egy példát:

P423 TÉTEL (p:Tételid, Tnév, ...)
CSALÁDFA (p:{rf:Felépülő-Tid+rf:Beépülő-Tid}, Darab)

A példa egy ipari környezet tipikus viszonyát mutatja. Egy termelési tétel több másikkal épülhet fel és több másikba épülhet be. A hálós visszamutatás tükrözésére vezettem be a **családfa-egyed** fogalmát. Az ilyen egyed kulcsa összetett. Két része az alapreláció kulcsának a szerepneve (r), és a hálós viszonyt kifejezendő kapcsolatot teremt (f) az alapreláció kulcsához (vö. létezésfüggéssel).

Általában is találhatunk két reláció között egynél több kapcsolatot úgy, hogy elkerüljük a felesleges normalizálást:

P424 TELEPÜLÉS (p:Tkód, Tnév, ...)
DOLGOZÓ (p:Szszám, ..., rf:Lakó-Tkód, rf:Munka-Tkód)

Nem kizárt, hogy egy dolgozónk más településen dolgozzon, mint ahol lakik. Ha csak ezt a tényt akarjuk rögzíteni, akkor helytelen volna külön, a dolgozók és települések kettős összefüggését tükröző relációba kiemelni az 'ismétlődő' településkódot. A DOLGOZÓ reláció a SYDES-ban jól-tervezettnek számít. Egy másik reláció kulcsának két szerepneve (r) idegen kulcsként (f) kapcsolatot teremt az adott reláció felé. Ezt a szerkezetet **többszörös kapcsolatnak** hívom.

A SYDES-módszerben leíró és kulcsrész attribútumként szerepnevet alkalmazok akkor is, ha ezzel elkerülhetem a szemantikai félreértést és/vagy világosabbá tehetem a modellt. Néha a kényszer diktálja a szerepnévnek a fenti speciális konstrukciók nélküli alkalmazását. Az ilyen esetekben beszélek **minősített névről**.

P425 MŰSOR-TERV (p:{Adó+**Terv**-Adásnap+Műsorszám}, ...)
MŰSOR-TÉNY (p:{Adó+**Tény**-Adásnap+Műsorszám}, ...)

A szerepnév itteni használatát nem kell magyaráznom. Az alpontot lezárhatom azzal a következtetéssel, hogy a szerepnév igen fontos szemantikai konstrukció, amelynek a relációs modell nem szentel elegendő, illetve kellő módon figyelmet.

4.5 A NORMÁLFORMÁK HATÁSA AZ ADATMODELL OPTIMUMÁRA

Az alábbiakban a közismert (1-5NF és BCNF) normálformákra vonatkozó megfontolásaimat ismertetem. A részletek tárgyalása előtt utalnék arra, hogy ezek a konstrukciók eleve csak korlátozott szerepet játszhatnak az adatmodellezésben, mivel csak kivetítési (projection) és összekapcsolási (join) szempontból optimalizálnak [I24, 263.o.]. Ezért a minimalitás, a teljesség és az egyértelműség ismérveinek a terén a normálformák csak részleges előrelépést jelentenek.

Mindez nem meglepő, mivel a reláció, a normálforma és a normalizálás nem fogalmi, hanem logikai szintű tényezők. Ennek dacára a normalizálást igen hasznos modellezési módszernek tartom, feltéve, hogy a mögöttes szemantikát gazdagítjuk.

T45 A normálformák és a normalizálás alapkoncepciói önmagukban nem alkalmasak arra, hogy segítségükkel lényegesen közelítsük az adatmodell optimumát. Kellő szemantikai kiegészítésekkel a normalizálás fogalmi modellezési értékét jelentősen növelhetjük.

Az alábbiakban először az ismétlődés problémáját vizsgálom. Ezután a részleges és a tranzitív függést együtt elemzem, mivel bennük sok közös vonást látok. Harmadikként tárgyalom a kulcstörő függést, ami sajátos szemantikai problémákat vet fel. Innen jó átvezetés adódik a röviden tárgyalt negyedik és ötödik normálformához.

4.5.1 NEM-NORMALIZÁLT RELÁCIÓK (ONF)

Azt a relációt, amely ismétlődő (periódikus) tulajdonságot vagy tulajdonságcsoporthoz tartozik, **nem-normalizáltnak** nevezzük. Az

ismétlődés szemléleti és kezelési egyoldalúságot okoz (vö. 1.1.2). Ezért az ismétlődés leválasztása külön relációba általában kedvezőbb struktúrát eredményez. (N.B.: ezt a megbontást a relációs modell kötelezővé teszi, mivel az ismétlődést tartalmazó, nem 'flat' struktúrát matematikailag eleve nem is szabadna relációként értelmezni.)

P426 BALESET-1 (p:Kárszám, Hely, ...(f:Rendszám, Kár-érték))

A belső zárójelben lévő tételek ismétlődnek. Így nehéz meghatározni azt, hogy egy gépkocsi milyen balesetekben vett részt. Könnyű viszont kikeresni az egy balesetben részt vett gépkocsikat. A struktúra féloldalas, eleve egy adott irányú kezelésnek kedvez. Ezért indokolt az ismétlődés leválasztása:

P427 BALESET-2 (p:Kárszám, Hely, ...)
KOCSIKÁR (p:{f:Kárszám+f:Rendszám}, Kár-érték)

A kocsikárok homogén módon érhetők el a két kulcsrész alapján. Úgy tűnik tehát, hogy célszerű törekednünk ismétlődés-mentes relációk megalkotására. Azonban ez nem történhet mindig olyan egyszerű módon (kivetítés), mint fent. Ezzel kapcsolatosan meg kell fogalmaznom egy segédtelet.

S451 A relációs adatmodell nem mérlegeli kellőképpen a periodicitás különböző szemantikai aspektusait. Ennélfogva nem képes arra, hogy differenciált optimumot nyújtson.

A modellezésben különbséget teszek az úgynevezett **valós** és a **virtuális** ismétlődés között. A megkülönböztetést indokolja, hogy a tulajdonságokra nevekkkel hivatkozunk, amelyek sajátos szemantikai tartalmat feltételeznek. A P426 példában jogos volt a lebontás. De alkalmazható-e ez a módszer alább?

P428 KIKÜLDETÉS-1 (p:Kszám, Szállásktg, Napidíj, Egyébktg)
KIKÜLDETÉS-2 (p:Kszám, Költség-1, Költség-2, Költség-3)

A második relációban a Költség háromszor ismétlődik, de csak virtuálisan. Ui. nem azonos a három költség-attribútum szemantikai

tartalma! Az 'ismétlődés' leválasztása nem jogos. Nincs rá matematikai mód és a megbontás nem visz optimumra.

Az adatmodellezésben célszerű különbséget tenni a **kulcs** és a **leíró** tulajdonságok ismétlődése között. Inhomogenitás csak az előbbieket esetében lép fel (P426), a leíróknál (P428) nem. A leíró (Napidíj) kulcs (Kszám) nélkül ki sem kereshető! A KIKÜLDETÉS-x reláció nem bontható meg normalizálással, azaz kivetítéssel, mivel a költségekhez hiányzik az azokat részben meghatározó kulcs. Alternatív megoldás lehet az alábbi:

P429 KIKÜLDETÉS-3 (p:{Kszám+Ktípus}, Költség)

Ez a megoldás nemcsak új kulcsrész (Ktípus), hanem egyben új reláció (KTÍPUSOK (p:Ktípus, Ktípusnév)) bevezetését is feltételezi. A P429 tehát rosszabb megoldás, mint a P428, mert sérti a minimalitás elvét. Kifejtésem válasz az [I29]-nek is, amely a P428 és P429 változatokat egyenrangúaknak tartja. Ha azonban a költségek sokszor nem-alkalmazható (@) értékűek, vagy a Ktípusok köre bővíthet, akkor valóban a P429 a jó megoldás.

A fenti példák egy másik vetületre is felhívják a figyelmet. Az ismétlődést célszerű megkülönböztetni **hierarchikus** vagy **hálós** jellege szerint. A két esetben ui. más a lebontás módja. Az ismétlődés hálós, ha az ismétlődésben van a kulccsal kölcsönösen független idegen kulcs (P426). Hierarchikus, ha a kulcs függ az (implicit) idegen kulcstól, mint alább:

P430 KERÜLET (p:Kerületszám, ... (f:Írányítószám, ...))

A P426-ban Kárszám <-/-> Rendszám. A lebontással összetett {Kárszám+Rendszám} kulcs keletkezik úgy, hogy mindkét rész maga is kulcs lesz. Lesz három reláció: BALESET, KOCSIKÁR és GÉPKOCSI. A P430-ban Írányítószám ----> Kerületszám. Ezért a lebontás csak két relációt (KERÜLET, KÖRZET) eredményez és azok kulcsa egyszerű.

Bebizonyosodott, hogy az alapnormalizálás, vagyis az ismétlődő tételek kiküszöbölése a nem-sík relációkból, sokszor nem egyértelmű, pusztán matematikai úton nem megoldható probléma. A valós és virtuális; a kulcson és a leíron alapuló; a hálós és a hierarchikus ismétlődést fel kell tudni ismerni, mivel esetükben más-más megoldás

az optimális. A függés jellegének a felismerése olyan szemantikai feladat, ami az adott reláción túlmutató elemzéseket is megkövetel.

4.5.2 RÉSZLEGES ÉS TRANZITÍV FÜGGÉSEK

Azt a relációt, amely ismétlődést nem tartalmaz, legalább első normálformájúnak (1NF) nevezzük. Ha a reláció az előzőn kívül még ún. részleges függést sem implikál, akkor legalább második normálalakban (2NF) van. A reláció harmadik normálformájú (3NF), ha minden leíró tulajdonsága függ a kulcstól, a teljes kulcstól és csakis attól (nincs tranzitívitas).

P431 EGYEDI-KÁR (p:{f:Kszám+f:Rendszám}, ..., Kocsitípus)
RENDELÉS-1 (p:Rszám, ..., f:Vevőkód, Vevőnév)

Az első reláció csak 1NF, mivel részleges függést tartalmaz. A második, a tranzitív függés miatt, csak 2NF. Mindkét esetben azonos tárolási és karbantartási (hozzáadási, törlési és módosítási) anomáliák lépnek fel. Mindkét esetben ugyanaz a megoldás módja: a kedvezőtlen függések kivetítése külön relációba. A RENDELÉS-1 megbontása ez:

P432 VEVŐ (p:Vevőkód, Vevőnév, ...)
RENDELÉS-2 (p:Rszám, ..., f:Vevőkód)

A 4.1 pontban döntési táblázattal rámutattam arra, hogy vannak olyan függéskombinációk, amelyek tranzitívitasnak tűnnek, de annak módjára nem oldhatók fel a függés trivialitása miatt. Már ez is feljogosít az alábbi kijelentésre:

S452 A relációs modell nem szentel kellő figyelmet a részleges és tranzitív függések mögötti belső szemantikának.

Az adatmodell tervezésben különbséget teszünk logikai és fizikai redundancia között. Az átfedést logikainak nevezem, ha ugyanaz a tulajdonság azonos névvel és szemantikai tartalommal több relációban is előfordul úgy, hogy egyik relációban sem kulcs. A P432 példában a Vevőkód tehát nem-redundáns (mert kulcs). Ha azonban a

RENDELÉS-1 és a VEVŐ reláció egyidejűleg létezne, akkor logikailag redundáns lenne bennük a Vevőnév.

A redundancia fizikai, ha egy reláción belül a kulcs értékétől függetlenül ugyanaz az érték(sor) jelenik meg. Pl.: több Rszám-nál azonos a Vevőkód és a Vevőnév párosa. A logikai redundancia mindig fizikaival jár, de nem minden fizikai redundancia jelent egyben logikait is. A megkülönböztetésnek több oka és következménye van.

Logikai redundancia esetén nem lép fel a hozzáadási és törlési anomália. Tehát az erre vonatkozó kitétel [I19] nem kellően árnyalt. Ilyenkor csakis a szemantikai tartalom alapos elemzése árulja el, hogy az átfedés valós, vagy a közös név miatt csak látszólagos (homonima). A logikai redundanciát és az azon alapuló normálforma-hibát automatikus eszközzel ki lehet szűrni. A csak-fizikait nem. Végül: a fizikai redundancia tárolási és karbantartási anomáliát akkor is okozhat, ha nincs explicit normálforma hiba. Ha a RENDELÉS-1 relációban nem szerepelne a Vevőkód, akkor is csak látszólag volna kedvező a normálformája. Az előbbieket miatt különbséget teszünk **nyílt** és **rejtett** részleges, ill. tranzitív függés között. (Ld. 4.3 pont!) Utaltam rá (4.4 pont), hogy a függés lehet **valós**, ill. **virtuális**, amit mérlegelni kell, mielőtt anomáliákról szólnunk.

Legújabbban nagyobb hangsúlyt fektetnek az időre, mint modellezési tényezőre. Ennek megfelelően célszerű lenne megkülönböztetni **stabil** és **dinamikus** függéseket. Az előbbi esetében ui. egyáltalán nem lép fel a változtatási anomália. Például: a gépkocsik típuskötött sajátosságait elvileg külön KOCSITÍPUS (p:Kocsitípus, ...) relációban is modellezhetjük. Mivel ezek a tulajdonságok nem módosulnak, a reláció alkalmazása lehet kérdéses akkor is, ha a 3NF-hez szükséges.

A relációs koncepció nem tudja kezelni a logikai redundancia jelenségét, a rejtett, illetve a virtuális függést, valamint a stabil függést. Minden esetben ugyanazt a mechanikus megoldást kínálja és erőlteti: a kivetítést mint normalizálást.

4.5.3 A KULCSTÖRŐ FÜGGÉS

Az a reláció, amely ismétlődést, részleges és tranzitív függést nem tartalmaz (3NF), még mindig felmutathat nem-kíváncsú jelenségeket, amint azt az alábbi példa szemlélteti:

P433 UTCA-1 (p:{Kerület+Utcasorszám}, ..., f:Irányítószám)

Feltételezzük, hogy az utcákat kerületen belül(!) egyedi sorszám azonosítja és Irányítószám ----> Kerület.

Azt a jelenséget, amikor egy összetett kulcs által meghatározott tulajdonságtól függ az összetett kulcs egy része, **kulcstörő függésnek** nevezzük. Ez a jelenség karbantartási anomáliákat okoz, mivel két tulajdonság (itt: a Kerület és az Irányítószám) kettős viszonyban áll egymással. A probléma általánosan is felvethető: Ha egy reláción belül két (vagy több) meghatározó van - mint a P433-ban a kulcs és az Irányítószám – és azok nem állnak kölcsönös függésben (vagyis nem kulcsjelöltek), úgy karbantartási anomália lép fel. Ha a relációban csak egy meghatározó van, akkor az Boyce-Codd normál-formájú [BCNF, I18]. (N.B.: ekkor már értelemszerűen legalább 3NF alakú is!)

A szakemberek egy része kezdetben azt javasolta, hogy kulcstörő függés esetén bontsuk le a relációt BCNF-re. Például:

P434 UTCA-2 (p:{Kerület+Utcasorszám}, ...)
KÖRZET (p:Irányítószám, f:Kerület)

Mások [I24, 255.o.] helyesen utalnak arra, hogy ez a megbontás információvesztéssel jár. (A keletkezett két relációból az eredeti nem állítható elő - nem tudjuk az utca irányítószámát meghatározni.)

Ebből arra következtetnek, hogy az ilyen reláció nem lebontható. Az utóbbi következtetésükkel csak részben értek egyet. Az alábbi kitételek kell tennem:

S453 A relációs koncepció BCNF alakja jól mutatja, hogy a helyes normálforma a legtöbb esetben nem érhető el csak matematikai alapon. A megfelelő reláció megtervezéséhez kettős - matematikai-szemantikai - elemzés szükséges.

Szerintem a tervezés nem merülhet ki abban, hogy csak a meglévő vagy feltételezett relációk adott tulajdonságainak az összefüggéseit elemezzük. A tervezés alkotó és átalakító folyamat is. A SYDES elemzés valóban fogalmi, szemben a tiszta normalizálással.

A P433 példa esetén matematikai alapon feltárom a kulcstörő függést: azt, hogy a relációban több meghatározó van. Nem próbálok

normalizálni, mert az tényleg nem vezet sikerre. A tartalmi elemzés viszont igen gyorsan kimutatja, hogy három fogalom - kerület, körzet és utca - természetes hierarchiát alkot. A P433 nem a normálforma hiba miatt rossz - az csak a jelenség -, hanem amiatt, hogy a kulcs sérti a természetes hierarchiát. Ezért helytelen a P434 megoldás is.

Ilyen esetben a SYDES-módszerben azonosítóváltoztatást javaslunk. Két megoldási változat közül is lehet választani:

P435 UTCA-3 (p:{Abszolút-utcasorszám}, ..., f:Írányítószám) vagy
 UTCA-4 (p:{Írányítószám+Utcasorszám}, ...)
 és KÖRZET (p:Írányítószám, ..., f:Kerület)
 és KERÜLET (p:Kerület, ...)

Az UTCA-3 reláció kedvezőbb, mivel a körzet-utca viszony idővel megváltozhat. Nem vitás, hogy ezek a megoldások elvileg tiszták és az sem, hogy azok nem pusztán matematikaiak. A természetes hierarchiát szemantikai alapon jól tükrözik. A szemantikai elemzést egy másik példán is bemutatom:

P436 OKTATÁS-1 (Diákkód, Tárgykód, Tanárkód)
 felbontva:
 OKTATÁS-2 (p:{Diákkód+Tárgykód})
 TANÁR (p:Tanárkód, Tárgykód)

A példát és annak felbontását az [I24, 251.o.] alapján tárgyalom. Feltételezés, hogy minden tárgyat több tanár ad elő, de egy tanár csak egy tárgyat oktat. Minden diákot csak egy tanár tanít adott tárgyra. Vagyis két függés áll fenn: Tanárkód ----> Tárgykód és Diákkód+Tárgykód ----> Tanárkód. A kulcstörés nyilvánvaló. A forrás szerint a két keletkező relációból az eredeti nem állítható elő, ezért az eredeti reláció nem bontható meg. A BCNF tehát nem segít.

A SYDES-módszerben a probléma értelmezése és megoldása más. A tárgy, a tanár és az oktatás fogalmak viszonya ebben a sorrendben hierarchikus úgy, hogy az első és a harmadik viszonya közvetett. Ezért a P436 helyett a P437-et javaslom:

P437 OKTATÁS-3 (p:{Tanárkód+Diákkód})
 TANÁR (p:Tanárkód, Tárgykód)

Szemben a P436-al itt a két reláció egyértelműen kapcsolható. Ott nem lehet a tárgyból a tanárra következtetni, itt a tanár egyértelműen mutatja a tárgyat. Igaz, a karbantartási anomáliát a P437 sem oldja fel. Erről a legutolsó alponthoz szólnok. Itt le kell szögezni egy alapvető SYDES-szabályt:

Egy reláció megbontása akkor és csak úgy megengedett, ha az eredményezett relációk kulcsai között funkcionális függések - azaz egyértelmű viszonyok - állnak fenn.

4.5.4 MAGASABB NORMÁLFORMÁK ÉS EGYÉRTELMŰSÉG

Az előző alponthoz láttuk, hogy egy 3NF reláció mutathat még karbantartási gondokat, amelyek BCNF módon nem (mindig) oldhatók fel sikeresen. Ezért vezették be a **többértékű függés** fogalmát [I22]. A konstrukció alkalmazását indokolni látszik egy sajátos redundancia és az abból fakadó karbantartási anomália, amit a következő példa szemléltet [I24, 256.o.]:

P438 TANÍTÁS-1 (Tanfolyam, Tanár, Tárgy)

Feltételezzük, hogy egy tanfolyamon több tanár tanít és a tanfolyamon több tárgy van. Ha egy tanár oktat a tanfolyamon, akkor minden tárgyat is oktat. Megfordítva: a tanfolyamon oktató tanárok mindegyike előadja mindegyik tárgyat. Ha tehát egy tanárt illesztünk be, akkor annyi sor kerül a relációba az adott tanfolyamhoz, ahány tárgy van azon. Ha egy tárgyat viszünk adott tanfolyamhoz, akkor annyi sort illesztünk be, ahány tanár tanít a tanfolyamon. Evidens, hogy a P438 redundáns és többszörös karbantartást igényel.

Amennyiben egy reláción belül egy tulajdonsághoz (Tanfolyam) egy másiknak (Tanár) mindig ugyanaz az **értékkészlete** tartozik függetlenül a többi tulajdonságtól (Tárgy), úgy az adott tulajdonság többértékű függéssel határozza meg a másikat. A többértékű függés alternatív. Azaz példánkban fennáll a Tanfolyam --->> Tárgy viszony is. A reláció a többértékű függés mentén megbontható. Az a reláció, amelyik ilyen függést nem tartalmaz, legalább **negyedik normálformájú**:

P439 RÉSZTVEVŐJE (p:{Tanfolyam+Tanár})
TARTALMAZZA (p:{Tanfolyam+Tárgy})

A P439 két relációja a P438 relációnak a többértékű függések mentén történt lebontásával keletkezett és így legalább 4NF (ténylegesen 5NF) alakú. Nem vitás, hogy a P439 feloldja a P438 redundancia-okozta anomáliáit. Mellesleg a P439 mindkét relációja **csupa-kulcs** és így alakja egyben BCNF. Ez a kitétel általánosítható: minden 4NF-reláció egyben BCNF is.

A többértékű függést és az azon alapuló normalizálást a bemutatott előnyök dacára sem tartom használható modellezési konstrukciónak. Ennek több oka van:

- Az előző alpont végén tett kitételemet fenntartom. A P439 relációi a lebontás után elméletileg ugyan kapcsolhatók, de gyakorlatilag(!) a megbontás után nincs akadálya, hogy egy tanfolyamra olyan tanár kerüljön, aki egy adott tárgyat valójában nem oktat - és ez fordítva is megtörténhet.

- Azaz valójában az OKTATJA (p:{Tanár+Tárgy}) relációt hiányolom. Ez nemcsak korlát a P439-re, hanem feloldja annak hozzáadási-törlési anomáliáit is. A P439 képtelen tükrözni azt, hogy egy tanár milyen tárgyakat oktat mindaddig, amíg nincs befogadó tanfolyam... Ez vonatkozik a tárgyra is.

- A többértékű függés, impliciten, eleve az adatmodell hiányosságát és szubjektívitasát implikálja. Ez a kitétel magyarázatot igényel.

A 4.3 pontban kétféle alapvető normalizálási módszert mutattam be: a dekompozíciót és a szintézist. A többértékű függés konstrukciója az utóbbi esetén nem alkalmazható, mivel ilyen függés eleve csak adott reláción **belül** értelmezhető. (Ezzel, alkalmazási köre is leszűkült.) A dekompozícióról viszont a 4.3 pont bizonyította, hogy elvileg nem ad biztosan teljes modellt. Ez áll a többértékű függésre is. Tegyük ehhez, hogy a P438-ban ismertetett relációt egyetlen tervező sem határozná meg alaprelációként. Az tehát csak egy egyedi szemlélet ('view' - [I24], 159.o.). Azt pedig aligha kell bizonyítani, hogy az egyedi szemléletek egyszerű összegzésével nem érhető el optimális modell.

A többértékű függés kizárja a szintézist, illetve az általam javasolt (4.3.2) ötvözött elemzési módszert. Nem biztosítja a modell teljességét és teret enged a többértelmű kapcsolatoknak. Ezért nem alkalmazom a SYDES-koncepcióban. Ezzel módszeremet nem éri komoly veszteség, mert a matematikai és szemantikai elemek egymást támogatják. Így én szemantikai alapon találok meg a P433 kulcstörési probléma P435 megoldását, amit más a többértékű függéssel fedezne fel.

A SYDES-módszerben a 4NF minden további nélkül elérhető. A 4NF mögött ui. mindig valamilyen összetett kulcs áll. Ha az összetett kulcs meghatározó (pl. Tanfolyam+Tárgy ----> óraszám), akkor nem a többértékű függés alapján fogjuk modellezni. Egyébként minden olyan relációpár, amely M:N fokú viszonyban áll egymással, egymáshoz köthető a kettő kulcsából álló csupa-kulcs relációval (Tanár+Tárgy).

Csak szemantikai elemzés döntheti el, hogy erre mikor van szükség. A többértékű függés ellen szól egyebek közt az is, hogy automatával nem fedezhető fel. Ez még inkább vonatkozik az ún. **összekapcsolás-függésre** (join-dependency, JD). Itt sokszor 'patologikus-esetek' [I24, 263.o.] is adódnak, amelyek még tapasztalt tervezők által sem tárhatók fel. Azonban nem ez a fő oka annak, hogy az ötödik normálformáról csak igen röviden emlékezek meg. Problémáimat az alábbi példa mutatja:

P440 SZCP (Szállítószám, Cikkszám, Projektszám)

A szállító/cikk/projekt SZCP reláció '**hármasan-lebontható**' a tulajdonságok párosaiból álló három relációra. A lebontásnak az az értelme, hogy a lebontott relációkban nem fogjuk többszörösen tükrözni pl. azt, hogy egy szállító milyen cikkeket képes szállítani. Tehát mintegy megoldódik a 4NF általam is felvetett problémája. Vagyis az, hogy a P439 a három fogalom (tanfolyam, tanár és tárgy) között, amelyek M:N-es viszonyban állnak, csak két összefüggést rögzít le a három helyett.

A szakirodalom szerint [I24] az ötödik normálforma a végső, tovább nem fokozható normalizálási alak. Ebben nincs okom kételkedni. Mint normálalak, az 5NF is csak kivetítési és összekapcsolási szempontból lehet optimális. Aligha vitatható ui., hogy az $R_x(A, B, C)$ reláció helyett az $R_y(A, B)$, $R_z(B, C)$ és $R_q(C, A)$ relációhármassal alkalmazása körkörös redundáns. Vagyis a minimalitás szempontjából nem

optimális. Ezért szokásos kitétel, miszerint a SYDES az ötödiknél kedvezőbb normálformát biztosít, elméletileg megalapozott is lehet. Ezt a gondolatot a következő alpont fejezi be.

4.6 RELÁCIÓK ÉS A SYDES - ÖSSZEFOGLALÓ

Az összefoglalót egy ilyesfajta dolgozatba talán nem illő, viszont a szemléletessége miatt talán mégis elfogadható képpel kezdem. A relációs függések, normálformák és normalizálási eljárások a matematikailag **steril** környezetrendszeret vetítik a tervező elé. Fel sem tételezik, hogy ez a környezet lehet **szemantikailag szennyezett** is. Ebben persze nem a matematika a hibás. Viszont a gyakorlatban kevésbé járatos, tapasztalatlan tervezők megtévedhetnek.

Mindjárt az összefoglaló elején meg kell kérdőjeleznem egy minden relációs tervező által szilárdnak hitt alapot. Az Armstrong-féle függési axiómákról van szó [I21], közelebbről az additivitási szabályról. Ennek értelmében ha fennállnak az $A \rightarrow B$ és $A \rightarrow C$ függések, akkor igaz, hogy $A \rightarrow B, C$. Ezt a SYDES-ban így jelöljük: $A \rightarrow \{B+C\}$.

Már régóta tudjuk, hogy az A plusz B nem azonos az A meg a B minőségi együttesével. Ezt az alábbi példa is igazolja:

P441 MUNKA (p:Mszám, ..., f:Vállalat, f:Anyag)

A reláció a főberuházót és a kritikus anyagot akarja rögzíteni. Egyenkénti kapcsolatban áll a vállalatokat és az anyagokat leíró másik két relációval. Hiba lenne összekapcsolni az ANYAGSZÁLLÍTÓ (p:{Vállalat+Anyag}) relációval, amely kulcsának részei látszólag(!) fellelhetők a P441 relációban. Vagyis az Armstrong-tétel szemantikai szempontból igen kétséges.

Általában véve is a relációs koncepció túlzott jóhiszeműséget feltételez a normalizálási módszerekben. Az igaz, hogy a P438 lebontásának eredménye, a P439 struktúráisan kedvezőbb. De mi gátolja meg a felhasználót a feltételezésnek ellentmondó adat bevitelében?

Ezért a relációs tervezéssel, általában az adatmodellezéssel kapcsolatosan megfogalmazható fontos kitétel az alábbi:

S454 Minden adatmodellezési struktúra az egyéb megfogalmazásokat helyettesítő korlát is. De nem minden adatmodellezési korlát adható meg struktúrális módon.

Étéren fontos eltérések vannak az 1-3NF, a BCNF és a 4-5NF alakok között. A megfelelő normálforma-alakok a BCNF alakokkal bezárólag elérhetők szemantikailag egyértelműen is, mert a funkcionális függés egyértelmű hierarchiát jelöl ki. Nem így a többértékű és a kapcsolási függés, amelyek kimondottan az M:N-es viszonyokra vonatkoznak.

Mint láthattuk, a 4NF esetében külön korlátrelációt kellett bevezetni. Az 5NF pedig körkörös, alapjaiban redundáns relációmeghatározást igényelt. Az 1-3NF alakoknál az adatbevitel a viszonylag egyszerű létezési/hivatkozási korlátokkal ellenőrizhető. A negyedik és az ötödik normálformánál amit a vámon nyerünk, azt elveszítjük a réven. A szerkezeti átfedést kiküszöböljük, de ezzel korlátellenőrzési problémát generálunk. A bevitt procedurálisan kell ellenőriznünk.

A relációs koncepcióban megadott funkcionális, többértékű és egyéb függések, valamint az azokon alapuló normálforma szerkezetek és normalizálási eljárások igen komoly, matematikailag megalapozott, elvileg konzisztens tervezési eljárásokat, végeredményben optimális adatmodell-szerkezeteket ígérnek.

A SYDES-módszerben egyáltalán nem becsülöm le ezen matematikai módszerek hasznát. Sőt: senki sem fog találni ma olyan szoftvert, amelyik a normalizálási módszereket olyan pontosan követné és kihasználná, mint a SYDES-szoftver. Az értekezés ezen fejezete nem a relációs koncepciók ellen, hanem azok tökéletesítése mellett szól. Javaslataim tételesen a következők:

- Felül kell vizsgálni az 1-3NF és BCNF normálalakok belső teljességét. Így új és hasznos alakokra bukkanhatunk.
- A síkállomány koncepciója nem tartható, mert az alapcél torpedózta meg azáltal, hogy procedurálisan kell megadni természetes attribútum-csoportokat. A virtuális attribútum matematikailag bevezethető, indokolható.

- Fel kell ismerni, hogy reális mértékben összetett adatmodellezési feladat esetén az optimumot soha sem érjük el, legfeljebb csak heurisztikusan közelíthetjük.

- A normalizáció nemcsak matematikai fogalom, mint azt eredeti megalkotói hitték. A pusztán számszerű összefüggések belső szemantikai hibákra hívhatják fel a figyelmet. A normálformahibák és a normalizálás alapján homonimák és szinonimák sora küszöbölhető ki. A viszony kölcsönös: a szemantikai hiba utalhat a normalizálás szükségességére. A normálforma-koncepció alapján sajátos szemantikai képletek, adatszerkezetek is megalkothatók.

- Végül: Ha valaki a matematikai (normálforma) jelzést komolyan veszi, de más optimum-megfontolásokkal is párosítja, akkor a csak matematikai alapúnál lényegesen kedvezőbb szerkezethez juthat. Ezt bizonyította a normálformák gyakorlati elemzése.

A relációs gondolat kifejtését és értékelését ezzel nem zártam le. A B.függelék, ahol a kiterjesztett relációs modellt tárgyalom, további SYDES-megfontolásokat rögzít. A fejezetet ezzel a megjegyzéssel hagyom abba: minden relációs gondolat hasznos az adatmodellezés szempontjából, ha a matematikai formalizmus nem takarja el a szemantikai lényegét, hanem éppen hogy feltárja azt.

A. FÜGGELÉK: A MODELLEZÉSI TÉNYEZŐK VISZONYLAGOSSÁGA

A fogalmi modellezésben mindig is lesz szubjektív elem, hiszen a fogalomalkotást emberek végzik. A fogalmi modell pedig fogalmakat (pl.: vevő) ábrázol még absztraktabb fogalmak (pl.: egyedtípus) segítségével. A szubjektivitás hatása csökkenthető lenne, ha az adatmodellezésben nem uralkodna terminológiai káosz. Sajátosságnak (property) nevezi az egyik koncepció azt, amit a másik tulajdonságnak (attribute) hív. A fogalmi szinonimáknál már csak a homonimák zavaróbbak. Ha például nem ugyanazt értjük az 'egyed' fogalmán. A szabványosítási kísérletek ellenére [I31] a helyzet nem javul. Új koncepciók születnek. Például az [I13] is új fogalmak tucatjait vezeti be. Egyelőre csak annyit tehetünk, hogy megpróbáljuk pontosan meghatározni konstrukcióinkat.

Ezen a téren kell szembenéznünk a viszonylagosság kérdésével. Amint azt a [H32] filozófiai célú értekezésemben kifejtettem, a modellezési irányzatok nagyrésze a terminológia szintjén összekeveri a valóságot és annak tükörképét. A szakirodalom számos pontján előfordul effajta kitétel [I24, 227.o.]:

'... a valós világ egy adott objektumát néhány ember egyednek tekintheti; mások sajátosságnak; ismét mások összefüggésnek.'

Ha ez így lenne, akkor sohasem születhetne közösen érthető fogalmi modell! Rossz a fogalmazás. Arra viszont jó, hogy rámutassak a viszonylagosság vetületeire.

A modell elemeinek a meghatározásakor (3.1.3 alpont) a valóságról szólva valós jelenségekről, azok sajátosságairól és összefüggéseiről beszéltem. Az egyed, tulajdonság és kapcsolat kifejezéseket mindig ezek tükörképeként említettem. A fenti idézet mondanivalója első közelítésben így pontosítható: '... a valós világ egy adott jelenségét néhány modellező egyedként; mások tulajdonságként; ismét mások kapcsolatként tükrözhetik.'

Ennek a viszonylagosságnak nyilván van határa, amire az idézet nem utal. A 'cikk' fogalmat, önmagában senki se fogja kapcsolatként modellezni. Nálam a viszonylagosságnak vannak az egyértelműséget fokozó korlátai. Nem úgy, mint a Chen-modellben, ahol az egyed és a kapcsolat fogalma világosan nem elválasztott - [I33].

Az idézet logikája szerint a 'kék'-et tekinthetjük egyednek **vagy** tulajdonságnak. A SYDES logikában létezik a 'kék' nevű jelenség. Ezt tekinthetjük objektumnak **és** sajátosságnak is. Ennek megfelelően a 'kék' modellezhető egyed és tulajdonság reprezentánsokkal is. Nem teljesen szabad módon. A SYDES definíciók szerint az információkkal leírni kívánt jelenséget egyedként, a jelenséget leíró sajátosságot tulajdonságként modellezzük. Ha tehát a színeket, köztük a kéket, nem akarjuk információkkal leírni, akkor azok egyedként nem modellezhetők (!!!), legfeljebb csak tulajdonságként. (N.B.: időlegesen (!) meghatározhatók tulajdonság nélküli egyedek is.)

Viszonylagos fogalompáros az esemény és az egyed is, amennyiben ezek egymást metsző halmazokat takarnak. Ugyanis vannak olyan változási jelenségek (pl.: rendelés), amelyeket információval akarunk leírni. Ezeket egyidejűleg egyedként **és** eseményként modellezhetjük. Viszonylagos fogalom az egyed és a kapcsolat is. Ha egy összefüggést információval akarunk leírni, akkor azt egyedként, egyébként kapcsolatként modellezzük.

Eddig a viszonylagosság egyik vetületéről volt szó, amely a kategóriák összefüggéseit érintette. A második vetület a típusok és az előfordulások viszonyára vonatkozik. Evidens, hogy a Szabó István nevű jelenség, környezetfüggően, lehet a dolgozó, a beteg, az adózóalany stb. egyedtípusok előfordulásával reprezentált. Egyes koncepciók kizárják, hogy egy környezetben egy konkrét valós jelenségnek több konkrét reprezentánsa legyen. Ez nagyon merev felfogás. A reprezentáló egyedtípusok tulajdonságainak az átfedése fogja eldönteni, hogy pl. a szállítókat és vevőket célszerű-e egyetlen partner típusban rögzíteni. Ez a horizontális viszonylagosság, ami a tulajdonságsorok átfedésével kapcsolatos.

A viszonylagosság harmadik vetülete az absztrakció szintjére vonatkozik. Ha például dolgozó egyedtípust alkotunk, akkor jogosan merül fel a kérdés, hogy miért nem ember (magasabb szint) vagy férfi-

és nödológozó (alacsonyabb szint) egyedet tervezünk. Ez a vertikális viszonylagosság, ami az előfordulások átfedésével kapcsolatos.

Az absztrakció síkjának és szintjének a kiválasztásában is segít a **szerepnév** fogalma (lásd a specializációt és az egyedaltípusokat a 4.4.4 alpontban). Nem vitás, hogy a horizontális és a vertikális viszonylagosság egyértelműen összefügg és nincs pontos mérték alkalmazásukra. Vagyis arra, hogy mikor fontosabb tényező a tulajdonságok, mint az előfordulások átfedése.

Zárógondolatként visszaautalok még az egyed, tulajdonság, kapcsolat hármának a viszonylagosságára. A SYDES-ban az egyedek és azonosító tulajdonságaik szigorú 1:1 viszonyban állnak. Ezért mindig tekinthetjük úgy, hogy az egyedek ez a tulajdonság 'képviseli'. Ha más egyedben leíróként/kulcsrészként/szerepnévként megjelenik, akkor a két egyed közötti viszonyt is képviseli. Ezért lehet a 'kék' kapcsolat is.

B. FÜGGELÉK: A KIBŐVÍTETT RELÁCIÓS MODELL

A SYDES koncepciójának sok részlete hasonlít az ún. kibővített relációs modell részkonceptióihoz. Ezért tartom fontosnak, hogy ezt a modellt külön is bemutassam. A koncepciót az [I24] alapján tárgyalom, nem az eredeti [I43] szerint. A SYDES alapkonceptióját már a [H05] megírásakor felvázoltam. Ez eleve kizárja azt, hogy a SYDES alap gondolatánál az RM/T modellt felhasználtam. Az alábbiakban explicit módon úgy bizonyítom ezt, hogy felhívom a figyelmet a népszerűvé vált kibővített relációs modell hiányosságaira.

1.pont. Az RM/T most is nélkülözi a **kapcsolatot**, mint struktúrális kategóriát. Az ún. hivatkozási integritás ezt a hiányt nem pótolhatja, mivel csak az alárendelt reláció felől minősíti a kapcsolatot. Nem fejezi ki azt, hogy a fölérendelt létezhets-e alárendelt nélkül (opcionális). A hivatkozási integritás alulról implikál fokot, de felülről nem (nem fejezhető ki, hogy az alárendeltek száma maximum egy).

Az egy kapcsolatra vonatkozó kitételek, definíciók, korlátok stb. külön-külön fogalmazandók meg és egymással nem ellenőrizhetők keresztbe (fok, opcionális, terjedelem). Logikai paradoxon, hogy ezeket a korlátokat egyedhez kötik, mivel azok nem egy egyedet érintenek, hanem kettő viszonyát.

2.pont. A rendszergenerálta **egyedkulcs** féloldalas koncepció. Az [I24] kijelenti, hogy erre azért van szükség, mert néha az egyednek csak a létét kell rögzíteni; vagy nem található hozzá felhasználói elsődleges kulcs; az változhat, illetve lehetnek szinonimái. Ez az érvelés paradox és kezdetleges.

Paradox, mivel ütközik azzal a kitéttel, hogy az egyedkulcsot a felhasználónak nem illik ismernie. Hogyan rögzíthető ekkor az egyed pusztán léte? Kinek mit mond egy önmagában álló rendszergenerálta kulcs? Kezdetleges, mert a felhasználónak mindig van elsődleges kulcsa. Ha más nem, az egyed teljes tulajdonságsora az! Ha ez nem volna, akkor miképpen lehetne biztosítani a relációs alapelv

érvényesülését, ami szerint két relációs sor nem lehet azonos? Ha a rendszergenerálta kulcs a kulcs, akkor létezhetnek azonos sorok.

Evidens az is, hogy felhasználói hivatkozások nélkül nincs lehetőség két jelenség összekapcsolására. Ezen túlmenően: a szinonimák nem jelenthetnek valós gondot a hivatkozásban, ha ellenőrizettek. Végül: a koncepció féloldalas. A felsorolt hivatkozási problémák a doméjnekkel kapcsolatosan is felmerülnek, azoknak még sincsen 'P-kulcsa'.

Azt, hogy létezhet jó hivatkozási rendszer is, hadd mutassam be a SYDES hivatkozási lehetőségein keresztül:

. Minden tényező rendelkezik egy **belső technikai kulccsal**, amely minden tényezőre - nemcsak az egyedekre - egyedi. A tényezők valamennyi összefüggését ez hordozza. Ismerete a kezeléshez nem szükséges; alkalmazónak nem is lehetséges. A modellezett fogalmak és a kulcsok viszonya szigorúan 1:1 fokú. A modellben egyazon elem nem szerepelhet kétszer.

. Az elemeknek van, a viszonyoknak lehet **elsődleges neve**. Ez a név kategórián belül egyedi. Az elsődleges névhez tetszőleges számú szinonima tartozhat. A szinonimákra komoly megkötések-ellenőrzések vonatkoznak a SYDES-ban. A homonimák elemszinten eleve kizártak. Módszereim vannak a homonimák viszonszinten történő 'visszacsempészésének' a kizárására.

. Alternatív hivatkozási mód a rövid-név (kód). Erre érvényesek a névvel kapcsolatos szabályok, két kivétellel. A kódnak nem lehet szinonimája. A kódot nem kötelező az alkalmazónak megadnia. Ha nem teszi, a SYDES automatikusan kiadja.

. Végül: a viszonyokra természetesen úgy is lehet hivatkozni, hogy kijelöljük a kapcsolt tényezők nevét (rövid-nevét). Ugyanakkor az RM/T koncepcióban van egy olyan elem, amit a SYDES-ban is alkalmazok. Ezt a [H08]-ban fejtettem ki. Arról van szó, hogy a kulcsot az egyed helyettesítőjének tekintjük.

3.pont. A modell **féloldalas**, mert az egyedeket és a tulajdonságokat nem hálós összefüggésben, hanem a P-reláción keresztül hierarchikus

kapcsolatban szemléli. Ha az E-reláció egyetlen hivatkozást tartalmaz, akkor ezt várnánk a P-relációra, mint doméjnre is. Lenne viszont egy EP-reláció, amely a kapcsolatokat írja le (a jelenlegi P-reláció).

4.pont. Az RM/T kizárja az **összetett kulcsot**. Nos, így valóban nehéz felhasználói hivatkozást találni, de ez tautológia. A csoportok tárgyalásánál (4.2 pont) utalok e megoldás negatív következményeire a hivatkozások terén.

5.pont. A szemantikai gazdagság **egyértelműséghiánnyal** párosul az RM/T modellben. Ilyen a jellemző-egyed, amelyre a rendeléstételt adják példaként. Mivel ez a rendelést a cikkhez köti, nem látható be, hogy miért nem viszony-egyed.

A függelék lezárásaként hangsúlyozom, hogy a kibővített relációs modellt igen fejlett koncepciónak tartom. A fenti kritikával születését és lényegét határoltam el a SYDES-étől.

C.FÜGGELÉK: A KUTATÁS FŐBB ÁLLOMÁSAI

Ebben a függelékben röviden összefoglalom tudományos munkám főbb állomásait.

Az alapokat a Közgazdaságtudományi Egyetemen szereztem. Itt ismerkedtem meg a számítógéppel. TDK keretében itt készítettem első dolgozatomat egy gyár információs rendszeréről. Később (1971) ebből a munkából született egyetemi-doktori munkám. Még előbb (1968) kezdtem programnyelvet tanulni.

Az adatbázis-gondolat kettős gyökerű. 1973-ban kaptam megbízást a közüzemi szolgáltatások adatbázistervének a kidolgozására (FSZDV). Ebben az évben találkoztam először adatbázis-szakértőkkel (UNDP) és hamarosan megtartottam az első, ilyen irányú hazai tanfolyamot.

A kutatások után 1975-ben ENSZ ösztöndíjat kaptam az USA-ba, ahol rendszertervezéssel és adatmodellekkel foglalkoztam. További munkám szempontjából meghatározó volt az 1976-os brüsszeli adatbázis-konferencia (VLDB). Ekkor váltam az adatmodellezésen alapuló rendszertervezés elkötelezettjévé.

Ebben az évben megírtam az adatbázisokról szóló alapművem, ami Magyarországon átütő sikert aratott [H05] és a témának ma is felvezető, meghatározó munkája. A rendszertervezés új szemléletének a fontosságát ez a mű mutatta be, de még csak érintőlegesen. A munka a legjobb publikáció címét nyerte el.

1980-ra kettős áttörést könyvelhettem el. Megjelent a nagyméretű rögzítő adatmodellezési könyvem [H08] és annak picinyke, de fontos részleteit szoftverben is megvalósító munkám [H10]. Nyilván nem alaptalan büszkeséggel említem, hogy a SZIAM volt az első termék, amit az IBM hazánktól (és az egész szocialista táborból) megvásárolt.

Ehhez kapcsolódóan kell említenem az IBM/ESRI előadásokat [H18]. 1983-84 során az IBM európai kutatóközpontjában négy ízben tartottam előadássorozatot. A konferencia-jellegű szervezésben 14-18 előadó szerepelt esetenként. Mind a négy alkalommal a magyarországi előadót nyilvánították a legjobbnak...

Ezután minden affelé irányított, hogy a feldolgozásmodellezéssel is foglalkozzam. Így született meg az ÁDÁM & ÉVA adat- és eljárás-modellezési szoftver. Használatára itthon még nem érték meg a

feltételek. 1984-es vizsgálata során a nyugatnémet Bayerische Motoren Werke A.G. (BMW) az adatmodellezési eszközök közül az ÁDÁM & ÉVA-t találta a legjobbnak.

A hazai szerkezetváltás kényszerített arra, hogy áttérjek a mikro-gépekre. A csere eredménye a SYDES [H30], amely teljesen általánosított adat-, eljárás- és környezetmodellezési segédlet. Az [I12] kapcsán kivívta a legjobb külföldi előadás címét. 1989-ben Neumann-díjas előadást [H30] ért a téma.

D. FÜGGELÉK: KAPCSOLÓDÓ PUBLIKÁCIÓIM

- [H01] Esztó, Z. - Halassy, B. - Krupa, P. - Krupa, P.-né - Mészáros, T.-né – Tollár, I. Számítógépes információrendszerek tervezési és módszertani eszközei. H.B.: Döntési táblázatok. SZÁMOK, 1975.
- [H02] Data Bases and Top-to-Bottom Systems Design. Compcontrol '77, Warsaw, 1977.
- [H03] Rendszerszervezés adatbázison alapuló számítógépes feldolgozásoknál. 1-2. rész. SZÜT, 1976. 6-7.sz.
- [H04] Mogyelirovanyije dännüh i projektyirovanyije bázü dännüh. KGST Főkonstruktori Tanács 2.mb., Moszkva, 1976.
- [H05] Adatbázisok kezelésének alapvető kérdései. SZÁMOK, 1978. Második kiadás: SZÁMALK, 1982.
- [H06] Funkcionális függésen alapuló adatmodell. I. Neumann kongresszus, Szeged, 1979.
- [H07] A koncepcióális adatmodell egy lehetséges megfogalmazása. IV. Informatikai Vándorgyűlés, Szeged, 1979.
- [H08] Adatmodellezés, adatbázis-tervezés. SZÁMOK, 1980.
- [H09] Description of the SZIAM Data Model Design Aid. Third International Seminar on DBMS, Warsaw, 1980.
- [H10] SZIAM - Modellezési Kézikönyv, 1980.
SZIAM - Használati Kézikönyv, 1980.
SZIAM - Programozási Kézikönyv, 1980.
SZIAM/IBM - Application Design Guide, 1982.
SZIAM/IBM - Program Description and Operation, 1982.
- [H11] Halassy, B. (sorozat-szerkesztő) Számítógépes információrendszerek fejlesztése. SZÁMOK, 1980-81.
- [H12] Az adatmodellezés elvi alapjai. 1-3. rész. Információ-Elektronika, 1981. 1-3.sz.
- [H13] Vállalati rendszerek információs modellezése és az adatbázis tervezés. Kandidátusi értekezés, 1981.
- [H14] Az információs rendszerek alapfogalmai. SZÁMALK, 1982.
- [H15] SZIAM: A Data Model Design Aid. First National Computing Congress, Helsinki, 1982.
- [H16] SZIAM: A Data Model Design Aid. Specification and Design of Software Systems. Conference on Operating Systems, Visegrád, 1982.
- [H17] Az adatfeldolgozás fejlettségéről. Számítástechnika, 1982. 12. sz.
- [H18] Formal Data Base Design - Data Modelling. IBM-ESRI Workshop on

- Database Design, La Hulpe, 1983.
- [H19] ÁDÁM & ÉVA adat- és eljárásmodellezési segédeszköz. II. Neumann kongresszus, Székesfehérvár, 1983.
 - [H20] Adatmodellezés a rendszerfejlesztésben. SZÁMALK, 1983.
 - [H21] ÁDÁM & ÉVA adat- és eljárásmodellezési segédeszköz. Információ-Elektronika, 1984. 2.sz.
 - [H22] Mennyi kétszer kettő...? Az ANSI/SPARC adatbázis-architektúra felülvizsgálata. Információ-Elektronika, 1984. 5.sz.
 - [H23] Automata navigadgeneráló. Datumbazoj. Internacia Komputado '84, 1984. 3.sz.
 - [H24] Adatbázisprogramok tervezését segítő automatikus navigálás-generátor. Információ-Elektronika, 1984. 6.sz.
 - [H25] ÁDÁM & ÉVA - Tervezési Kézikönyv, 1984.
ÁDÁM & ÉVA - Programleírás és működtetés, 1984.
Lefordítva angolra is.
 - [H26] Kollektív rendszerfejlesztési módszer. Információ-Elektronika, 1985. 3.sz.
 - [H27] ADAM & EVA - A Data- and Processmodel Design Aid. Datenbank Entwurf. Ein aktuelles Problem beim Anwender? Data Bases - Theory and Application, Basel, 1986.
 - [H29] SYDES: A Method and Tool for Information Systems Analysis and Design. Comparative Review of Information Systems Design Methods (CRIS '88), London, 1988.
 - [H30] SYDES: Módszer és eszköz az integrált rendszerfejlesztéshez. IV. Neumann Kongresszus, Pécs, 1989.
 - [H31] Az alábbi vállalatok részére gyakorlati modellezési anyagokat készítettem, elméleti összefoglalóval:
1975. ARTISJUS 1983. PSZTI
1976. KÖTUKI 1984. BMW
1981. MÁJUS 1. 1985. CSŐSZER
1982. Posta Jármű 1986. MŰSZI
1983. OLAJTERV 1987. CHINOIN
 - [H32] Az információmodellezés ismeretelméleti problémái. MKKE Kézirat, 1979.

IRODALOMJEGYZÉK

- [I01] INFOTECH: Structured Analysis and Design. State of the Art Report. Infotech International, 1978.
- [I02] Hartmann, W., Matthes. H. & Proeme, A.: Management Information Systems McGraw-Hill, Amsterdam, 1968.
- [I03] Oktatási anyagok a következő intézetektől:
 - Computer Systems Methodology
 - Control Data Institute
 - DELTAK
 - Diebold
 - Honeywell Inc. ('BISAD')
 - International Business Machines
 - National Computer Centre stb., stb.
 - Ide értendők az általam meglátogatott tucatnyi (!) külföldi egyetem/kutató intézet munkaanyagai, jegyzetei, kiadványai.
- [I04] Kertész, J.-né és mások: Számítógépes információs rendszerek tervezése és dokumentálása (ARDOSZ '79) SKV, BP, 1979.
- [I05] Hice, G.F., Turner, W.S. & Cashwell, L.F.: SDM: Systems Development Methodology North-Holland, Amsterdam, 1974.
- [I06] IBM: Business Systems Planning: Information Systems Planning Guide IBM Form Number GE20-0527, 1975.
- [I07] Lundeborg, M., Goldkuhl, G. & Nilsson, A.: Information Systems Development - A Systematic Approach Prentice-Hall, U.S.A., 1981.
- [I08] Knuth, E., Halász, F. & Radó, P.: SDLA: System Descriptor and Logical Analyzer Az [I11]-ben jelent meg.
- [I09] Teichrow, D. & Hershey, E.A. III.: PSL/PSA: A Computer Aided Technique for Structured Documentation and Analysis of Information Processing Systems IEEE Trans. SE-J, 1977.
- [I10] Stoff, V.: Modell és filozófia. Kossuth, 1973.
- [I11] Olle, T.W., Sol, H.G. & Verrijn-Stuart, A.A. (eds.): Information Systems Design Methodologies: A Comparative Review. North Holland, Amsterdam, 1982.
- [I12] Olle, T.W., Verrijn-Stuart, A.A. & Bhabuta, L.: Computerized Assistance During the Information Systems Life Cycle. North Holland, Amsterdam, 1988.
- [I13] Az IFIP WG 8.1 1989. június 4-8. ülésének vitaanyaga. Kéziratok. Sesimbra, Portugália, 1989.
- [I14] MacDonald, I.: Automating the Information Engineering Methodology

Megjelent az [I12]-ben.

- [I15] Sernadas, A., Bubenko, J. jr & Olivé, A. (eds.): Information Systems: Theoretical and Formal Aspects North Holland, Amsterdam, 1985.
- [I16] E. F. Codd: A Relational Model of Data for Large Shared Data Banks CACM 13, No.6. 1970.
- [I17] ANSI /X3/ SPARC: Interim Report: Study Committee on Data Base Management Systems. ACM SIGMOD Newsletter, 1975.
- [I18] Astrahan, M.M. et al.: System R: Relational Approach to Database Management ACM TODS 1, No.2. 1976.
- [I19] Codd, E.F.: Further Normalization of the Data Base Relational Model. In: Data Base Systems, Courant Computer Science Symposia Series, 6. Englewood Cliffs, N.J.: Prentice Hall, 1972.
- [I20] Casey, R.G. & Delobel, C.: Decomposition of a Data Base and the Theory of Boolean Switching Functions IBM Research J. 17, No.5. 1973.
- [I21] W. W. Armstrong: Dependency Structures of Data Base Relationships. Proc. IFIP Congress, Stockholm, 1974.
- [I22] R. Fagin: Multivalued Dependencies and a New Normal Form for Relational Databases ACM TODS 2, No.3. 1977.
- [I23] Aho A.V, Beeri, C. & Ullman, J.D.: The Theory of Joins in Relational Databases ACM TODS 4, No.4. 1979.
- [I24] Date, C.J.: An Introduction to Database Systems Addison-Wesley, Massachusetts, 1981.
- [I25] Fagin, R.: The Decomposition Versus the Synthetic Approach to Relational Database Design Proc. 3rd International Conf. on VLDB, 1977.
- [I26] Kent, W.: Consequences of Assuming a Universal Relation ACM TODS 6, No.4. 1981.
- [I27] Delobel, C.: A Theory About Data in an Information System IBM RJ 964, 1972.
- [I28] Bernstein, P.A.: Synthesizing Third Normal Form Relations from Functional Dependencies ACM TODS 1, No.4. 1976.
- [I29] Nijssen, G.M. (ed.): Architecture and Models in Data Base Management Systems, North-Holland, Amsterdam, 1977.
- [I30] Rustin, R. (ed.): Data Models: Data Structure Set versus Relational Proc. ACM SIGMOD Workshop on DDAC, 1974.
- [I31] Van Griethuysen, J.J. et al. (eds.): ISO TC97/SC5/WG3: Concepts and Terminology for the Conceptual Schema, Preliminary Report ISO-ANSI, New York, 1981.
- [I32] Ld. az [I24] 5.kiadásának 6.fejezetét!
- [I33] Chen, P.: The Entity-Relationship Model: Toward a Unified View of

- Data. ACM TODS 1, No.1. 1976.
- [I34] CODASYL Data Base Task Group Rep.: ACM, New York, 1971.
 - [I35] Raver, N. & Hubbard, G.U.: Automated Logical Data Base Design Concepts and Applications IBM Systems J., 1977.
 - [I36] Van Griethuysen, J.J. (ed.): ISO TC97/SC5/WG3: Concepts and Terminology for the Conceptual Schema and Information Base ISO-ANSI, New York, 1982.
 - [I37] Senko, M.E., Altman, E.B., Astrahan, M.M & Fehder, P.L.: Data Structures and Accessing in Data-Base Systems IBM Systems J., No.1. 1973.
 - [I38] Hall, P., Owlett, J. & Todd, S.J.P.: Relations and Entities Az [I44]-ben jelent meg.
 - [I39] Petri, C.A.: Introduction to General Net Theory. In: Net Theory and Applications Springer, 1980.
 - [I40] Falquet, G. et al.: Concept integration as an Approach to Information Systems Design. Az [I12]-ben jelent meg.
 - [I41] Brinkkemper, S., Brand, N. & Moormann, J.: Deterministic Modelling Procedures for Automatic Analysis and Design Tools
 - [I42] Preeyanont-Takahashi, S.: SIAM: A Methodology for Systematic and Interactive Modelling between User and Analyst. Az [I12]-ben.
 - [I43] Codd, E.F.: Extending the Database Relational Model to Capture More Meaning. ACM TODS 4, No.4. 1979. Az [I12]-ben jelent meg.
 - [I44] Nijssen, G.M. (ed.): Modelling in Data Base Management Systems North-Holland, Amsterdam, 1976.
 - [I45] Vetter, M. & Maddison, R.N.: Database Design Methodology Prentice-Hall, New Jersey, 1981.

TARTALOMJEGYZÉK

0. ÖSSZEFOGLALÓ.....	3
0.1 A 'SYDES'-GONDOLATRÓL.....	3
0.2 A KUTATÁS KÖRÜLMÉNYEIRŐL.....	4
0.3 A TÁRGYALÁSMÓDRÓL.....	5
0.4 AZ ALAPVETŐ TÉTELEKRŐL.....	6
0.5 A HIVATKOZÁSI KONVENCIÓRÓL.....	8
1. A TERVEZÉSI MÓDSZEREK KORLÁTAI ÉS KELLÉKEI.....	9
1.1 AZ INFORMÁCIÓS RENDSZER LÉNYEGE.....	10
1.1.1 ADATOK ÉS INFORMÁCIÓK.....	11
1.1.2 INFORMÁCIÓS ESEMÉNYEK ÉS TEVÉKENYSÉGEK.....	11
1.1.3 FELHASZNÁLÓK ÉS ERŐFORRÁSOK.....	12
1.1.4 SZABVÁNYOK ÉS ELJÁRÁSOK.....	13
1.1.5 AZ INFORMÁCIÓS RENDSZER VETÜLETEI.....	13
1.2 INFORMÁCIÓS RENDSZERTERVEZÉS.....	14
1.2.1 TERV (PLAN) ÉS TERV (DESIGN).....	14
1.2.2 ELEMZÉS ÉS TERVEZÉS.....	15
1.2.3 TERVEZÉS, SPECIFIKÁCIÓ ÉS KONSTRUKCIÓ.....	16
1.2.4 TERV, TERVEZÉS ÉS DOKUMENTÁCIÓ.....	17
1.2.5 TERVEZÉS ÉS SZERVEZÉS.....	17
1.2.6 TERVEZÉS ÉS MODELLEZÉS.....	17
1.3 TERVEZÉSI SZEMLÉLETI TORZULÁSOK.....	18
1.3.1 ESZKÖZ-MEGHATÁROZOTTSÁG.....	18
1.3.2 FELADAT-KÖTÖTTség.....	20
1.3.3 DIVATORIENTÁCIÓ.....	22
1.4 A MÓDSZEREK KELLÉKEI.....	23
1.4.1 ALAPKONCEPCIÓ.....	24
1.4.2 STRUKTÚRA/TERMINOLÓGIA.....	25
1.4.3 MEGFOGALMAZÁSI MÓD.....	26
1.4.4 KRITÉRIUMRENDSZER.....	27
1.4.5 TERVEZÉSI ALGORITMUS.....	28
1.4.6 DOKUMENTÁCIÓS FORMA.....	28
1.4.7 TECHNIKAI TÁMOGATÁS.....	29
2. A SYDES-KONCEPCIÓ ALAPVETŐ VONÁSAI.....	31
2.1 AZ ALKALMAZÁSI MODELL TÉNYEZŐI ÉS FELÉPÍTÉSE.....	32

2.1.1	TÍPUS, ELŐFORDULÁS, HALMAZ.....	32
2.1.2	ELEM, VISZONY ÉS STRUKTÚRA.....	33
2.1.3	ALKALMAZÁSI MODELL - LEÍRÁS ÉS TERMINOLÓGIA..	34
2.2	A TERVEZÉSI MODELL TÉNYEZŐI ÉS FELÉPÍTÉSE.....	35
2.2.1	KATEGÓRIA, METASTRUKTÚRA, METALEÍRÁS.....	35
2.2.2	JELLEMZŐK ÉS ÉRTÉKEK.....	36
2.3	MODELL ÉS METAMODELL.....	38
2.4	A SYDES KERETMÓDSZER LÉNYEGE.....	40
2.4.1	FIZIKAI, LOGIKAI ÉS FOGALMI SZINT.....	41
2.4.2	LEKÉPEZÉSI, KÖRNYEZETI ÉS RENDSZER SZINT.....	43
2.4.3	TERVEZÉSI ORIENTÁCIÓK.....	46
2.4.4	A SYDES TERVEZÉSI KERET.....	47
2.5	A SYDES-SZOFTVER ÉS HASZNÁLATA.....	50
2.5.1	A TERVEZÉSI SEGÉDESZKÖZÖK FUNKCIÓI.....	50
2.5.2	A SYDES TERVEZÉSI LOGIKÁJA.....	52
3.	A SYDES RENDSZERMODELL FŐVONÁSAI.....	56
3.1	A SYDES FOGALMI ADATMODELLJE.....	57
3.1.1	AZ ADATMODELLEZÉS CÉLJAI ÉS KÖVETELMÉNYEI..	57
3.1.2	OPTIMUMKRITÉRIUMOK ÉS MODELL-IRÁNYZATOK....	61
3.1.3	A SYDES ADATMODELL ALAPKATEGÓRIÁI.....	63
3.1.4	A SYDES ADATMODELL ÖSSZEFÜGGÉSKATEGÓRIÁI...65	
3.2	A SYDES FOGALMI FELDOLGOZÁSMODELLJE.....	71
3.2.1	A FELDOLGOZÁSMODELLEZÉS CÉLJAI, IRÁNYZATAI	71
3.2.2	A SYDES FELDOLGOZÁSMODELLEZÉSI KATEGÓRIÁI..	73
3.2.3	A SYDES ADATMODELL-KATEGÓRIÁI.....	78
3.3	ALTERNATÍV TERVEZÉSI ALGORITMUSOK.....	80
4.	RELÁCIÓS NORMÁLFORMÁK A SYDES.....	82
	ADATMODELLEZÉSBEN.....	82
4.1	A RELÁCIÓS NORMÁLFORMÁK TELJESSÉGÉRŐL.....	83
4.1.1	A TELJESSÉGVIZSGÁLAT ALAPJA ÉS MÓDJA.....	84
4.1.2	SZOKATLAN FÜGGÉSKOMBINÁCIÓK.....	86
4.1.3	EGYÉB NORMÁLFORMA MEGFONTOLÁSOK.....	89
4.1.4	NORMÁLFORMA-TELJESSÉG ÉS A SYDES.....	92
4.2	TULAJDONSÁGCSOPORTOK AZ ADATMODELLBEN.....	93
4.2.1	CSOPORTOS AZONOSÍTÓ-TULAJDONSÁGOK.....	94
4.2.2	CSOPORTOS LEÍRÓ-TULAJDONSÁGOK.....	96

4.2.3 CSOPORTOK MODELLEZÉSE A SYDES-BAN.....	97
4.3 NORMALIZÁLÁSI ELJÁRÁSOK ÉS A SYDES-.....	99
KATEGÓRIÁK.....	99
4.3.1 ADATMODELL DEKOMPOZÍCIÓ ÉS SZINTÉZIS.....	99
4.3.2 A SYDES ELEMZÉSI MÓDSZERE.....	101
4.3.3 NORMALIZÁLÁS ÉS A SYDES-KATEGÓRIÁK.....	104
4.4 NORMÁLFORMÁK ÉS EGYÉRTELMŰSÉG.....	106
4.4.1 A NORMÁLFORMÁK ÉS ATTRIBÚTUM-HIVATKOZÁSOK	106
4.4.2 VIRTUÁLIS ÉS VALÓS NORMÁLFORMA-HIBÁK.....	110
4.4.3 A '@'-REDUNDANCIA ÉS A FELTÉTELES FÜGGÉS.....	113
4.4.4. SPECIALIZÁCIÓ ÉS EGYEDALTÍPUSOK.....	114
4.4.5 SZEREPNÉVEN ALAPULÓ SYDES-STRUKTÚRÁK.....	116
4.5 A NORMÁLFORMÁK HATÁSA AZ ADATMODELL.....	120
OPTIMUMÁRA.....	120
4.5.1 NEM-NORMALIZÁLT RELÁCIÓK (ONF).....	120
4.5.2 RÉSZLEGES ÉS TRANZITÍV FÜGGÉSEK.....	123
4.5.3 A KULCSTÖRŐ FÜGGÉS.....	124
4.5.4 MAGASABB NORMÁLFORMÁK ÉS EGYÉRTELMŰSÉG.....	127
4.6 RELÁCIÓK ÉS A SYDES - ÖSSZEFOGLALÓ.....	130
A. FÜGGELÉK: A MODELLEZÉSI TÉNYEZŐK.....	133
VISZONYLAGOSSÁGA.....	133
B. FÜGGELÉK: A KIBŐVÍTETT RELÁCIÓS.....	136
MODELL.....	136
C.FÜGGELÉK: A KUTATÁS FŐBB ÁLLOMÁSAI.....	139
D. FÜGGELÉK: KAPCSOLÓDÓ PUBLIKÁCIÓIM.....	141
IRODALOMJEGYZÉK.....	143
TARTALOMJEGYZÉK.....	146

AZ ÉRTEKEZÉS CÉLJA ÉS LÉNYEGE

Közismert tény, hogy az információs rendszerek tervezésére szolgáló módszerek, - az 'orgver', ha úgy tetszik -, fejlődése messze elmaradt a hardver és a szoftver rohamos fejlődése mögött. Számítástechnikai intézményeink még ma is alkalmazzák a szekvenciális állományra, köteget feldolgozásra és különálló felhasználókra alapozó (1968-as!) BISAD módszert [I03]!

Nem túlzás azt állítani, hogy a gyakorlatban a módszerek ezrei (!) léteznek. Hiszen szinte minden alkalmazónál (sőt: alkalmazásnál) más-más tervezési megoldásokat használnak annak dacára, hogy azok hibái már régen kiütköztek. A 'software-engineering'-nek nevezett szakma csak nem akar mérnöki jellegű tudománnyá válni...

Mitől és mennyiben rosszak a ma elterjedt információs rendszertervezési módszerek? Miért csak lassan kerülnek át a gyakorlatba az ezen a téren elméletileg sikeresnek bizonyult megoldások (pl. a normálforma-elemzés, a Petri precedencia vagy az egyed-életciklus koncepciója)? Miért nincsenek tervezési szabványok, mint a mérnöki tudományokban?

Dolgozatomban nemcsak az ilyen és hasonló kérdéseket firtatom, hanem határozott kiutat is mutatok abból, amit jogosan nevezek tervezési válságnak. Az értekezés már a téma méreténél fogva sem adhat végleges megoldásokat. Felhívja viszont a figyelmet a mai módszerek közös, elméletileg hibás gyökereire és kiszélesíti a tervezési diszciplína tudományos alapjait.

A tudományos munka módszere és körülményei

1970 óta foglalkozom a rendszertervezési módszerekkel. 1972-1982 között a témáért felelős szakmai vezető voltam a SZÁMOK-ban.

Mindennapos kötelezettségként kellett felkutatnom, értékelnem és bevezetnem tervezési eljárásokat. 1971-ben az ICL és a Honeywell

(BISAD) anyagát sajátítottam el. 1973-ban a CDC-vel szerződünk. Ekkor ismertem meg az INFOTECH, a DELTAK, a DIEBOLD, a CSM [I03] rendszertervezési módszereit.

Akkoriban a SZÁMOK képviselte e téren az egyetemi szintet. Ez főleg annak volt köszönhető, hogy nyugati egyetemekről mintegy száz szakértőt fogadhatott a UNDP projekt keretében. Többjük programját én szerveztem meg és velük közösen tartottam tanfolyamokat. Így az első nyilvános döntési táblázat és adatbázis-kezelés kurzust is.

1972 óta publikálok. Nézeteimet mintegy félszáz szakmai úton volt lehetőségem külföldi intézményekben megkonzultálni. Az 1975-ös féléves USA (UNDP) ösztöndíj mozdított el végleg az adatbázis-elmélet és az azon alapuló modellezés irányába. Ezt a tárgyat nemcsak az egyetemeken (Ann Arbor, Minneapolis, Columbia) ismerhettem meg, hanem az IBM kutatóközpontjában, az adatbáziskezelőt készítő MRI cégnél és a gyakorlatban, a Chase Manhattan bankban. Itt adatbázis-kezelő kiválasztását végeztük matematikai modellezési módszerrel.

1975-ben fogalmaztam meg először az adatmodellezésre épített rendszertervezés gondolatát. A modellezésen alapuló adatbázis-tervezés gondolata nem volt új. Viszont annak számított a koncepció kiterjesztése teljes tervezési módszerre. Ekkor ugyanis még a PSL/PSA típusú tervezési ideák uralkodtak [I09]. Ez idő óta tartottam előadásokat e témában konferenciákon. 1978-ban jelent meg első könyvem [H05], amely alapozó műnek számított. Az adatmodellezésről írt munkám [H08] több magyar felsőoktatási intézményben kötelező tananyag. 1982-ben modellezési módszeremet bemutathattam az IBM európai kutatóközpontjában (ESRI) is. Az előadás sikerét jelzi, hogy azt háromszor meg kellett ismételniem [H18].

Időközben áttértem az oktatásról a szoftverfejlesztésre. Sorban készültek el tervezési segédeszközeim. A SZIAM, amit normalizálási eljárása miatt az IBM is megvett [H15]. Az ÁDÁM & ÉVA, amit a BMW a legjobb, a kereskedelmi forgalomban kapható tervezési segédletként értékelte [H19]. Legutóljára a SYDES, amely külföldön [H29] és itthon is [H30] nagy sikert aratott.

Ezeket az eszközöket mintegy 20 gyakorlati esetben alkalmaztam. Így ritka szerencseképpen sikerült megvalósítanom az elmélet és a gyakorlat egységét. Az eszközök a [H08]-ban leírt elveket valósították meg. Majd a gyakorlat alapján magát az elméletet tovább

finomítottam. Ennek az eredményeit könyvben még nem publikáltam: ez az értekezés fogja össze a lényegét.

Kutatásaimat a nyíltság jellemezte. Eredményeimet gyorsan, ellenszolgáltatást nem várva tettem közzé. Az utóbbi szokatlan megjegyzést körülményeim indokolják. Ugyanis az utóbbi időkig bevételkötelezett, minimális pénzügyi támogatású, önellátó részleg vezetője voltam.

Az értekezés felépítése és a tézisek

Az általam választott téma igen szerteágazó és óriási szakanyagot ölel fel. Egy nem érvelő, a részleteket be nem mutató, csupán kalauzoló természetű tervezési módszertani kézikönyv terjedelme is 6-700 oldal. Nem áll tehát módomban, hogy mondanivalómat akár csak közelítőleg teljesen kifejtsem. Mindazt, amiben koncepcióm eltér a bevett módszerektől, nincs lehetőségem bemutatni. Ezért kettős megoldást alkalmazok, amit az alábbiak során ismertetek.

Az értekezés a mondanivalómat összefoglaló, a konvenciókat és alapfogalmakat ismertető, a felépítést indokló, tézist nem tartalmazó 0.fejezettel indul. A fejezet annyiban fontos, hogy abban fejtem ki koncepcióm szemléletét. Ezt legjobban a SYDES mozaikszó tükrözi.

A SYDES az angol SYstem DEsigner System kifejezés rövidítése, amit rendszertervező vagy rendszertervezési rendszernek lehet fordítani. A SYDES hármas lényegű. Tervezési módszertan; arra alapuló és azt támogató tervezési szoftver segédeszköz; a módszert és az eszközt megalapozó koncepció, mondhatni filozófia.

A ma elterjedt tervezési módszerek monolitikusak, azaz egyetlen, merev és zárt elképzelésre épülnek. Teljesen kizárják az egyéb elvek alkalmazását. Ugyanakkor és éppen az előzőek miatt nem teljesek, vagyis nem fedik le az információs rendszer valamennyi aspektusát.

Emiatt a gyakorlati felhasználók maguk kénytelenek e módszereket saját ízlésüknek megfelelően kiegészíteni. Ebből fakad az, hogy szinte alig van két szervezet, amely valóban azonos módszert alkalmazna.

Az általam kidolgozott módszer elébe megy ennek a problémának. A SYDES, kiejtés (szájdusz) szerint, oldalakat jelent. Módszertani keretében nem kizáróaknak, hanem egymást kiegészítőeknek

tekintem az általában ellentétesnek tartott nézeteket. Néhány példa: az irodalom [I24] ellentétesnek nevezi a normálforma dekompozíciót és szintézist - én egybeötvözöm azokat. Kizáróaknak tekinti az adat-, eljárás- és eseményorientált módszereket [I11] - én összhangot találok köztük. Vitában állt a relációs és hálós koncepció [I30] - én egybegyűrtam a kettőt.

Ekképpen a 0.fejezet, ha úgy tetszik, felfogható egyetlen átfogó, implicit tézisnek is. Azt sugallja, hogy a monolit tervezési eljárások lényegük szerint alkalmatlanok az integráltnak szánt információs rendszerek tervezésére.

Az 1.fejezet ezt a tézist segédítéletben (S1) úgy fogalmazza meg, hogy a ma széleskörűen használt tervezési módszerek súlyos módszertani korlátokat mutatnak. Segédítélet használók a következők miatt: A módszertani hiányosságokat 1972 óta ostromlom, tehát a gondolatot magaménak érzem. Véleményemet sokan osztják: az egyáltalán nem tudományos újdonság. Ugyanakkor az 1.4 pontban felvázolt implicit indoklást újszerűnek, úttörőnek tartom és tartották mások is [I12].

Az 1.fejezetben először munkameghatározást adok az információs rendszerre, illetve annak tervezésére. Az előbbiben az információs rendszert információk; az azokra vonatkozó információs események és tevékenységek; az előbbiekkal kapcsolatos felhasználók és erőforrások; továbbá a mindezeket szabályozó szabványok és eljárások szervezett együttesének nevezem. Így vázolom fel a rendszer három (adat, feldolgozás és környezet) vetületét. Egyben rámutatok arra is, hogy a tervezés nem azonos a szűken értelmezett szoftverfejlesztéssel.

A tervezés definiálása során kísérletet teszek a tervezéssel kapcsolatos fogalmak világos elhatárolására. (Ez, szerintem, eddig még nem történt meg.) Különbséget teszek a tervezés és a fejlesztés; a terv (design) és a kivitelezési terv (plan) között. Megvilágítom a tervezés és az elemzés; a tervezés és a specifikáció; a tervezés és a dokumentáció összefüggéseit. A modellezést magasfokúan tudatos és minőségre (eleganciára) törekvő tervezésként határozom meg.

Ezek után a jelzett segédítélet (S1) igazolása már egyszerű. A részletezésben rámutatok a fizikai (S11) és a logikai (S12) függőség gondjaira. Vagyis arra, hogy a módszerek eszközfüggőek és nem elméleti alapokon, hanem az ad hoc felhasználói szemlélet alapján struktúráltak. Nem jelentene gondot bebizonyítanom a mai módszerek

bármelyikéről, hogy e függéseknek kitettek. Főleg kedvezőtlennek tartom a divatból használt módszerek alkalmazását (S13). A kitekintés nélküli gyakorlati szakember ezeket olyan feladatok megoldásánál is felhasználja, amelyre lényegüknél fogva alkalmatlanok.

Az 1.fejezetben kimondottan eredeti az 1.4 pont, amelyben ismertetem a rendszertervezési módszerek 'kellékeit', elemeit. Ha ezt a módszertani kelléktárat összevetjük a mai tervezési eljárásokkal, akkor azokról azonnal kiderül, hogy egyoldalúak, hiányosak, merevek.

A hét kellék szerintem a következő: az információs rendszer alapkoncepciója; struktúrája; megfogalmazási módja/terminológiája (nyelv); optimumkritériumai; tervezési algoritmusai; dokumentálási formája és tervezési segédeszközei (amelyekben az előző hat elem megismétlődik).

Az 1.4 pont nemcsak ismerteti ezeket a kellékeket, hanem azt is elárulja, hogy a SYDES-konceptióban miként válnak ezek sokoldalúvá. A fejezet végére a SYDES-gondolat fölénye a hagyományos tervezési eljárásokkal szemben világossá válik.

A 2.fejezet a kellékek közül az információs rendszer immanens struktúráját és az arra épülő tervezési algoritmus-keretet mutatja be. Az alkalmazási (pl. bérszámfejtési) rendszer terve elem- és viszony-típusokból áll (ilyen típus pl. a dolgozó rekord és a törzsszám adattétel, amelyek egymáshoz kapcsoltak). Mondanivalómban új, hogy a típusfogalmat kiterjesztem az eseményekre és tevékenységekre, vagyis az információs rendszer feldolgozási vetületére is. Ezzel egy homogénen kezelhető alkalmazási rendszerstruktúrát kapunk.

Bevezetem a tervezési rendszer fogalmát és azt analóggá teszem az alkalmazásával. A metatípusokat kategóriáknak hívom. Ilyen kategória a rekord, a program, az adattétel stb. A mai tervezési rendszerek testreszabottak, amennyiben eleve adott kategóriákkal dolgoznak úgy, hogy ezek jellemzői (pl.: adathossz, programnyelv, rekord hosszjelleg stb.), sőt ezek lehetséges értékei is behatároltak. A SYDES lényegesen fejlettebb koncepció ebből a szempontból, mivel előre ad egy 'metakészletet' - felhasználható kategória, jellemző, érték halmazt -, de ugyanakkor lehetővé teszi e halmaz tetszőleges bővítését is. Vagyis a SYDES tervezési metakeret. Ismereteim szerint az első a világon.

A tervezési rendszer struktúrája kapcsán kell rámutatnom arra, hogy a mai módszerek a kategóriák természetes struktúráit eleve

eltorzítják (T21). Ezért, ha a módszer egyéb elemei hibátlanok is lennének, már akkor is eleve nem-kíváncsós függőségek keletkeznének. Pl.: a hagyományos módszerekben adattételt csak rekordtípushoz kötötten lehetett meghatározni hierarchikusan, holott e két kategória természetes viszonya hálós.

Az adatmodellezésben régóta ismertek a fogalmi, logikai és fizikai absztrakciós szintek. A tervező kompromisszum mentesen; a felhasználói, időleges követelményekkel terhesen; és eszközfüggő módon fogalmazhatja meg a rendszertervet. Rá kell mutatnom arra, hogy ezen szinteket a módszerek nem, vagy nem következetesen alkalmazzák (T22). Ennek részben az az oka, hogy még ma is sokan ragaszkodnak az ANSI/SPARC által javasolt [I17], logikailag eleve hibás (S221) konstrukcióhoz. A SYDES ezt a problémát feloldja. Ezen túlmenően - egyedi módon - a színhármast a feldolgozásvetületre is bevezeti, megalapozva az alább ismertetett keretet. A SYDES-ban a környezeti vetületet szintként is értelmezzük. Ezzel az információs rendszer három aspektusa homogén módon megfogalmazhatóvá vált.

A mai módszerek nem alkalmazzák konzekvensen az információs rendszertervek vázát alkotó vetületek és szintek makróstruktúráit.

Többek közt ennek a következménye az, hogy a tervezési algoritmusban a tényezőket nem tudják egyenrangúaknak, homogén kezeléseknek tekinteni. Ebből következően a mai módszerek adat-, esemény- vagy tevékenységorientáltak. Ez azt jelenti, hogy az egyik vetület döntő módon meghatározza a másikat; a másik vetületnek nincs önálló léte, optima; a tervezési algoritmus lényege az, hogy valamilyen tényezőket másokat meghatározó módon kell rögzíteni, be kell azokat 'fagyasztani'.

Az értekezésben kimutatom, hogy a befagyasztás nem lehet az integrált rendszerek tervezésének alapja (T23). Ismertetek egy új alapelvű tervezési módszert. A SYDES módszerkeret orientációmentes és az ún. 'fuzzy' tervezési logikára épült. Ebben a tervtényezők meghatározásának nincs eleve kötött sorrendje. A dokumentáció megkövetelt struktúrája biztosítja a terv teljességét. A 'fuzzy' alapelv nem új - azon alapulnak a prototípust alkotó módszerek. A SYDES-ban kimondottan új az, hogy eleve teljesítendő optimumkritériumokat állít fel. A tervezéssel nem lehet továbblépni ezek megsértése esetén. A

terv nem akkor készül el, ha a végső dokumentáció teljes, hanem akkor, ha teljes és optimális. A SYDES igen erős a kritériumok terén.

Nyilvánvaló, hogy az optimumkritériumok - amelyek az érthetőség, egyértelműség, teljesség, minimalitás és megvalósíthatóság kategóriáiba soroltak a SYDES-ban -, csakis számítógépes támogatás esetén érvényesíthetők. A korszerű fejlesztéshez a módszer és az eszköz rugalmas összhangjára van szükség. A SYDES erre vonatkozó megalapozó kísérletnek is tekinthető.

A 3.fejezet egyrészt expliciten szemlélteti a SYDES-módszer fogalmi modellezési kategóriáit, másrészt impliciten bizonyítja a tervezési keret elvének létjogosultságát. Azt, hogy a SYDES logikailag és fizikailag független tervet tesz lehetővé. Az első három pontban az adat-, a feldolgozás- és a kettőt összekötő adat-almodell kategóriáit, illetve azok összefüggéseit mutatom be. A negyedik pont tervezési algoritmus alapelveket ismertet és bizonyítja, hogy azok egymást kiegészítő, illetve egymással helyettesíthető változatok.

Eredetileg a fejezetben nem kívántam tételket megfogalmazni. Ez a rész a módszertani kellékek közül a struktúrát, illetve a terminológiát (nyelv) vezeti be, finomítja. A fejezet fogalmazása során egyre határozottabbá vált az a véleményem, hogy a jelenlegi módszerek sikertelenségének egyik alapvető oka az, hogy kategóriáik, struktúráik nem alkalmazás-orientáltak (T3). Gondoljunk csak arra, hogy pl. a relációs modell nem ismeri a csoport, ezen alapvetően természetes konstrukció fogalmát!

Indulásként ki kell jelentenem, hogy a módszeralkotók a modelleket szupertervezők játékának, elméleti szakemberek vitatárgyának, nem pedig a felhasználóval való kommunikáció alapvető eszközének tekintik (T31). A koncepcióteremtők vagy meghatározott fejlesztési környezetből kerülnek ki (pl. eleve reláció-orientáltak), vagy elméleti szakemberek, akik gyakorlati alkalmazással nem foglalkoznak. Az eredmény mindkét esetben ugyanaz: szemantikailag szegény alapelvek születnek (T32). Ebből olyan furcsaságok fakadnak, mint a csoportok említett tagadása vagy az adatok közötti származtatási összefüggések (precedenciák) struktúrális megadásának a hiánya (T33).

Az 1-3.fejezetek átfogó gondolatokat mutatnak be, bár a 3.fejezet felé haladva egyre konkrétan. A nyilvánvalóan új gondolatok dacára az értekezésnek ebben a részében nehéz volt tézisekben megfogalmazni a

mondanivalót. Mondhatnám úgy is, hogy e fejezetek horizontálisan, szélességében fogták meg a tervezés összetett diszciplináját. Illő és kíváncsi, hogy egy kicsit bepillantunk a részletekbe is. Ezért ízelítőként betekintést adok a SYDES koncepció mélységeibe is.

A dolgozat 4.fejezete jellegében, tárgyalásmódjában lényegesen eltér a korábbiaktól. Ez az a kettősség, amire már utaltam. Népszerűsége és tudományos kezelése miatt a relációs normálformák tervezési részletét választottam tárgyamul. Sajnálattal állapíthatom meg, hogy a témában megjelent több ezer publikáció ellenére a normálformáknak az információs rendszertervezésben betöltött szerepét félreértik (T4).

Ezt a megállapítást több részlettel is alá tudom támasztani. A normálformák sorát a szakemberek teljesnek jelentették ki.

Bizonyítom, hogy a normálformák sora belülről hiányos (T41). Ez azt jelenti, hogy az 5NF alaknál valóban nincs tökéletesebb, de az 5NF-en belül a függési kombinációkat logikátlanul és hiányosan definiálták.

Ez nem véletlen. Az általam kimutatott hiányzó függéskombinációk mindegyike a csoportokkal kapcsolatos. Bizonyítottam, hogy a csoport-konstrukció hiányában a relációs modell alkalmatlan a rendszertervezésre (T42).

A jelenlegi normalizálási eljárások vagy dekompozíció, vagy (kizáró vagy) szintézisen alapulnak. Bizonyítható, hogy egyik módszerrel sem alkotható optimális adatmodell. A kettő együttese heurisztikusan jobb közelítést ad, mint a kettő külön-külön (T43). A SYDES normalizálási algoritmusa kettős jellegű.

Az adatmodell egyik kritériuma az egyértelműség. A gyakorlat során elméletileg is igazolt érdekes összefüggésre bukkantam, közvetlenül a saját normálformákhoz (lásd: T41) kötötten. A normálformák nem csak matematikai konstrukciók. Alkalmazásuk az adatmodell szemantikai hibáit is feltárhatja (T44).

Az előző tézisnek van egy másik interpretációja, vetülete is. Mégpedig az, hogy az alapvető normálformák mai értelmükben szemantikailag szegényesek (T45). Így nem egyértelmű a periodicitás fogalma (S451); félreértelmezett a részleges és a tranzitív függés (S452); valamint a BCNF is (S453). Nem következetes a relációs koncepcióban, hogy egy tényt mikor kell struktúrában és mikor korlátban kifejezni (S454).

A 4.fejezet sok olyan új gondolatot tartalmaz, amit a relációs szoftver- és alkalmazástervezőknek érdemes lenne megfontolniuk. Az

egyik függelék egy filozófiai kérdést fejteget, a viszonylagosságát. A másik az annyira fontos hivatkozások (kulcsok és nevek) kérdését taglalja az I43] modellhez kapcsoltan. Publikációim [Hxx] jegyzéke és a munka leírása következnek. Irodalomjegyzékkel [Iyy] zárul a dolgozat.

A további kutatás irányai

A SYDES-ot nem eredménynek, hanem kezdetnek tekintem. Sok még a teendő az adatok és a feldolgozások belső összefüggéseinek a külön-külön és az együttesen való megfogalmazása terén.

A kutatást három irányban folytatom. Keresem az automatikus navigálás és programgenerálás további lehetőségeit [H24]. A ma adott fogalmi struktúrához (vö. 3.fejezettel) kutatom a logikait és a kettő közti leképezés helyes módját. Végül kísérletezek a szoftvertámogatás tökéletesítésével.

A két irodalomjegyzék a fontosabb saját és kapcsolódó publikációkat sorolja fel (nem zárt sorrendben). Az értekezés elkészüléséért nagyon sokaknak vagyok hálás.

Megboldogult szüleimnek, akik az igényességet nevelték belém. Tanárainknak, főleg Pogány tanár úrnak, akinek a logikai készséget és Smidt tanár úrnak, akinek az angol nyelv képességét köszönöm. Volt munkatársaimnak, főleg Matók Györgynek, aki először ösztönzött a publikálásra. Kollégáimnak, Lévai Máriának és Nagy Erzsébetnek, akik átfésülték az értekezést. Családom tagjainak, akik türelmükkel támogattak e nagy munkában. Végül pedig Annak, - e megjegyzéssel nem bántok senkit, nekem viszont ez a hitem -, Aki az egésznek értelmet ad.

Budapest, 1989. augusztus