

PETKOVICS IMRE

INFORMÁCIÓS RENDSZEREK ÉS ADATBÁZISOK

JEGYZET

SZABADKA, 2002.

TARTALOMJEGYZÉK

1. ELŐSZÓ	4
2. A MODERN SZERVEZETEK ÜGYVITELE	6
2.1. AZ ÜGYVITEL DINAMIKÁJA	6
2.2. AZ ÜGYVITEL HATÁSA A SZERVEZETEK SZERKEZETI FELÉPÍTÉSÉRE	8
2.3. ÜGYVITEL, KOMMUNIKÁCIÓ ÉS INFORMÁCIÓS RENDSZEREK	10
3. A RENDSZER FOGALMA ÉS KIALAKÍTÁSA	12
3.1. A RENDSZER FOGALMA ÉS MEGHATÁROZÁSA	12
3.2. A RENDSZER HATÁRAI ÉS KÖRNYEZETE	15
3.3. A RENDSZER FELOSZTÁSA – TAGOLÁSA	18
3.4. KETTŐS RENDSZERNÉZET	19
4. INFORMÁCIÓS RENDSZER – RÉGI FOGALOM ÚJ TARTALOMMAL	20
4.1. FOGALMAK ÉS FELFOGÁSOK	20
4.2. AZ INFORMÁCIÓS RENDSZER FOGALMA – MODERN SZEMLÉLETBEN	23
4.2.1. Adat, információ, tudás	24
4.2.2. Információs tevékenység és információs esemény	28
4.2.3. Információs erőforrások	29
4.2.4. Az információs rendszerek felhasználói	30
4.2.5. Szabványok és eljárások	31
4.3. AZ INFORMÁCIÓS RENDSZER VETÜLETEI	33
4.4. AZ INFORMÁCIÓS RENDSZER SZINTJEI	34
5. TECHNIKAI ESZKÖZÖK – HARDVER ÉS SZOFTVER.....	39
5.1. HARDVER	41
5.1.1. Számítógép-architektúrák	41
5.1.1.1. Masszívan párhuzamos feldolgozás – MPP számítógépek	41
5.1.1.2. Szimmetrikus multiprocesszoros rendszerek – SMP	42
5.1.1.3. Vektorarchitektúrák	44
5.1.1.4. Mainframek	44
5.1.1.5. Egyedi (Specifikus) architektúrák	46
5.1.2. Alaplapok	47
5.1.3. Processzorok	51
5.1.4. Táruk – memóriák	53
5.1.5. Multimédia	57
5.1.6. Bemeneti/kimeneti (I/O) egységek	61
5.1.7. Kommunikáció	67
5.2. SZOFTVER	71
5.2.1. Operációs rendszerek	72
5.2.2. Segédprogramok	73
5.2.3. Felhasználói programok	74
6. AZ INFORMÁCIÓS RENDSZER FEJLESZTÉSE	78
6.1. A RENDSZERFEJLESZTÉSI TECHNOLÓGIA FEJLŐDÉSE	80
6.2. INFORMÁCIÓS RENDSZERFEJLESZTÉSI ELVEK	83
6.2.1. Általános rendszerfejlesztési elvek	84
6.2.1.1. A háromszoros és háromszintű tervezés	84
6.2.1.2. A rendszerelemzés és a rendszertervezés viszonya	86
6.2.1.3. A strukturálási és iterációs elvek	87

6.2.1.4.	Felhasználó-orientáltság	88
6.2.1.5.	Modellezés, absztrakció, modularizálás	89
6.2.1.6.	Szabványosítás, dokumentáció és minőségellenőrzés	90
6.3.	AZ INFORMÁCIÓS RENDSZEREK FEJLESZTÉSÉNEK MÓDSZERTANA	94
6.3.1.	Alapkoncepció	95
6.3.2.	Struktúra és terminológia	95
6.3.3.	Megfogalmazási mód	96
6.3.4.	Kritériumrendszer	97
6.3.4.1.	Valóságghűség	98
6.3.4.2.	Érthetőség	99
6.3.4.3.	Egyértelműség	99
6.3.4.4.	Minimalitás	100
6.3.4.5.	Teljesség	101
6.3.4.6.	A kritériumok következetes, harmonikus alkalmazása	101
6.3.5.	Tervezési algoritmus	102
6.3.6.	Dokumentációkezelés	105
6.3.7.	Technikai támogatás	106
6.3.8.	Metaszabványok	107
7.	AZ INFORMÁCIÓS RENDSZER ADATVETÜLETE	108
7.1.	AZ ISMERET	108
7.2.	AZ EGYED	109
7.3.	A TULAJDONSÁG	110
7.4.	A KAPCSOLAT	111
7.5.	AZ EGYEDTÍPUS KÜLSŐ SZERKEZETE	113
7.5.1.	A kardinalitás (kapcsolat-fok) szemantikája	114
7.5.1.1.	Az M:N típusú kapcsolat-fok szemantikája	114
7.5.1.2.	Az 1:N típusú kapcsolat-fok szemantikája	117
7.5.1.3.	Az 1:1 típusú kapcsolat-fok szemantikája	119
7.5.2.	A rekurzív vagy visszamutató egyedviszony	121
7.5.3.	Gyönge egyedtípus	121
7.5.4.	Azonosító-függő egyedtípus	122
7.5.5.	Kapcsolattípusok közötti kapcsolatok	123
7.5.6.	Tranzitív szerkezetű modellrész	124
7.6.	AZ EGYEDTÍPUS BELSŐ SZERKEZETE	125
7.6.1.	Tulajdonságtípusok közötti függések	126
7.6.2.	Normálformák	129
7.6.3.	Az egyedtípus szerkezeteinek megfeleltethetősége	133
7.7.	ADATMODELL, ADATBÁZIS	134
7.8.	ISMERETKEZELÉSI MÓDOK	135
8.	AZ INFORMÁCIÓS RENDSZER FELDOLGOZÁSVETÜLETE	139
8.1.	AZ ISMERETEK MEGJELENÉSI FORMÁI	139
8.2.	BEMENET, RÉSZEREDMÉNY ÉS KIMENET	142
8.3.	AZ ADATBÁZIS ÉS A FELDOLGOZÁSVETÜLET	143
8.4.	A NAVIGÁLÁS KONCEPCIÓJA, LEKÉRDEZÉSEK ÉS JELENTÉSEK	147
8.5.	FELDOLGOZÁSTERVEZÉS	152
8.5.1.	A feldolgozás modellezése	152
8.5.2.	Feldolgozástervezés középső szinten	154
8.5.2.1.	Adatkezelési gráf	155
8.5.2.2.	Adatfolyam diagram	156
8.5.3.	Makrotervezés	160
8.5.4.	Mikrotervezés	166
8.5.4.1.	Döntési táblázatok	167
8.5.4.2.	Előzmény-háló	169
8.5.5.	Feldolgozás-konzisztencia	170

9.	AZ INFORMÁCIÓS RENDSZER KÖRNYEZETVETÜLETE	173
9.1.	A FELHASZNÁLÓI KÖRNYEZET	173
9.1.1.	<i>Felhasználói pszichológia</i>	173
9.1.2.	<i>Munka-pszichológia</i>	175
9.2.	A SZABVÁNYRENDSZER	177
9.3.	AZ INFORMÁCIÓS RENDSZER DOKUMENTÁCIÓJA	178
10.	FELHASZNÁLT IRODALOM	181

1. ELŐSZÓ

Manapság már általánosan elfogadott az a nézet, hogy a fizikai valóság három alapvető „alapanyaga” az anyag, az energia és az információ. Ezt a felfogást igazolják az élőlények is, és az ember által létrehozott szervezetek (szervezett rendszerek) is. A szervezetek nem tudnak fennmaradni információáramlás nélkül, sőt a fejlődéshez a szervezethezük folyamatos javítására, továbbfejlesztésére van szükség. Ezt a célkitűzést az ember teremtette rendszerek csak a külső környezetükben bonyolódó folyamatok és jelenségek megfigyelésével és az azokat leíró információk gyűjtésével, rendszerezésével és az azokhoz való alkalmazkodással valósíthatják meg. A rendszer stabilitása a benne lejátszódó folyamatok dinamikájának elemzésével becsülhető meg¹.

Az információt ma ugyanolyan erőforrásnak tekintik, mint a nyersanyagot, a munkaeszközöket vagy a munkaerőt: kulcsfontosságú szerepet tulajdonítanak neki a szervezetek életében illetve működésében. Fontosságát, szükségességét, egyszóval stratégiai szerepét minden tudományos és szakmai fórumon hangsúlyozzák. Ebből ered az igény az ismeretek (adatok, információk) felkutatására, elrendezésére, tárolására, kezelésére, feldolgozására és megjelentetésére. Új információk információs rendszerek segítségével is biztosíthatók. A modern információs rendszerek számítógépeken megvalósított szoftvertermékeket jelölnek. Az információs rendszerek fejlesztése bonyolult vállalkozás. Szerteágazó szakterületeket ölel fel, sok ember közös erőfeszítését feltételezi, akik a szakterületük alapos ismerete mellett még fogékonyak a mások problémáira, kellő tisztelettel viselkednek a problémákkal, kitartóak a problémák megoldásában és kommunikatívak. Az információs rendszerek tényezőivel sokan és sokat foglalkoztak a hetvenes évek elejétől: ekkor kezdték ugyanis komolyan tanulmányozni az információs rendszerek fejlesztését. Átfogó, teljes képet az információs rendszerek működéséről, tényezőiről, a tényezők ismérveiről és fejlesztéséről azonban, az irodalomban egy helyen sehol sem adnak. Ez a jegyzet kísérletet tesz az információs rendszerek általános szintű bemutatására, általános felépítésére, az információs rendszerek tényezőinek jellemzésére és a tényezők statikus és dinamikus összefüggéseinek bemutatására.

A jegyzetformába gyűjtött ismeretanyag elsősorban a Szabadkai Műszaki Főiskola másodéves Informatika szakos hallgatóinak készült, akik az »Információs rendszerek és adatbázisok« c. tárgy keretén belül ismerkednek az információs rendszerek és adatbázisok rejtelmeivel. A heti két óra előadás és két óra gyakorlat nagyon szűkös keretnek bizonyult a

¹ A.J.Lerner gondolatai szabadon fogalmazva.

tárgy ismeretanyagának átadására, ezért a jegyzetben sok minden nem kapott (kaphatott) helyet. Igyekeztem elsősorban a helyes szemléletet megfogalmazni és kialakítását hangsúlyozni, ezért az anyag jobbára elméleti jellegű². A gyakorlatokon a hallgatók konkrét, életből vett, egyszerűsített problémákat oldanak meg, konkrét eszközök (szoftver) segítségével, az előadásokon megismert eljárások alapján.

A leendő informatikus mérnököknek csak annyit mondanék kedvesinálóként a bevezetőben, hogy akkor választottak helyesen szakirányt, ha az ágazat vezérfonala (fő tárgya, témája: esetükben az informatika) egyben a hobbijuk is. És végül kívánom, hogy minden informatikus hallgatónak ne csak szakmája, de hivatása legyen az informatika!

A jegyzetben kétséget kizáróan akadnak kisebb-nagyobb hibák-hiányosságok (ha valamilyen, akkor ebben teljesen biztos vagyok). Az olvasatok számával a hibák száma is, természetesen, fogatkozik, de a rendelkezésre álló rövid idő alatt nem lehetett őket teljesen kiiktatni. Szívesen fogadok mindennemű és -szándékú megjegyzést³, ami a jegyzetben leírtakat jobbá, szebbé és hasznosabbá kívánja tenni.

Petkovics Imre, okl. villamosmérnök, magiszter,

Számítástechnika és informatika szakirány

peti@vts.su.ac.yu

Szabadka, 2001. július.

P.S.: A fordítás szerb nyelvről (mert ez a tananyag szerb nyelven előbb jelent meg) nem sikerült szolgálai, szó szerint pontosra, aminek külön nagyon örülök. Nem is vállalkoztam volna a tükörfordítás feladatára. Bár az ismeretanyag majdnem teljesen ugyanaz, a könyvnek mégis kicsit más, eltérő a lelkülete.

Szabadka, 2002. szeptembere.

² "Semmi nem olyan gyakorlati, mint egy jó elmélet." (Boltzmann)

³ Vitā vituperāre facillimum est, emendāre difficillimum. A hibát korholni a legkönnyebb, kijavítani a legnehezebb.

2. A MODERN SZERVEZETEK ÜGYVITELE

2.1. AZ ÜGYVITEL DINAMIKÁJA

A modern vállalkozások boldogulása a mindinkább teret hódító piacgazdaság útvesztőin nagyon nehéz. A mai, dinamikus világ sok buktatót és kockázatot rejteget a túlélést kereső cégeknek mint valaha. A világpiac globalizációs törekvései, a felgyorsult technológiai fejlődés és az Internet adta lehetőségek új utak, új ötletek, új üzleti logika felkutatására ösztönzik a vállalkozásokat, cégeket irányítóit a fennmaradás illetve fejlődés érdekében. A vállalatvezetők feladata, hogy olyan körülményeket biztosítsanak, amelyek lehetővé teszik termékeik illetve szolgáltatásaik minőségének folyamatos javítását a költségek csökkentése, de legalább szintentartása mellett.

»Meghatározó tényezővé válni és annak (vagy egyáltalán) megmaradni a piacon«, ez a mai vállalatok legfontosabb célkitűzése. Ez azt jelenti, hogy a termékek és szolgáltatások állandó minőségellenőrzése folyik az igényes cégeknél, figyelve az ágazati konkurrenciára, és egyáltalán a környező világra, hiszen csak így becsülhető meg a vállalat mindenkori helyzete és kilátásai. A környezet legfontosabb szubjektumai az alvállalkozók, a szállítók, a megrendelők, az ide vonatkozó törvények, előírások és szabványok, valamint (utolsóként, de nem utolsó sorban) a fogyasztók. A legjobbnak tűnő ötletek, legzseniálisabbnak vélt újítások is halva születnek, ha a piac (a fogyasztók) nem fogadja el őket. A piac működéséről (kereslet-kínálat alakulásáról), az ágazati szervezetek eredményeiről és a partnerekről tanúskodó adatoknak egyre nagyobb a becsülete a vállalatvezetők körében. Sőt, ezen adatok szerepe együtt növekszik magával a vállalattal, illetve a termékválaszték bővülésével. Az egyre bővülő adattömeg-igény csak a vállalkozás és környezete közötti kommunikáció élénkítésével és bővítésével, illetve a belső, szervezeti egységek közötti kommunikáció fokozásával érhető el.

A modern cégek ügyvitelét elsősorban a külső és belső környezetük változása alakítja. A szervezetek válaszreakciója (felelete) az egyes változásokra, illetve a válaszidő rövidsége határozza meg a cég eredményességét és üzleti sikerét. A mobilitás és a változásokhoz való alkalmazkodás dinamikája a siker szükséges feltételei közé tartoznak. Ez pedig gyors és jó minőségű kommunikációt (adatáramlást), valamint gyors és az igényeknek megfelelő adatfeldolgozást követel meg. Az ügyvitel fejlesztése céljából elemzik a termelést: vajon minden jól működik a termelési folyamatban, lehet-e javítani az eredményességen vagy a minőségen? Mivel lehetne gazdagítani a terméket illetve szolgáltatást? Hogyan lehetne új piacokat vagy vásárlókat szerezni? Hol vásárolni a nyersanyagot, hol termelni és hol értékesíteni az árut

figyelembe véve a földrészek, országok nyújtotta össz lehetőségeket (fizikai föltételek, jogi körülmények, munkaerő képzettsége, munkaszokása, bére, igényei, stb.) a termelés és szállítás költségeinek visszaszorítása (minimalizációja) és az eladási ár (olvasd haszon vagy profit) legkedvezőbb szintre hozása (maximalizása) mellett, nem megelégedve egy elfogadott minőségszint biztosításáról.

Ilyen szempontból vizsgálandó egy cég információs rendszere is: mennyire hűen követi és segíti a szervezetben zajló folyamatokat; milyen könnyen, vagy egyáltalán alaktható-e a megváltozott körülményekhez és milyen az alakítások költségvonzata; milyen szintű döntéseket segítenek az információs rendszer által előállított jelentések: stratégiai, taktikai vagy operatív (napi) döntéseket. Az információs rendszer biztosította adatok (információk) vajon elegendőek-e, pontosak-e, megfelelőek-e és időben érkeznek-e a döntések meghozatalának szempontjából? Az információs rendszer ma már több, mint egyszerű segédeszköze a vállalatvezetésnek, mindinkább az ügyvitel szerves részévé válik. A vállalatvezetéssel szembeni elvárások az ügyvitel menetét illetően a következő módon fogalmazhatók meg:

- a) az eredményesség folyamatos javítása a rendelkezésre álló erőforrások jobb elosztásával és ütemezésével,
- b) a cég meghatározó tényezővé váljon a piacon azokban az ágazatokban amelyekben érdekelt; működjön együtt az ágazati és konkurens cégekkel, amit csak a gyors és eredményes cégközi kommunikáció tud biztosítani és
- c) állandóan bővítse termékeinek és szolgáltatásainak jegyzékét ügyelve a fogyasztók igényeinek változására, valamint a termékek és szolgáltatások minőségére.

A fent leírt elvárások megvalósítása érdekében a cégvezetés az információs rendszerek alkalmazásától a következő segítséget várja el:

- 1) a pontos, kielégítő és időben érkező jelentések (adatok, információk) lehetővé tegyék a szervezet (cég) eredményes működését,
- 2) az információs rendszer dinamikusan alkalmazkodjon a piaci körülmények és igények, valamint a szervezetben történő változásokhoz egyaránt,
- 3) legyen képes a legkülönbözőbb kódolású (multimediális) adatok (információk) fogadására és feldolgozására, érkezzenek ezek az adatok bármely, a világon elfogadott (hardver és/vagy szoftver megvalósítású) kommunikációs csatornán keresztül, úgymint helyi hálózat (LAN, Intranet), külső – vezetékes vagy vezeték nélküli – hálózat (WAN, Internet, EDI, WAP) stb., és
- 4) csökkentse a fejlesztés és az ügyvitel költségeit¹.

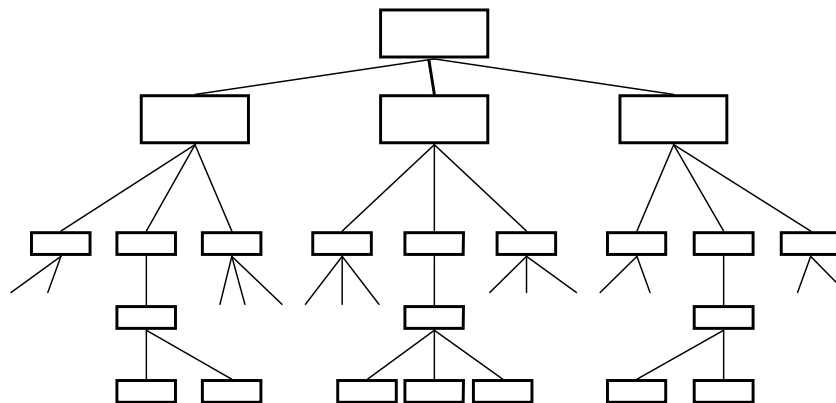
¹ Az irodalom ([22.], 152. oldal) útmutatása alapján.

2.2. AZ ÜGYVITEL HATÁSA A SZERVEZETEK SZERKEZETI FELÉPÍTÉSÉRE

A fejlődés üteme napról napra gyorsul a tudományok, a technológia és az üzleti élet minden területén. Ez fejlődés, a piaci viszonyok folytonos változása és a fogyasztók igényeinek állandó változása szinte naponta újabbnál újabb kihívásokat fogalmaznak meg a szervezetek (cégek) számára. A vállalkozások a régi (klasszikus, hierarchikus) szerkezeti felépítést kénytelenek elhagyni, ha meg akarnak felelni az új elvárásoknak.

A régebben jól bevált, többszintű vezetői réteggel ellátott hierarchikus felépítésű vállalat a munkamegosztást a funkcionális-szervezeti egységek szintjén valósította meg, ezeken belül pedig a hasonló munkák csoportosításával, majd pedig a képzettségi szint jelentette a munkamegosztás alapját. Az irányítási funkció a főnök-beosztott viszonyra épült. A közgazdászok egy része erről a szervezési formáról azt vallja, hogy legfeljebb csupán a „pozitív nulla” eredményszintre képes, ami a gazdaság egészének a pangásához vezet. Ennek a klasszikus szervezetfelépítésnek az ábrázolása látható az 2.1. ábrán egy gúla segítségével, többretegű vezetési-irányítási szint megjelölésével. Az ilyen felépítésű szervezet túlságosan lomha a mai, gyorsan változó körülményekhez való alkalmazkodásra, mert:

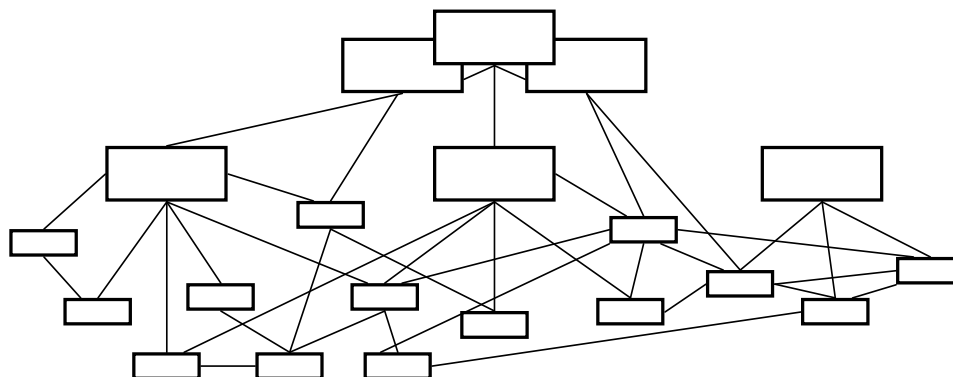
1. A vezetési-irányítási szintek elferdíthetik a legmagasabb vezetői szinten megfogalmazott utasításokat (információkat), és mindenképp késleltetik az utasítások eljutását a közvetlen végrehajtókhoz, ami a változásokra adott válaszokat késleltetik,
2. A munkákat és folyamatokat több szervezeti egység végzi, ezeknek az együttműködése pedig, a hierarchikus szervezés miatt nehézkes.



2.1. ábra

A klasszikus szervezés hiányosságait igyekeznek kiküszöbölni az új szervezési formák, amelyek kevés számú irányítási szinttel rendelkeznek. Az ilyen szervezésű vállalkozások

“lapított gúlával” ábrázolhatók sematikusán. Az új szervezési formákra inkább a “vízszintes kapcsolatok” jellemzőek, szemben a hierarchikus szervezésre jellemző “függőleges kapcsolatokra”. Az új szervezési forma “általános képe” a 2.2. ábrán látható. A két legismertebb



2.2. ábra

modern szervezési forma a project-orientált és mátrix szervezés.

A project-orientált szervezés alapján működő cégek a konkrét munkákhoz csoportokat (teameket) rendelnek, kiemelve ezzel a csoport tagjait a meglévő szervezeti struktúrából. Ezek a szervezetek célorientáltak, és ahogy a munkákhoz teameket, úgy a végrehajtókhoz feladatokat és határidőket rendelnek. Minden munkához költségkeret és egy, a határidők és a költségkeret betartásáért felelős személyt, project vezetőt jelölnek ki, aki irányítja a csoportmunkát, ütemezi, egyeztet és ellenőrzi a tagok munkáját és teljesítményét.

A mátrix szervezés ötvözi a meglévő (hierarchikus) szervezést a project-orientált szervezéssel. Ez a szervezési mód jól alkalmazható abban az esetben, ha a cégnek több, rövidebb lélegzetű, szűk határidejű, kisebb projecteket kell megvalósítania egyidőben (párhuzamosan). Itt nem jelennek meg új szervezeti alakzatok (teamek), mint a project-orientált szervezésnél, de a munkavégzőket többszörös feladattal (több projectben való részvétellel) is megbízhatják.

Az új szervezési modelleket néha heterarchikus strukturáknak is nevezik a szervezeti egységek közötti vízszintes kapcsolatok nagy száma miatt, ám a függőleges kapcsolatok is megtalálhatók itt, igaz a hierarchikus szervezéshez viszonyítva elenyésző mennyiségben. A modern szervezetekben a vezetők (irányítók) száma kicsi, hiszen csak így lehet őket (a szervezeteket) könnyen és eredményesen irányítani. Ez az előfeltétel a cég túlélésére, illetve fejlődésére valamint a fogyasztók igényeinek kielégítésére.

A mai szervezetekben a szervezési szintek szokványos száma három. A legfelsőbb vezetés (top management, vagy divatosan, az igazgató: CEO – Chief Executive Officer) képviseli a gúla csúcsát, ahol a cég stratégiáját dolgozzák ki, biztosítják (»beszerzik«) a munkát és a megfelelő

erőforrásokat. A második szint a középvezetők szintje, akik a cég stratégiájának a megvalósítását végzik. Ők szervezik a munkát a teamekben, kiosztják a feladatokat a teamtagok között, ütemezik és összehangolják a feladatok elvégzését, ellenőrzik az elvégzett feladatokat és motiválják a csoport tagjait.

Az új szervezésfajtákat alkalmazó cégeknél egy meghatározó jellegű tényre kell rámutatni a munkaszervezést illetően: nagyon gyakori a külső munkatársak alkalmazása a nagyobb munkák (projectek) jól körülírt feladatainak megoldására. A külső munkatársak alkalmazásának meghirdetése ma főképpen az Interneten történik, mi több, a jelentkezések, a munkafeladatok megfogalmazásai, a szerződéskötések, a kész munkák (feladatok) benyújtása és a munka kifizetése is így (az Interneten keresztül) bonyolódik.

2.3. ÜGYVITEL, KOMMUNIKÁCIÓ ÉS INFORMÁCIÓS RENDSZEREK

A mai szervezetek ügyvitele elképzelhetetlen a modern kommunikációs eszközök használata nélkül. A »Gondolkodom, tehát vagyok!«² szólás a mai cégekre vonatkoztatva a »Kommunikálok, tehát vagyok!« alakot ölti. A szervezet sikeressége ugyanis ma főleg az adatcsereinek minőségétől, mennyiségétől és gyorsaságától függ, ha mindez egy jól átgondolt és megvalósított információs rendszerrel párosul (persze a kiváló minőségű termékek és szolgáltatások megléte alapértelmezésnek tekintendő). A szükséges adatok a szervezethez nagyon változatos és eltérő formában érkeznek a rendelkezésre álló különböző kommunikációs csatornákon keresztül. A feldolgozásuk és a szükséges jelentések elkészítése az információs rendszer és a kisegítő-karbantartó személyzet feladata. A cél az igazgatási-döntéshozó szint gyors ellátása a legfontosabb információkkal (jelentésekkel) a vezetői szintnek legmegfelelőbb, leggyorsabban felhasználható alakban.

A kommunikációs rendszereket (hálózatokat) is a számítógépek, vagyis a számítástechnika működteti. Kézi központok, amelyekben a kapcsolatteremtést, nyilvántartást, naplózást, üzenetátadást és egyéb kedvezményeket élő munkaerő végzi, már nem léteznek. Mindezeket és sok más egyéb lehetőségeket ma már számítógépek (mikroprocesszoros rendszerek) valósítják meg. A helyi, városi, regionális és nemzetközi számítógép-hálózatokat is maguk a számítógépek irányítják speciális szoftverek segítségével.

A kommunikáció fejlettségét ma már egyértelműen a számítástechnika fejlettsége szabja meg. A számítógépeknek talán csak a hálózatok digitalizására, a vezeték nélküli adatátvitelre és a

széles spektrumú adatátvitel megvalósítására nincs közvetlen és meghatározó befolyásuk, egyenlőre. A nagyspektrumú átvitel lehetővé teszi a nagy adattömegek eleddig elképzelhetetlenül nagysebességű adatátvitelét (ez a kép és hanganyagok valamint az animációk és filmek hálózatokon keresztüli utaztatását teszi lehetővé).

A kommunikációs technológia és a számítástechnika közösen fejlesztik az üzenetek küldését és fogadását a klasszikus és mobil számítógépek és egyéb mobil eszközök között (E-mail), esetleg automatikusan is (EDI – Electronic Data Interchange, ami szabványos formátumú dokumentumok számítógépek közötti közvetlen cseréjét jelöli), de ilyen fejlesztés a WWW (World Wide Web), a Gopher vagy WAP típusú adathozzáférési alkalmazás is (ez utóbbi a mobiltelefonok elengedhetetlen velejárója lesz már holnap), ezenkívül ide tartoznak még az Internetes csevegésre szolgáló különböző szoftverek (IRC, telefonálás az Interneten keresztül, a hang- és képátviteli szolgáltatások, stb).

Ezekre az alaplehetőségekre aztán csakhamar egyéb kényelmi szintet növelő alkalmazások is készültek, bankszámla-ellenőrzés, banki átutalások Interneten keresztül, távvásárlás, távoktatás, távmunka, sőt távgyógyítás (teleshopping, teleteaching, teleworkig, telecooperation, e-business, e-banking, e-hospital stb.). Az ilyen alkalmazások száma napról napra bővül.

A felsorolt alkalmazások sajátosságok, ha másért nem, hát az Interneten való jelentésmegjelenítés miatt. A feldolgozás azonban nagyrészen a régi, megszokott stílusban bonyolódik. Éppen ezért még az ilyen alkalmazások számára készült információs rendszerek tervei sem kell, hogy jelentősen eltérjenek a szokványos rendszertervektől. A jegyzetben foglalt ismeretek tehát a Web alkalmazások fejlesztésénél is alkalmazhatók.

² Cogito, ergo sum!

3. A RENDSZER FOGALMA ÉS KIALAKÍTÁSA

Az ember évezredek óta fogalmakat alkot és alakít át a körülötte lejátszódó eseményekről, jelenségekről és tárgyokról (objektumokról). Az ember, azonban, elsősorban társadalmi lény¹, épp ezért ugyanazt a nyelvet (ugyanazokat a közösen elfogadott kifejezéseket kell használnia a közösen kialakított fogalmakra) kell beszélnie mint embertársának akivel meg akarja értetni magát. A rendszer kifejezés (fogalom) használata ma nagyon divatos az élet minden területén. Ez a szó, mintha tiszteletet szerezne a használójának vagy használója mondandójának, pedig nem ritka a téves alkalmazása sem.

Kétségen kívül kijelenthető, hogy az élet valamennyi dolga bonyolult. Ezért nem is kell csodálkozni, hogy egyre gyakrabban használjuk a rendszer kifejezést: minden összetettebb dologra alkalmazható ez a szócska. Magának a fogalomnak a behatárolása azonban éppen emiatt nehéz, hálátlan, majdnem lehetetlen. Mivel azonban az információs rendszer is a rendszerek kategóriájába tartozó fogalom, meg kell ismerkednünk a rendszernek, mint a legáltalánosabb filozófiai fogalomnak a lényegével.

3.1. A RENDSZER FOGALMA ÉS MEGHATÁROZÁSA

A szakirodalom nagyon sokféle meghatározást (definíciót) rejteget a rendszer lényegének megfogalmazására. Egyesek nagyon lakonikusak (rövidek), és csak a rendszer legfőbb(nek vélt) tulajdonságait hangsúlyozzák, mások részletezőek és hosszúk, a teljesség átfogásának igényével születtek. A rendszer első meghatározását a magyar származású kanadai biológus, Ludvig von Bertalanffy jelentette meg a “General System Theory” c. munkájában, az ötvenes években, amely a következőket tartalmazza a rendszerről:

“A rendszer egymással összefüggő elemek halmaza (komplexuma)”².

A hatvanas években B. Langefors az információs rendszerek tervezésének formális eljárásait dolgozta ki, és munkáiban a rendszer következő általános definícióját adta:

“A rendszer olyan objektumokból épül fel, amelyeket a részeinek tekintünk, és amelyek valamilyen módon kapcsolatban állnak egymással..”³.

¹ Sokmindent tud nélkülözni az ember, csak más emberek társaságát nem – szabadon, L. Bjorne alapján [16.]

² Forrás: [22.], 26. oldal

³ Forrás: [4.], 173. oldal

A legteljesebben és legáltalánosabban talán Zalai Béla fogalmazta meg a rendszer fogalmát “Allgemeine Theorie der Systeme” c. 1914-ben mejelent tanulmányában:

“Minden, ami egyáltalán elgondolható, szerkezetét tekintve rendszer. Az 'elgondolható' azonban éppúgy jelent 'objektumot', mint 'szubjektumot', 'tárgyat' éppen úgy, mint 'tétélezést', 'adottságot' ugyanúgy, mint 'felfogásmódot', röviden mindent, ami mint élmény, vagy mint a tudományos elvonatkoztatás terméke, bizonyos önállóságra tett vagy tehet szert.”⁴

A későbbiekben nem születtek ennél “kézzelfoghatóbb” definíciói a rendszer fogalmának, és ez nem igénytelenséget vagy lazaságot takar a mai kutatók részéről, hanem a feladat bonyolultságát. Dr Halassy Béla azt állítja, hogy:

“A rendszer egymással összefüggő tényezők szervezett együttese”⁵.

Dr Raffai Mária a rendszerről alkotott képét a következő meghatározással írja le:

“A rendszer meghatározott struktúra szerint egymással összefüggő és kölcsönhatásban lévő, egymással és a struktúrákkal összhangban lévő elemek együttese.”⁶.

B.Djordjević a “Vodoprivredni sistemi” c. monográfiában három rendszer-definíciót ad, a rendszer fogalmának teljesebb megvilágítása érdekében. Alább ez a három meghatározás következik.

A rendszer egy bizonyos struktúrát jelöl, amelybe bizonyos ideig anyag, energia és információ alakú bemenetet töltenek, és amelyikből egy másik időszakban anyag, energia és információ alakú kimenet nyerhető.”⁷.

A rendszer valamilyen kölcsönhatás alapján összetartozó elemek halmazát jelenti. A valós rendszereknél az elemek valamilyen erőforrás valamilyen átalakítására szolgáló objektumokat jelentenek a kapcsolatok pedig tömeg-, energia- és információszállító csatornákat.”⁸

A rendszer az elemeknek egy teljeskörű gyűjteménye, amelyekben az elemek valamilyen kapcsolatban állnak egymással, így valósítva meg egy közös célt. A rendszer és környezete között a kölcsönhatás folyamatos.”⁹

⁴ Forrás: [31.], 6. oldal

⁵ Forrás: [11.], 16. oldal

⁶ Forrás: [22.], 26. oldal

⁷ Forrás: [6.]

⁸ Forrás: [6.]

⁹ Forrás: [6.]

Összefoglalva az előzőkben leírtakat megállapíthatjuk, hogy a rendszer fogalmának megfogalmazásakor olyan általános leírást kell adni, amely minden valós és képzelt rendszert jól jellemez és ugyanakkor nem zár ki egyetlen rendszernek tekinthető dolgot sem. Ez esetben azonban csak igen általános ismérveket rögzíthetünk a definícióban:

- a rendszer az össz létező valós és/vagy képzelt elemek halmazából bizonyos megfontolásból elkülönített részhalmaz, amelyet egészként, összetartozóként szemlélünk és kezelünk,
- a rendszer elemei között elkerülhetetlen a kölcsönhatás jelenléte a közös cél (a rendszer céljának) elérése érdekében,
- a rendszer állandó kapcsolatot tart fenn a környezetével és
- a rendszerben a három alapvető létkategória (anyag, energia és információ) átalakulása illetve a környezettel való cseréje bonyolódik.

Az elemekről és kapcsolataik jellegéről azonban nem sokat tudunk, illetve mondhatunk, mert minden további részletezés már olybá szűkítené a rendszer fogalmát, hogy bizonyos rendszerkategóriákat eleve kizárnánk, már a definíció megalkotásakor illetve elfogadásakor.

Nehéz a már jól átgondolt és sokak által elfogadott meghatározásokból egy általánosabbat és jobbat létrehozni, de próbáljuk meg átfogni az eddigi definíciókban szerepelteket:

A rendszer a valós és/vagy képzeletbeli elemek olyan (egységbe, egészbe) rendezett halmaza, amelyben minden egyes elem az összes többivel meghatározott (meghatározható) kapcsolatban van, létrehozva így egy struktúrát, amely az elemnek a létezését és szükségességét is igazolja. Minden rendszer valamilyen cél megvalósítása érdekében jött létre, még akkor is, ha ezt a célt csak nehezen, vagy egyáltalán nem tudjuk behatárolni.

Mivel a rendszer tényezői (az elemek) jellegüket, viselkedésüket és természetüket illetően nagyon különbözőek lehetnek, és ez a megállapítás a tényezők közötti kapcsolatokra is teljes mértékben érvényes, a rendszer és környezete közötti kapcsolatok ugyancsak sokfélék. Ebből kifolyólag a rendszereknek nagyon sokféle osztályozását ismerjük [6, 22]. Az információs rendszerek fejlesztése szempontjából a rendszerek csoportosítása másodlagos, viszont a rendszerstruktúra fogalmát nem kerülhetjük meg.

A fenti definíciók már lényegében tartalmazzák a rendszerstruktúra fogalmát, amelyet a rendszer tényezői és kapcsolataik képeznek. A rendszerstruktúra statikus és dinamikus szempontból vizsgálható. A statikus rendszerstruktúra kialakításának fogalma alatt a rendszer tényezőinek (elemeinek) és a tényezők közötti kapcsolatok (ok-okozati összefüggések, vagyis a

kapcsolatok logikai függésének) meghatározását értjük a rendszer egy konkrét (statikus) állapotában. A dinamikus rendszerstruktúra azokat az elem- elemcsoport- illetve rendszerrészműveleteket öleli fel, amelyek a kitűzött cél eléréséhez vezetnek.

Bennünket a továbbiakban csak azok a rendszerek fognak érdekelni, amelyeket több ember közös erőfeszítése, gondolkodása¹⁰ hív vagy hívott életre azzal a szándékkal, hogy egy konkrét, közvetlen célt megvalósítsanak velük. Az ilyen rendszerek közös megegyezések, konvenciók szerint készülnek.

A rendszer fogalma nem abszolút fogalom ("Minden relatív, csak a fény terjedési sebessége abszolút – állandó – egyenlőre" – szállóige – forrása ismeretlen). Mindenki megfogalmazhat magának tetszése szerint rendszereket. Mi több, érdekeitől függően ugyanaz a személy ugyanarról az objektumról (dologról) többféle rendszert is megalkothat. Példaként elgondolkodhatunk arról, hogy egy konkrét gépkocsiról milyen rendszert alakít ki a tulajdonos, a bádigos, a villanyszerelő (pardon: gépkocsi villamossági szerelő), a fényező és mondjuk a mechanikus. Az egyes rendszerek esetleg nem is hasonlítanak egymásra, pedig ugyanaz a gépkocsi a megfigyelés tárgya!

A rendszerek relatív voltáról B. Langefors így vall:

Egyrészt minden rendszer, amelyet környezeti hatások érnek, a környezetét is felölelő nagyobb rendszer alrendszerének tekinthető, másrészt, a vizsgált rendszer minden elkülöníthető része potenciális rendszerként viselkedik.¹¹

A megfigyelt rendszer és a környezete között fellépő kölcsönhatást felfoghatjuk úgy is, mint a rendszerünk és más rendszerek között jelentkező kölcsönhatás. A kölcsönhatásban lévő rendszerek ekkor egy másik, nagyobb rendszert alkotnak. Másrészt, mivel nincs megszabott határ a rendszer nagyságára nézve, bármely rendszer jól megfogalmazható részét tekinthetjük rendszernek.

3.2. A RENDSZER HATÁRAI ÉS KÖRNYEZETE

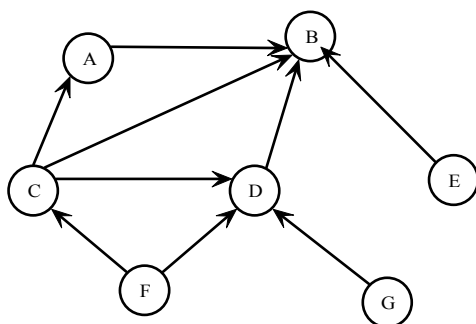
A rendszer vizsgálatát csak a pontos behatárolása után lehet megkezdeni. A rendszer lényegét pontosan meg kell fogalmazni és meghúzni a határait. A rendszer határainak kijelölése szorosan összefügg a rendszerleírás módjával. Két alapvető rendszerleírási módot alkalmaznak a gyakorlatban:

¹⁰ Az értelmes cselekedetekhez nem elég egyetlen fej (ész). Forrás: [16.], F.I. Dosztojevszki alapján, szabadon.

- a grafikus és
- a numerikus leírási módot.

A numerikus rendszerleírást a technika területén (különösen az irányítástechnikában) alkalmazzák leginkább. A grafikus leírási mód nem egységes: más megegyezések érvényesek a természettudományok és ismét mások a humán tudományok területén.

A grafikus leírás a műszaki tudományok területén a rendszer definíciójából indul ki, mégpedig abból a kitételből, hogy a rendszert egymással kapcsolatban álló tényezők alkotják (3.1. ábra). Így a rendszer grafikus képe tulajdonképpen egy irányított gráf, amely segítségével listaszerkezetek (a tényezők – elemek egymásutánisága szűrhető le : melyik elem melyiket követi, és melyik elem melyiket előzi meg) alakíthatók ki. Ilyen módon a rendszer statikus felépítését (struktúráját) mutatjuk be. A rendszer és környezete között zajló kölcsönhatást a 3.2. ábrán látható módon ábrázolják.

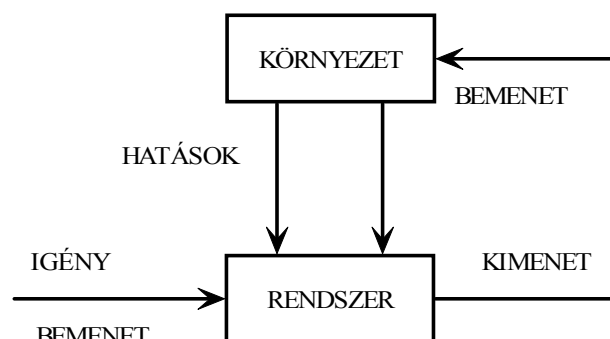


3.1. ábra

A humán tudományokban és az informatikában a rendszereket gyakran körök formájában ábrázolják. A körvonal ugyanis a valós világot két részre osztja: a számunkra fontos részre – ez a megfigyelt rendszerünk – és a számunkra kevésbé fontos részre – ez a rendszerünk környezete. Az életben minden mindennel összefügg, mégis, a teljes fizikai valóságot lehetetlen (ha lehetne is, értelmetlen lenne) egyetlenn rendszerbe gyömöszölni. Valahol tehát vágni kell, pontosabban szólva meghúzni a leírni kívánt rendszer határait, kihangsúlyozva ezzel azokat a tényezőket (elemeket), amelyek bennünket különösen érdekelnek. A rendszer fejlesztése előtt elemzés zajlik, és az elemzés első lépése éppen a rendszer meghatározása (milyen kifejező ez a szó: a határok meghúzását is magában rejti!), illetve elemeinek (tényezőinek) megjelölése. A határ

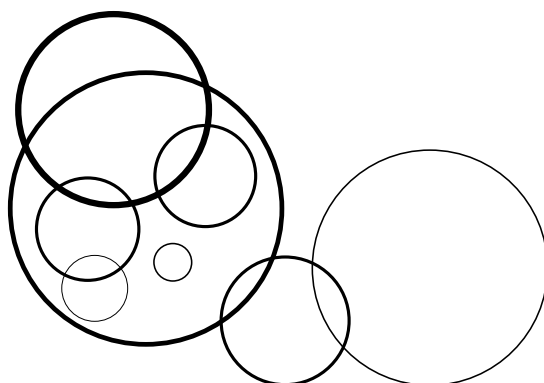
¹¹ Forrás: [4.], 174. oldal

berajzolásával a rendszer lényegét fogalmazzuk meg. A rendszerek egymáshoz való viszonyát a 3.3. ábra szemlélteti..



3.2. ábra

A rendszerhatárok minősítik az elemeket (tényezőket): a határon belül a megfigyelt rendszer tényezői helyezkednek el kapcsolataikkal együtt, a határon kívülre a környezet elemei szorultak. A környezeti elemek és rendszerelemek közötti kapcsolatokat elemezni kell: a szempontunkból jelentéktelen kapcsolatokat meg kell szüntetni, de a fontosak megmaradnak. Ezeken a környezet felé megmaradt kapcsolatokon keresztül bonyolódik a rendszer és környezete között a kölcsönhatás.



3.3. ábra

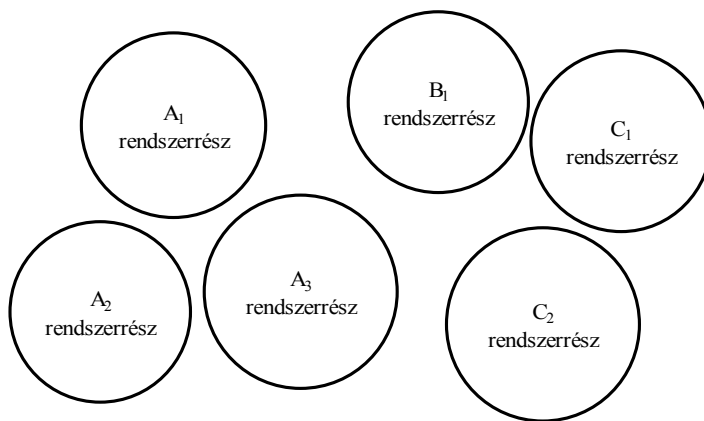
A határok egyértelmű és pontos kijelölése kényes és nehéz feladat (az valóságban sehol egy támasz, zsinog vagy demarkációs vonal), de a jól bejelölt határok kedvező alapot biztosítanak a későbbi elemzésnek. A környezeti elemek közül csak azok maradnak meg, természetesen, amelyek felé a rendszerelemektől mutat számunkra jelentős kapcsolat.

Fentebb már volt szó a rendszerek relatív mivoltáról: tudniillik, hogy bárki önkényesen meghatározhat magának akárhány rendszert. Ez esetben felvetődik a kérdés, hogy vajon létezik-e

valamilyen tárgyilagossá mércé a rendszer határainak kialakítására? Általános útmutató adható csak válaszként erre a kérdésre: a rendszer határait csak a rendszer lényege és céljai határozzák meg.

3.3. A RENDSZER FELOSZTÁSA – TAGOLÁSA

Mint ahogy ezt már kifejtettük, a rendszer részét is rendszernek kell tekintenünk. A rendszerek, általában igen bonyolultak, mondhatnánk úgy is, hogy túlságosan bonyolultak ahhoz, hogy egészben egyszerre át tudnánk tekinteni, fel tudnánk fogni őket. Ezért kell felosztáshoz folyamodni. Nem egy ördögös új ötletéről van szó, hiszen ezt az elvet már a régi rómaiak is ismerték, amiről a „Divide et impera!” szólás tanúskodik. A hangsúly a felosztás mikéntjében van: a rendszerrészeknek *összetartozóaknak* kell lenniük. Ezt példázza a 3.3. ábra, amelyen a rendszerrészeket körök ábrázolják, és mint látható az összetartozást a körök átfedésével érzékeltettük. Az ellenpélda a 3.4. ábrán helyezkedik el, ahol három külön rendszer független (rosszul taglalt) rendszerrészei láthatók. A rendszer felosztásáról már csak a meghatározása (definiálása) után érdemes gondolkodni. A rendszer meghatározása annak megismerésével és működésének felfogásával, megértésével kezdődik, majd a tényezők megjelölésével és megnevezésével folytatódik, végül a belső és külső környezet kialakításával, vagyis a határok meghúzásával fejeződik be.



3.4. ábra

Ma még nem adható pontos formális lépéssor (metodológia) a tagolás elvégzésére, amely egy tetszőleges rendszer konkrét, pontos és jó tagolását biztosítaná. Ismert és elfogadott,

azonban, az a hozzáállás, hogy a rendszer felosztása csak annak definiálása után következhet be, vagyis a következő (fentebb is említett) lépések elvégzése (megvalósítása) után:

- a célok (célhierarhia) pontos megfogalmazása (miért, mi célból működik vagy tervezik a rendszert),
- a tartalom meghatározása, vagyis a tényezők iktatása, definiálása,
- a határok bejelölése (a tényezők fontosságának mérlegelése) és ezzel egyidejűleg
- a belső és külső környezet létrehozása.

3.4. KETTŐS RENDSZERNÉZET

A kettős rendszernézet fogalma már régóta ismeretes a fejlesztők számára az irodalomból. Sokan vallják ma is, hogy létezik

- elsődleges és
- másodlagos rendszer.

Ez a felosztás az eredendőségen, és nem a fontosságon alapul. A legtöbb szervezetben (a termelővállalatok túlsúlya miatt) a termelés képezi az elsődleges, a róla vezetett ismeretek pedig a másodlagos rendszert. Az elsődleges rendszer léte mindig is (függetlenül attól, hogy mi képezi az elsődleges, és mi a másodlagos rendszert) a másodlagos rendszer létének a feltétele. Más a helyzet egy statisztikai hivatalnál, biztosító intézetnél vagy banknál: ezeknél a kimutatások (egyres cégek termelési adatai, különböző káresetek adatai, tőzsde- és pénzpiaci adatok) képezik a primáris (elsődleges), maga a “termelés” (különböző jelentések) pedig a másodlagos rendszert.

A valós világról bennünk kialakult kép (elképzelés) ismereteink alapján alakul, formálódik. Mindenképpen a valóság az elsődleges, ismereteink csak másodlagosnak számítanak. Ha nincs a valós világ, ismeretek sem fogalmazódhatnak meg vele kapcsolatban.

A valós rendszer (a termelő vállalatok termelési folyamata vagy az adatok a biztosító intézetekben, bankokban, hivatalokban) az, amelyről ismereteket, adatokat akarunk vezetni, tehát ezt kell elsődleges rendszernek tekintenünk, maga az adatgyűjtés és –kezelés a valós vagy elsődleges rendszerről pedig a másodlagos rendszert öleli fel. Újabb időkben a valós rendszer kifejezés helyett gyakran kérkednek a “University of Discourse” (UoD) kifejezéssel, ami semmi mást nem jelent, mit azt “amiről szó van”. Ettől ugyan a valós rendszer még valós rendszer marad, és semmi sem lesz sem jobb, sem rosszabb egy információs rendszerben.

4. INFORMÁCIÓS RENDSZER – RÉGI FOGALOM ÚJ TARTALOMMAL

4.1.FOGALMAK ÉS FELFOGÁSOK

A számítógépek alkalmazása az élet minden területén észlelhető. Egyes területeken a számítógépek elengedhetetlen “kellékké” váltak, ami természetes is. Másrésztől azonban divatossá vált már egy ideje számítógéptulajdonosnak lenni (mint ahogyan manapság WAP-os telefont birtokolni, pedig még támogatás – tehát WAP-os oldalak – nincs hozzá): akinek nincs számítógépe az maradi. Ez eddig rendjén is volna, a baj csak az, hogy a gépvásárlástól kezdve minden tulajdonos informatikusnak képzei magát. Érteni a gép kezeléséhez és informatikusnak lenni két nagyon különböző dolog. A szövegszerkesztő, levelező- vagy rajzprogram használatának ismerete még senkit sem informatikussá. Az információs rendszerek esetében keveredik a cél, az eszköz és a módszer fogalma: ha egy alkalmazás (információs rendszer) lassan vagy rosszul működik, gépparkot újítanak, szoftvert frissítenek ahelyett, hogy az alkalmazásba “néznének bele”, vagyis ellenőriznék a feladat megoldását (a hiba ugyanis legtöbbször ott lapul).

Fontos rámutatni, hogy a számítógépet és a rajta futó szoftvereket ügyesen kezelő személy nem tekinthető magától értettődően egyúttal és egyidőben informatikusnak is. Ennek csupán egyetlen, de nyomós oka van: a gondok az informatikában általában (pontosabban főleg) nem technikai jellegűek (nem hardverrel vagy szoftverrel – adatbáziskezelőkkel illetve alkalmazásfejlesztőkkel kapcsolatosak). A számítógép használata nem mindig kapcsolatos az informatikával. Az informatikában a problémát az információs rendszerek bonyolultsága és nagy száma jelenti.

Az informatika fogalmát többféleképpen is megfogalmazták. A sokféle definícióból kettőt ismerünk csak meg: egy lakonikust (rövidet) és egy hosszabbat, átfogóbbat. Az utóbbi szerint:

“Az informatika az információ áramlásának különböző módozataival, feldolgozásának és hasznosításának módszereivel, a termelékenységre és hatékonyságra gyakorolt hatásaival, megfigyelési és ellenőrzési célokra való felhasználásával és végezetül a társadalmi-gazdasági fejlődést és a társadalmat alakító szerepével foglalkozik”¹. Az informatika fogalma a fenti definíció szerint nagyon általános, hiszen felöleli az adatok rendszerbe fogását, a feldolgozási eszközöket, az adatfeldolgozás módozatait az ismeret születésétől a végfelhasználóhoz való eljuttatásáig, illetve a jelentésekbe foglalt adatok,

¹ A budapesti Központi Statisztikai hivatal meghatározása, forrás: [20.], 8. Oldal.

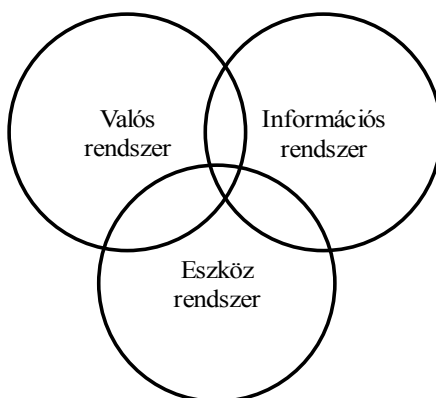
információk hatásvizsgálatát a gazdasági szubjektumokra. Az előbbieket szerint az informatika egy olyan átfedő (interdiszciplináris) tudomány, amely felöleli a hírközlés (kommunikáció) valamennyi formáját, a számítástechnikát és annak alkalmazását, a mikroelektronika és robotika eredményeinek gyakorlati alkalmazás-lehetőségeit, az irányítás és vezérlés gyakorlatának bizonyos területeit, illetve mindezek érvényesítését (és érvényesítésének lehetőségeit) a gazdaságban, a természet- és társadalomtudományokban. Az informatika ebben az értelmezésben számunkra túlságosan sokrétű fogalmat takar.

A rövidebb meghatározás szerint, **“Az informatika az ismeretek megismerésének, azok célszerű elrendezésének és kezelésének a tudománya. Az informatikus az a szakember, aki ebben a tudományban jártas”**².

A valós világ megismerése, ábrázolása és az azt támogató információs rendszer megvalósítása közben az informatikus mindig a következő három rendszert és azok összefüggéseit kell, hogy kövesse:

1. az objektív valóságot (valós rendszer – VR),
2. a róla kialakított ismereteket (információs rendszer – IR) és
3. az ismeretek kezelését biztosító rendszert (amely a hardvert, a szoftvert, az élőerőt – lifewart és a szervezést – az orgvert, egyszóval az eszközzrendszert – ER jelöli).

Itt három, részben átfedő rendszerről beszélhetünk (4.1. ábra³), amelyek az átfedések mellett aránylag nagy függetlenséggel is rendelkeznek. A három rendszer harmonikus egységének a megteremtése a végleges cél, amit manapság egyre inkább figyelmen kívül hagynak. A mai átlagember az eszközök bővületében él: a gyors számítógépek és a legújabb szoftverek jelentik számára a végcélt.



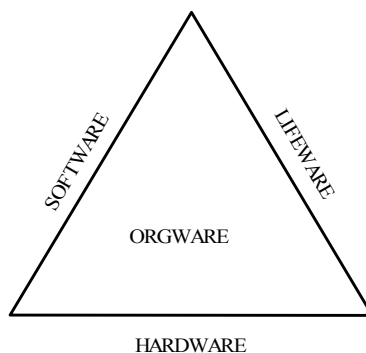
4.1. ábra

² Forrás: [11.], 35. oldal

³ Forrás: [11.], 36. oldal

Az eszközrendszer grafikusán is ábrázolható⁴, ahogy ezt a 4.2. ábra is mutatja. Az eszközrendszer négy részrendszerének szoros összetartozására utal (vagy legalábbis szeretne utalni) a 4.2. ábra eredeti angol kifejezésekkel és az összetartozást szimbolizáló zárt vonal alkalmazásával.

A **hardvert** (hardware) a rendszer megfogható, anyagi eszközei alkotják: a számítógép(ek) az összes tartozékaikkal, bemeneti, bemeneti-kimeneti és kimeneti egységeikkel, a vezetékes és vezeték nélküli hálózat megvalósításának berendezéseivel, az elsődleges adatgyűjtést, -rögzítést, előfeldolgozást és –elosztást végző egységekkel. A hardver rohamos léptekkel fejlődik, alkalmazásaink nem tudnak lépést tartani az ilyen mérvű fejlődéssel, és a (hardver) lehetőségeknek csupán néhány százalékát használják, bár a szoftverek újabbnál újabb változatai egyre igényesebben a hardverrel szemben (sokak szerint a hardver csupán a szoftverkövetelmények miatt fejlődik, mások szerint a helyzet pontosan fordított). Általános becslések szerint a hardver fejlettségi foka tíz-tizenöt évvel megelőzi a szoftverét.



4.2. ábra

A **szoftver** a (rendszer szintű) program-megoldásokat jelöli: az operációs rendszereket, a különböző kiszolgáló-karbantartó programokat, a programnyelveket, adatbáziskezelő rendszereket, szövegszerkesztőket, táblázatkezelőket, rajzprogramokat, kép- hang- és filmfeldolgozó valamint más általános célú (nagyobb embercsoport számára érdekes) problémát érintő alkalmazásokat. A szoftverfejlődés talán látványosabb volt az utóbbi években a hardverénél, de egyesek még a fent említett lemaradást is kevésnek taksálják, és húsz évet emlegetnek. A mai adatbáziskezelő rendszerek, a szoftverfejlődést igazolandó, olyan bonyolultak, hogy a laikusok talán csak elindítani tudják őket, de érdemlegesen használni nem, a problémakörhöz jól értő szakemberek (rendszerelemzők, rendszertervezők) pedig lehetőségeiknek csak néhány százalékát tudják alkalmazni.

⁴ Forrás [6.]

Az információs rendszerek megvalósítását végző szakemberek alkotják az élő erőt, a **lifewaret**. A mai informatikai káderek a számítástechnikában talán jobban képzettek mint kellene (ami természetesen nem baj), viszont az információs rendszerek természetét, tényezőit, tényezőinek számát, jellegét és összefüggéseit nem ismerik eléggé (ami azonban hiba).

Az eszközrendszer negyedik részrendszere, az **orgver** az előző három eszköz-rendszerrész struktúrába szervezését valósítja meg. Az eszközrendszer struktúrájának kialakítása nem önös érdekből történik, hanem azzal a céllal, hogy az információs rendszer a megrendelő valamennyi elvárását maradéktalanul teljesíteni tudja. Az orgver az információs rendszer fejlesztéséért és működtetéséért felelős a szervezés oldaláról.

Az információs rendszer kifogástalan működésének szempontjából az eszközrendszer négy rendszerrészének működési és egyéb szempontokból is harmonizálniuk kell. Bármely rendszerrész elhanyagolása vagy nem megfelelő működése (együttműködése) a maradék három rész bármelyikével az információs rendszer működésében előbb-utóbb komoly zavart idéz elő.

4.2.AZ INFORMÁCIÓS RENDSZER FOGALMA – MODERN SZEMLÉLETBEN

Az információs rendszer, ahogy a neve is elárulja, maga is rendszer, tehát rá is érvényesek a rendszerelméletben megállapított és bizonyított törvényszerűségek. Itt a megfelelő pillanat, hogy rámutassunk az információs rendszer és az információk rendszere (információ-rendszer) fogalmak közötti különbségre. Az információk rendszere csak az adatok (információk) elrendezésére, szervezésére szorítkozik, az információs rendszer ennél sokkal többet takar. Az információs rendszerek és fejlesztésük megismerésének első lépéseként az információs rendszer fogalmával ismerkedünk meg.

»Az információs rendszer

1. adatoknak (információknak);
2. a velük kapcsolatos információs eseményeknek;
3. a rajtuk végrehajtott információs tevékenységeknek;
4. az előzőekkel kapcsolatos erőforrásoknak;
5. az információk felhasználóinak;
6. ill. A fentieket szabályozó szabványoknak és eljárásoknak

szervezett együttese.«⁵

A fenti definíció egyes tényezőinek alaposabb jellemzése persze elkerülhetetlen, de lássuk előbb magát a meghatározást, úgy egészében véve. Az első megállapítás a fenti definíció kapcsán az, hogy az információs rendszerek az inhomogén rendszerek családjába tartoznak. Ezt igazolja a tényezők és kapcsolataik fajtáinak, jellegeinek különbözősége. A felhasználók, adatok, standardok, események, tevékenységek, hardver és szoftver különbözőségét, gondolom nem szükséges feltétlenül bizonygatni.

Az információs rendszer kísérőjelzői a mindennapi beszédben éppúgy, mint a szakirodalomban gyakran tévesek, túlzók vagy ismételtetések (tautológiát) rejtenek. Gyakran használják a komplex és integrált jelzőket, sőt kapcsoltan is mindkét jelzőt, valahogy így: »komplex, integrált információs rendszer«. Az inhomogén rendszerek (amelyek közé az információs rendszerek is tartoznak) már eleve (»ab ovo« vagy a priori) komplexek, tehát nem kell ezt még a jelzővel is hangsúlyozni (tautológia). Az integrált jelző ugyancsak tautológiát rejt, hiszen minden rendszer a tényezők szervezett – integrált együttese (már a fenti definíció szerint is). A »vezetői« jelző használata járatlanságra, butaságra vall, bár divatos volt és ma is az még a vezetői információs rendszer (MIS – Management Information System) kifejezés. Vezetői információs rendszer, mint olyan ami a nevéből kikövetkeztethető lenne, magában nem is létezik, vagy ha igen, hát nem sok értelme van. A vezetők (menedzserek) számára szükséges információk mindenképpen az alapinformációkból származnak, és egyfajta kivonatként (információ-kivonatként) foghatjuk fel őket. A döntésekhez elengedhetetlen információk (adatok) ugyanis mindenképpen megtalálhatók egy „mindent átfogó, komplex, integrált információs rendszer” rendszerrészeiben. Nevezzük azonban ezt a „mindent átfogó, komplex, integrált információs rendszer”-t egyszerűen csak információs rendszernek.

4.2.1. Adat, információ, tudás

Az adat, információ és tudás fogalmát sok elméleti és gyakorlati tudós-szakember formálta már a mai napig. Az első két fogalmat különös gonddal kezelték és kezelik, rámutatva a köztük megfogalmazható lényegbeli különbségekre.

Az adat legegyszerűbb megfogalmazásában a hangsúly a lejegyzésen van, vagyis ez a szemlélet azt vallja, hogy az adat tetszőleges adathordozón (a barlang falán, a csárda gerendáján,

⁵ Forrás: [11.], 43. oldal.

papíron, mágneses médiumon, stb.) lejegyzett valamiféle ismeret a valós világ bizonyos jelenségéről, a lejegyzett adat későbbi sorsától teljesen függetlenül. Mellékes tehát, hogy ezt a lejegyzett ismeretet használják-e, és ha igen, mire a későbbiekben. Ha ezeket a rögzített adatokat a továbbiakban a döntések meghozatalában, vállalatvezetésben és –irányításban használják, úgy az adatok információkká alakulnak át. A lejegyzett ismeretek adat jellegüket egészen addig megőrzik, amíg esély van arra, hogy információvá alakuljanak. Ezután az ismeretek adat jellegüket is elvesztik⁶.

Ez az adatokkal és információkkal kapcsolatos nézet ma már nem állja meg a helyét. Ma már azt valljuk, hogy az adat az információ »megtestesülése«, konkrét megjelenési formája, olyan jelhalmaz amelyet számítógépen is feldolgozhatunk, de »szegényebb«, kevésbé jelentős, mint az információ. Az információ tartalmilag gazdagabb az adatnál. Az adat értelmezhető, de nem értelmezett ismeret. Az információ ezek szerint értelmezett adat, értelmezett ismeret. Az adatok tehát információhordozók, de az adatok »információgazdagsága« igencsak viszonylagos fogalom, hiszen nagymértékben függ az adat vagy információ befogadójától. Ugyanaz az adat egyesek számára továbbra is csak adat marad (a befogadó nem tudta értelmezni), másoknak esetleg értékes információt hordoz (ők értelmezni is tudták a kapott adatot).

Az adat tehát olyan értelmezhető, de nem értelmezett jelsor, amely objektumok állapotát és/vagy tulajdonságait, vagy a valós világ egyéb jellemzőit fogalmazza meg (rögzítésre, szállításra és feldolgozásra megfelelő alakban) a későbbi felhasználás és/vagy feldolgozás céljából.

Az információelmélettel foglalkozó kutatók a mai napig nem tudtak az információ fogalmának pontosabb meghatározásában megegyezni. Ezt igazolandó, lássuk az információ fogalmának különféle definícióit, melyeket nagyon nehéz közös nevezőre hozni.

B. Langefors véleménye a következő:

“Információnak tekintünk minden tudást, üzenetet, amelyet bizonyításokhoz használhatunk, vagy amely lehetővé tesz döntéseket.”⁷

A Webster's Dictionary-ban az információ két definíciójával is találkozhatunk:

Az információ tudásanyag, intelligencia, hír.

Az információ a tudás vagy intelligencia átadása illetve fogadása.

H. Rittel norvég kutató az információt a tudásszint változásához köti:

⁶ Forrás: [6.]

“Az információ egy olyan tevékenység, amely valaki tudásának a megváltoztatásához vezet.”⁸.

A következő meghatározásban az információ a rendezett ismereteket jelöli:

“Az információ nem egyéb, mint rendezetlen adatsorok vagy tények valamely folyamat által használható formára alakított változata.”⁹.

Hasonló elveket vall NoszkayErzsébet is az információ fogalmáról:

“... az információ nem általában valamiféle új ismeret, hanem olyan új, illetve feltárt ismeret, amely a már meglévő ismeretek (adatok, tények) rendszerezéséből, összevetéséből, elemzéséből, értékeléséből, modellszerű felhasználásából stb. származik.”¹⁰.

Az információ értelmezett adat, állítás, ismeret, közlés, hír, amely csökkentheti a bizonytalanságot, döntés és irányítás (vezérlés) alapjául szolgálhat, és mindig gazdagabb, több, mint az adat és az összes fenti fogalom, mert gondolkodásra, döntésre, tevékenységre készíteti a címzettet (az információ befogadóját).

Az adatok értelmezhetősége és értelmezése nemesíti őket információvá. Az értelmezésig azonban el is kell jutni, méghozzá a következő lépéseken keresztül:

1. érzékelés,
2. észlelés és
3. felfogás illetve megértés.

Az informálódás vagy ismeretszerzés mindhárom fenti mozzanata kritikus jelentőségű. Bármelyikben történik is valami rendellenesség, vagy maradjon ki akár egy lépés is, nem jöhet létre információ (ismeretszerzés).

Az érzékelés, mint a neve is mondja, érzékszerveink működését jelenti az adatok fizikai “felfogására”. Az érzékszervek (főleg a látás és hallás szervének) csökkent szintű működése hátrányos helyzetbe hozhatják “gazdájukat”, az ismeretek címzettjét.

Az észlelés mozzanata (azért kedvelem ezt a kifejezést, mert jól érzékelteti az ismeretszerzés folyamatának gyors bonyolódását) érthető meg talán a legnehezebben az ismeretszerzés folyamatában. Az észlelés azonban jól megfogható a következő hétköznapi (valószínűleg már mindegyikünkkel megtörtént) eset kapcsán. Reggeli közben az emberek nagy

⁷ Forrás: [22.], 13. oldal.

⁸ Forrás: [22.], 13. oldal.

⁹ J Frates szavai, [22.], 13. oldal.

része a napi sajtót lapozza. Ha ilyen helyzetben valaki szól hozzájuk, gyakran a következő kérdéssel felelnek vissza: Mit is mondtál? Az érzékelés tehát megtörtént, de az észlelés nem, hiszen a megszólított ember a figyelmét másfelé irányította! Ilyen esetekben, mihelyt észbekap az ember, hogy valaki hozzá szólhatott, megindul az érzékelés “visszajátszása”, és ha ez sikertelen volt, következik a visszakérdezés. A második példa az észlelésre legyen a szolgáltató vállalatok számlája. A számla-jelentés annyira túlszűfolt adatokkal, hogy általában érthetetlen és a fogyasztó számára felfoghatatlan, hogy hányadán is áll a számla kiállítójával. Egy időben például (pedig ekkor az informatikai részleg főnöke voltam a szabadkai áramelosztóban) nem tudtam, hogy pontosan mennyi is lesz a villanyszámlámon a tartozás (fogyasztás) összege, bár a fogyasztást, ugye le tudtam olvasni a fogyasztásmérőről. A számításához szükséges elemek is a számlán helyezkedtek el valahol, de annyira eldugva, hogy ember legyen a talpán, aki megtalálja valamennyit! Nota bene¹¹: az észleléshez az ismeretek fizikai szervezése is hozzájárul, ezért a fontos ismereteket mindig feltűnő, szembetűnő helyen kell megjelentetni, hogy azok azonnymódon észlelhetők legyenek.

A felfogás illetve megértés az utolsó lépést, mozzanatot jelölik az értelmezés előtt. Az ismereteket legtöbbször kijelentés (állítás) vagyis hír formájában fogalmazzuk meg. A kijelentést csak az a címzett tudja felfogni, aki érti annak minden szavát, másszóval, ha a hír megfogalmazója és megcélzottja (címzettje) »ugyanazt a nyelvet beszélik«.

Az ismeret értelmezése nem más, mint az új állítás (hír, adat) összevetése, összekapcsolása a már előbb megszerzett információkkal. Ha nem léteznek előző információk, amelyekkel az új kijelentés (ismeret) összeköthető, az ismeret értelmezése nem lehetséges, vagyis új információ nem keletkezik. Új ismeret (információ) akkor sem keletkezik, ha a kapott adat kötése az előzőekben már megtörtént. Információ akkor születik amikor a kapott adat (hír, állítás) tartalmáról gondolkodunk, amikor valamilyen véleményt fogalmazzunk meg vele kapcsolatosan, amikor elhelyezzük a már meglévő ismereteink közé (között).

Az előző megállapításokra alapozva bátran kijelenthető, hogy a számítógépeken csupán adatokat (és nem információkat) tárolunk azzal a céllal, hogy később ezeket visszakeressük, átalakítsuk, feldolgozzuk. Információk csak az emberek fejében születnek, és az információs rendszerek, vagy feldolgozások csak akkor jók, ha megkönnyítik és meggyorsítják az értelmezés folyamatát.

Ismereteink a valós világról a jelenségeinek és a folyamatainak megismeréséből és tanulás útján keletkeznek, és következtetés (vélemény) megfogalmazását, ok-okozati összefüggés-

¹⁰ Izvor: [20.], strana: 9.

¹¹ Jegyezd meg jól

vizsgálatot, elvonatkoztatást és új ismeretek megszerzését teszik lehetővé egy magasabb elvonatkoztatási szinten, amit tudásnak nevezünk.

»A tudás a valós világnak, az abban létező dolgoknak, tényeknek, eseményeknek, jelenségeknek, az azok között fennálló kapcsolatoknak, okozati összefüggéseknek az emberi tudatban történő visszatükröződése.«¹²

4.2.2. Információs tevékenység és információs esemény

Az információs tevékenységek és események szorosan kapcsolódnak egymáshoz, ezért célszerű együtt elemezni őket. Az adatokkal (információkkal) kapcsolatos tevékenységek egy társadalomban nagyon sokrétűek lehetnek, de bennünket csak az információs rendszerek szempontjából érdekelnek a tevékenységek, így a meghatározás a következő lehet:

Az információs tevékenység olyan szervezett műveletsor, amely adatkezelési, -előállítási és vezérlő műveletekből áll. Ez utóbbiak az előző két műveletfajta összehangolására, sorrendiségének kialakítására alkalmasak.

Az adatkezelési műveletek (bevitel, javítás, törlés, másolás, mentés, rendezés, megjelenítés stb.) nem hoznak létre új ismereteket (információt). Az adatbevitel és adatmódosítás (javítás) egyesek szemében talán új információ születését jelzi, de fogjuk ezt föl úgy, hogy csak a meglévőt tudjuk rögzíteni és módosítani.

Az adatelőállítási műveletek új információkat (ismereteket) hoznak létre. Legegyszerűbb példája az adatelőállítási műveleteknek a számla tételértékének kiszámítása, amely a mennyiség és az egységár összesorzásából áll. Vegyük észre, hogy amíg az adatkezelési műveletek eszközfüggőek, addig az adatelőállítási műveletek a felhasználó igényeit tükrözik.

A vezérlő műveletek biztosítják azt az összekötő kapcsolatot, amely egy információs tevékenység keretein belül megvalósítja a kezelési és előállítási műveletek megfelelő váltakozását és ütemezését biztosítva ezzel a tevékenység szervezettségét.

Az információs események határolják be az információs tevékenységeket: elindítják, illetve lezárják azokat.

Az információs eseményeket természetesen a felhasználói igények hívják életre. A felhasználói igények megismerésével fogalmazhatók meg tehát az információs események. A felhasználói igény indítja az információs eseményt, amely kiváltja a megfelelő információs tevékenységek végrehajtását. A tevékenységek végrehajtása lehet sikeres vagy sikertelen, így

több (legalább kettő) lezáró eseménye van egy-egy tevékenységnek. Másrészt ugyanazt az információs tevékenységet több különböző esemény is kiválthatja, így könnyen belátható, hogy az információs események és tevékenységek láncolatával van dolgunk.

Az információs események a rendszerben zajló fontos változásokhoz kötődnek. A változások fontosságát a felhasználó szemszögéből (sőt, vele karöltve) kell megítélni. Ilyen fontos változások lehetnek a következők:

1. a jelenségek (egyed típusok) életciklusa, amely a születést, változást és megszűnést öleli fel (új vevő megjelenése, meglévő vevő adatváltozásai, vevői szerepkör megszűnése)
2. a változások szempontjából az idő is fontos tényező¹³, amely információs eseményeket indíthat el: pl. a forgalmi adót D (tizenöt) naponként kell befizetni.
3. az állapot megváltozása bizonyos esetekben információs eseményeket válthat ki: ha a raktárkészlet állapota egy adott minimális szint alá süllyed, az elindíthatja az eseményt a rendelés létrehozására.

A hasonló információs aktivitások képezik a funkciókat, a rokon funkciók pedig az alrendszereket alkotják, amelyekből a rendszer egésze felépül.

Még egy tényre fel kell hívni a figyelmet, bármennyire is hihetetlenül hangzik manapság: egy jól megtervezett információs rendszer élet- és működőképes a számítógép segítségével is. A cél természetesen az, hogy minél nagyobb legyen az automatizáció foka a tevékenység-háló megvalósításában. Ettől függ ugyanis, hogy a számítógépes feldolgozás milyen mértékben segíti az információs rendszerben az adatkezelést.

4.2.3. Információs erőforrások

Általánosan elfogadott az a vélemény, hogy a rendszer és az erőforrások között lényegileg ugyanaz a viszony áll fenn, mint a cél és az eszközök között. Machiavelli óta tudjuk, hogy a cél szentesíti az eszközt, másszóval, hogy a cél az elsődleges, az eszköz pedig a másodlagos fogalom. Gyakran mégis az eszköznek tulajdonítanak fontosabb szerepet, ez pedig azért rossz, mert az információs rendszer fejlesztésének kérdése háttérbe szorul. Tágabb értelemben az adat, az információs esemény és tevékenység, de még a szabványok is, nemcsak tényezői, de erőforrásai is az információs rendszernek. Szűkebb értelemben az **erőforrás** fogalma alatt az információs rendszer fejlesztéséhez, működtetéséhez, karbantartásához szükséges **időt, pénzt** és

¹² Forrás: [22], 15. oldal.

¹³ Tempora mutantur et nos mutamus in illis. – Változik a világ (idő), s vele együtt (benne) mi is.

technikai eszközöket (hardver, szoftver) értjük. Fontos hangsúlyozni, hogy az erőforrások nem konvertálhatók tetszőlegesen egymásba.

Vegyük elsőként az erőforrások közül a pénzt. Bármennyire is növekszik manapság a pénz imádata, ne becsüljük túl ezt az erőforrást. Igaz ugyan, hogy pénz nélkül az információs rendszer egyetlen tényezője sem biztosítható, még kevésbé a tervezés, megvalósítás és működtetés. A pénzmennyiség sokszorozásával azonban nem csökkenhet arányosan a fejlesztéshez szükséges idő, mert a valós világ elemzése és leírása (modellezése) időigényes, és a szakmában jártas, lelkes elemzők-tervezők nem rendelkeznek “TURBO” gombbal a világ összefüggéseinek gyorsabb felismerésére.

A gyöngye információs rendszer-terveket sokszor a nem megfelelő vagy az eszközök és erőforrások hiányával magyarázzák. Ennek megfelelően nyugodtan hivatkozhatnánk arra, hogy azért olyan silány és felületes ez a jegyzet, mert a számítógépem sűrűn és sokszor, de főleg váratlanul lefagyogatott (valószínűleg a memória-chipek összeférhetlensége miatt). Vagy ugyanez, csak megfordítva: százszor jobbminőségű munkát is tudtam volna teremteni, ha gyorsabb és stabilabb gépem lett volna. Persze mindez nagy csacsiság, mert a jegyzet minősége tőlem és egyedül csak tőlem függ (nem pedig a rossz »golyóstolltól«).

Az információs erőforrások ugyancsak tényezői az információs rendszereknek, és mivel az információs rendszer a tényezők szervezett együttese, az erőforrások összhangba hozása az információs rendszer többi elemével (tényezőjével) elengedhetetlen.

4.2.4. Az információs rendszerek felhasználói

Az információs rendszer definíciójában szereplő valamenyi tényező egy-egy összetett rendszert képez a valóságban. Az adatok az információs tevékenységek végrehajtása folyamán kerülnek feldolgozásra, a tevékenységeket pedig események indítják el (és zárják le), ami az elemek (tényezők) szervezettségére, szoros összefüggésére utal. Az információs rendszer fogalmának meghatározásában a tényezők többes számban szerepelnek (adatok, tevékenységek, események). Az adatok tehát a tevékenységekhez és eseményekhez kapcsolódnak, de ugyanakkor az adatok között is léteznek kapcsolatok, a különböző tevékenységek között is, és az események is összefüggők.

Az információs rendszer felhasználói is egy sajátságos rendszert alkotnak, amelybe beletartozik minden olyan személy és szervezeti egység ha bármilyen kapcsolatba kerül az információs rendszerrel:

A felhasználók olyan személyek vagy embercsoportok (szervezeti egységek) amelyek valamilyen kapcsolatba kerülnek az ismeretekkel, illetve az információs rendszerrel.

Mivel a személyek többfajta szerepben is jelentkezhetnek az információs rendszer életében az ismeretek, információk használatát illetően, megállapítható, hogy a felhasználó nem abszolút fogalmat elől.

A **végző-felhasználó(k)** az(ok) a személy(ek) aki(k) számára (akik miatt) az információs rendszer készül. Egy biztosító intézet számára tervezett információs rendszer végfelhasználói maguk a biztosítottak. Vegyük észre, azonban, hogy az évvégi zárszámadás a részvényeseknek készül, így a zárszámadás szempontjából ők tekinthetők végző-felhasználóknak (a felhasználó viszonylagos fogalom).

Az **alkalmazási-felhasználó(k)** az(ok) a személy(ek) akik közvetlen kapcsolatban vannak az információs rendszer ismereteivel, és gyakran a végző felhasználókkal is. A biztosító intézet szerződéskötő ügynöke a biztosítási szerződés (ajánlat) alkalmazási felhasználójának tekinthető (függetlenül attól, hogy leül-e valaha is a számítógéphez). A fizetési lista szempontjából, természetesen ő is végző-felhasználó.

A **fejlesztő(k)** ugyancsak felhasználó(k). Ugyanazokkal az adatokkal foglalkoznak, mint az előző két csoport tagjai, csak másféle aspektusból. Hol az egyik, hol a másik »szerepbe bújnak« (végző felhasználó, alkalmazási-felhasználó) a fejlesztés folyamán a tökéletesebb végtermék érdekében.

A **vezető** specifikus (sajátságos) felhasználó: ő rendelkezik az erőforrásokkal, megbecsült, hatalommal bíró személy. Ő az a személy, aki ébren tudja tartani az érdeklődést a fejlesztés iránt, és rá tud mutatni a szervezet fejlődési irányára.

A **szervezeti egységek** szintén felhasználóknak tekinthetők a fogalom tágabb értelmében. Ha a biztosító intézet egyik szerződéskötő ügynöke szabadságra megy vagy megbetegszik, szerepét, feladatait szervezeti egységének egy másik ügynöke veszi át, hiszen a szervezeti egységek felelősek a rájuk bízott feladatkörök ellátásáért.

4.2.5. Szabványok és eljárások

Egyesek áhítják a szabványokat, mások irtóznak tőlük. Akik bilincsként, szabadságot megkötő, korlátozó tényezőként tekintenek a szabványokra, bizonyosan nincsenek elragadtatva a létezésüktől, még kevésbé a terjedésüktől. Akik viszont a hasznos útmutatót, az általánosan elfogadott, kipróbált elvek, törvényesen előírt elvárások kelléktárát látják bennük, szívesen alkalmazzák a mindennapi munkájukban. A szabvány persze lehet akár egyik, akár másik, akár

mindkét (jó és rossz) dolog egyszerre, mert a szabvány fogalma alatt az információs rendszer tényezőivel kapcsolatos konvenciókat, megegyezéseket kell érteni. Ezek a megegyezések egymással is összefüggnek, vagyis struktúrát alkotnak, ami azt jelenti, hogy szabványok sajátos rendszert (szabványrendszert) képeznek az információs rendszeren belül.

A szabvány fogalom meghatározása szólhat a következő módon is:

A szabvány megegyezés, konvenció vagy előírás az információs rendszer valamely tényezőjével kapcsolatban.

Vannak általánosan elfogadott vagy előírt szabályok úgy adatokra mint eljárásokra nézve. Vegyük csak a már többször említett biztosító intézetet és gondolkodjunk el annak (de akármely más vállalatnak a) könyvviteléről. Törvény által szabályozott, hogy milyen adatokat kell vezetni a könyvviteli kimutatásokban, milyen jelentéseket kell kidolgozni, pontosan milyen eljárások alapján, és milyen alakban kell ábrázolni a kért jelentéseket a pontosan meghatározott határidőkre. Éppen ezért a nyugati világban gyakran a szabványok és eljárások kettősségében gondolkodnak és beszélnek.

Első közelítésben talán úgy tűnik, hogy az eszközökre (technikai eszközökre: hardver, szoftver) legkönnyebb szabványt alkotni. Elvben a hardver eszközök szabványosak, a gyakorlatban azonban más a helyzet. Valószínűleg ritkábban jelenne meg a képernyőn a "General Failure ..." kezdetű üzenet, ha az öt-hat éve vásárolt számítógépembe már eredetileg is, és a folyamatos javítgatások alkalmával is szabványos alkatrészek kerültek. Mégis azt állították róla a vásárláskor, hogy PC kompatibilis... A szoftverekkel sem jobb a helyzet: a Corel 2.0-ban megrajzolt rajzaimat sehogyan sem tudtam "megetetni" a Corel 7.0-val. A szoftverek újabb verziói ritkán hasonlítanak a régebbi megvalósításokra, még arra sem, amelyiket éppen felváltják. Mivel a hardver-gyártók és szoftverházak nem, vagy csak nagyon nehezen tudnak megegyezni egy közös szabványban, a felhasználóknak saját, belső, hardver-szoftver szabványokat kellene kialakítaniuk. Az informatikai erőforrások másik két eleme, az idő és a pénz még nehezebben szabványosítható. Egymással is kapcsolatosak, de külön-külön is nehezen becsülhető meg az egy-egy informatikai feladat megoldásához szükséges idő és a feladat megoldásának pénzbeli értéke. Az információs rendszer fejlesztése nem mérhető csupán a forráskód sorainak számával, ahogy ezt Amerikában a hetvenes években tették, mert a fejlesztés egyrészt nemcsak programozásból áll, másrészt a fejlesztés minőségét a kapott adatok (információk) használhatósága, aktualitása, fontossága is meghatározza. Ez utóbbiak természetesen nemcsak az információs rendszer minőségétől függnék, de az információs rendszer által feldolgozandó (bemeneti) adatok minőségétől is (Garbage in, garbage out).

A felhasználói szabványok létezése ugyancsak elengedhetetlen. A felhasználói szabványok a használt fogalomkészletet és a fogalmak elnevezéseit tartalmazza. A végső- és alkalmazási felhasználó számára a felhasználói szabványok egyértelművé teszik az információs rendszer funkcióit, aktivitásait, jelentéseit, stb.

Az előzőek alapján megállapítható, hogy a szabványoknak és eljárásoknak az információs rendszerek fejlesztése és üzemeltetése foyamán a következő hármas szerep jut:

1. eligazítják a felhasználókat (különösen a fejlesztési fázisban a rendszertervezőket, rendszerelemzőket és programozókat), hogy mit hogyan kell csinálni,
2. korlátként épülnek be az információs rendszerbe (az adatok csak a szabványos ellenőrzések – hossz, típus, értékhatár – utáa kerülnek rögzítésre) és
3. tájékoztatják a felhasználókat (mit, hol és milyen formában kell keresni).

Az elismert tudományágak illetve tudományok már régóta kiépítették szabványaikat, illetve szabványrendszerüket. Az informatikában csak most szorgalmazzák komolyabban a szabványok (szabványrendszer) kialakítását: talán ezzel magyarázható, hogy az informatikát még nem ismerik el egységesen önnálló tudománynak.

4.3.AZ INFORMÁCIÓS RENDSZER VETÜLETEI

Az információs rendszer, mint tudjuk inhomogén rendszer, konkrét és absztrakt tényezők szervezett együttese. A konkrét elemek közé a valóságban létező tényők tartoznak, így a technikai eszközök, a szabványok és a felhasználók. Az absztrakt tényezők az emberi gondolkodás eredményét, termékeit ölelik fel, amelyek a valós világ minél hübb ábrázolására hivatottak, ezek pedig az adatok, az események és a tevékenységek. Míg a konkrét tényezők fizikailag adottak, addig az absztraktakat és a közöttük lévő kapcsolatokat ki kell gondolni, fel kell ismerni, meg kell fogalmazni, el kell nevezni. Az előzőekből már adódik egy vetületrendszer (konkrét és absztrakt tényezők vetülete), de az adatok eltérő viselkedése (mássága) a tevékenységekhez és az eseményekhez viszonyítva egy harmadik vetület bevezetését követeli. Az információs rendszer három vetülete így a következő:

1. adatvetület,
2. feldolgozásvetület és
3. környezeti vetület.

Vegyük észre, hogy egy-egy vállalatban a használt fogalomrendszer (az adatok) igencsak időtálló (stabilak). A gyakori adatkiegészítések és módosítások inkább fejlesztési

hiányosságokra, mint az adatok változékonyságára utal. Az adatvetület viszonylag objektív, tárgyilagos. Az adatok függnék legkevésbé az információs rendszer többi tényezőjétől (még a hozzájuk legközelebb álló feldolgozásoktól is nagymértékben függetlenek).

A feldolgozásvetület az adatvetülethez viszonyítva változékonnyabb, kevésbé tárgyilagos, és nem is olyan független az információs rendszer többi tényezőjétől, mint az adatvetület. A feldolgozásvetület instabilitását a felhasználó jelenlétének tudhatjuk be, aki újabbnál újabb feldolgozásokat (tevékenységeket, jelentéseket) és eseményeket gondolhat ki az adatvetület megbotlygatása nélkül. Az adathoz viszonyítva a feldolgozás többszörös is lehet (ugyanazokat az adatokat többféleképpen is fel lehet dolgozni, ki lehet mutatni). A feldolgozás változékonysága azonban nem szabad, hogy kihasson az adatvetületre (adatszerkezetre), és ez az állítás, **a logikai adatfüggetlenség elve** szerint fordítva is igaz kell, hogy legyen: az adatszerkezeti változásoknak nem szabad befolyásolniuk a feldolgozásokat.

A környezeti vetület az információs rendszer valós tényezőit fogja össze: a felhasználókat, az erőforrásokat (idő, pénz, hardver, szoftver) valamint a szabványokat és eljárásokat. A felhasználóknak a tárgyilagostól sokszor távol álló (szubjektív) kérelmei, a hardver és a szoftver behatárolt lehetőségei, és a gyakran kvázi-tárgyilagos szabványok megakadályozzák a rendszertevezőt abban, hogy a tudása szerinti lehető legjobb információs rendszert alkossa meg. A környezeti vetület változékonysága ugyancsak egy olyan tényező, amellyel számolni kell. **A fizikai adatfüggetlenség elve** azonban kimondja, hogy az eszközök változásának nem szabad kihatnia az adatstruktúrára és fordítva: az adatszerkezet változása ne eredményezze az eszközök megváltoztatását.

4.4. AZ INFORMÁCIÓS RENDSZER SZINTJEI

Az adat-, feldolgozás- és környezeti vetületet tekinthetjük az információs rendszerek tényezőinek horizontális (egymás mellé rendelt) csoportosítását. Tudjuk azonban, hogy ezek a vetületek nem függetlenek egymástól, hiszen egymással kölcsönösen összefüggő tényezőkből állnak. Ahogy a vetületek között, úgy az egyes vetületek tényezői között is kapcsolatok vannak. Az adat- és feldolgozásvetület tényezői (elemei) tudatos (nem ösztönös) emberi tevékenység eredményei, mégpedig különböző elvonatkoztatási szinteken. A környezeti vetületnek, mivel objektíven létező dolgokat ölel fel, csak egy absztrakciós szintje van, a fizikai szint. A másik két vetületnek azonban a fizikai mellett két újabb elvonatkoztatási szintje is van. Ezek szerint az információs rendszernek három absztrakciós szintjét különböztetjük meg:

1. fogalmi (konceptuális),
2. logikai és
3. fizikai szint.

A fogalmi szint a valós világ információs rendszerrel leírandó részét, a lényegét, a tartalmat ábrázolja, maga a megoldás a logikai és fizikai szinteken ölt testet. A három szint egymástól függetlenül is fejleszthető, sőt csakis úgy kell, vagy kellene fejleszteni. A három elvonatkoztatási szint figyelmen kívül hagyása a tartalom és a megoldás egyvelegét eredményezi, ami semmiképpen sem előnyös. A fejlesztés folyamás általában a felhasználó szemüvegén keresztül valóságkép a kiindulópont, és ez a kép legtöbbször a valóságnak csak egy torzított mása. Másrészt a fejlesztési csoport gyakran a technikai eszközök kalodájában ülve evezget (evickél) a cél felé, vagyis korlátoktól terhes megoldás születik (születhet). A fenti érvelés logikusan hangzik, amit a gyakorlat is igazol. A megrendelő joggal várja el, hogy a kérelmeit és észrevételeit vegyék figyelembe a fejlesztés folyamán, a fejlesztőknek pedig már a hardver- és szoftverkorlátokon jár az eszük. Mindez természetesen rendjén is volna, ha a fejlesztők nem zsúfolnák egyetlen tervbe a tartalmat és megoldást is az összes számukra ismert korlátokkal együtt. No, de miért baj ez? Azért, mert egy-egy vállalkozás lényege, tartalma sokkal kevésbé változékony, mint mondjuk a számítógépek és adatfeldolgozási eszközök. Amennyiben csak egy terv készül torzított valóságtartalommal, felhasználói igényekkel, eszközkorlátokkal, úgy a géppark felújításakor újra kell tervezni az egész információs rendszert.

Ha komolyan vesszük az információs rendszer három vetületének létjogosultságát és szükségességét, úgy három tervet kell készítenünk. A fogalmi (konceptuális) szinten a valóság hű tükrözése, a logikai szinten a felhasználó valóság-képének megalkotása, a fizikain pedig a számítógépen megvalósítható, a felhasználónak megfelelő megoldás a cél. A fogalmi szintről »lefelé« haladva az objektív valóság mindinkább »elferdül«, ami teljességgel érthető az előző gondolatmenet alapján.

Az adatvetület fejlesztése folyamán a rendszertervező első feladata a megrendelő-felhasználó által használt fogalmak összegyűjtése, értelmezése és rendszerezése. Az életben, vállalkozásban használt fogalmak egymással összefüggnek, ezeknek az összefüggéseknek a feltárása, illetve a fogalmi struktúra kialakítása és dokumentálása képezi a rendszertervező következő feladatát. Ez nem más, mint a fogalmi adatszerkezet kialakítása. A **fogalmi adatszerkezet** a valós világ hű, tárgyilagos, csorbítatlan képét tükrözi vissza. Más szóval a **fogalmi szintű adatterv** a valós világ (valós rendszer) hibátlan mása, visszatükröződése.

A felhasználó kérésére a rendszertervező tudatosan elrontja a valós rendszerről alkotott (a fogalmi szintű adattervvel ábrázolt) képet. A felhasználó igényeit, kéréseit felfoghatjuk úgy is,

mint valamiféle betartandó szubjektív vagy kvázi-szubjektív korlátokat. A rendszertervező persze tudatlanul is elferdítheti a valós világot. Ez akkor történik, ha nem öleli fel az információs rendszer összes tényezőit, vagy ha a valós rendszer a rendszertervező számára sem teljesen ismert. A **logikai adatszerkezet** a fogalmi adatszerkezetnek egy olyan változata, megváltoztatott alakja, amely a felhasználói igényeknek megfelelő korlátoknak is eleget tesz, esetleg a rendszertervező tévesen kialakított valóság-képéből ered. A **logikai szintű adatterv** a felhasználói igényeknek vagy a rendszertervező téves valóságérzékelésének megfelelő, de a valós világ csorbítatlan képétől eltérő ábrázolást tartalmaz.

Az információs rendszer fejlesztése folyamán a rendszertervezőknek számolniuk kell az adatfeldolgozási eszközök (hardver, szoftver) korlátaival és készek kell lenniük a kompromisszumok felkutatására és kimunkálására. A **fizikai adatszerkezet** az adatok ábrázolásmódjának és az adathordozókon való elhelyezésének a leírását tartalmazza, és gyakran **fizikai szintű adatterv**nek is nevezik.

Az adatterveket minhárom absztrakciós szintre kötelezően ki kell dolgozni, méghozzá az egyedüli logikus sorrendben: először a fogalmi (konceptuális), majd a logikai, végül a fizikai szintű adattervet.

A **feldolgozásvetület fejlesztése** a valós világ eseményeinek és az azokkal kapcsolatos tevékenységek feltérképezésével és elemzésével kezdődik. A valós világ eseményei információs eseményeket váltanak, ezek pedig beindítják az információs tevékenységeket. A megfogalmazott információs események és a velük kapcsolatos tevékenységek csupán a feladat megfogalmazását szolgálják és a feldolgozás fogalmi szintű lényegét (magát a megfogalmazott feladatot) jelzik. A **fogalmi szintű feldolgozásterv** tehát magát a feladatot fogalmazza meg, az információs rendszer keretén belül megvalósításra váró információs eseményeket és tevékenységeket.

Az egyérelműen megfogalmazott feladat többféle módon is megoldható. A feladat egy elvi megoldásának leírását (az elvégzendő műveletek megnevezése, megfogalmazása és eljárásokba való tömörítése) a **logikai szintű feldolgozásterv** tartalmazza. A logikai szintű feldolgozásterv a felhasználók kvázi-objektív, kvázi-szubjektív illetve szubjektív elvárásainak, az adatkezelő és –feldolgozó szoftver korlátaival valamint a rendszertervezők szokásainak keresztüzében alakul és formálódik. Egy fogalmi szintű (konceptuális) feldolgozástervnek (megfogalmazott feladatnak) természetesen több elvi megoldása is lehetséges.

A **fizikai szintű feldolgozásterv** mindenképpen eszközfüggő, és a logikai szintű feldolgozásterv, az elvi megoldás konkrét megvalósítását tartalmazza. Ide tartozik a beviteli képernyők és jelentések aprólékos, hardverfügő ismérveinek megfogalmazása, a programmodulok meghatározása, a menürendszer kinézetének leírása, stb.

A feldolgozásvetület mindhárom absztrakciós szintjére ki kell dolgozni a feldolgozsterveket. A tervek kidolgozásának sorrendje, mint az adatvetület esetében, itt is nagyon lényeges, és pontosan előírt. A rendszertervező először a feladatot kell, hogy megfogalmazza (fogalmi, konceptuális szint). Ez után következhet csak a felhasználónak is megfelelő vagy tetsző elvi megoldás (logikai szint) kialakítása, majd végül a megoldás konkrét leírása a fizikai eszközök vetületében (fizikai szint).

Az információs rendszerterv leírásában általában nem választják szét a három absztrakciós (elvonattatási) szintet sem az adat-, sem a feldolgozásveületben. A legismertebb és leggyakrabban használt tervezési eljárások is csak kétféle terv kidolgozását látják elő: a logikaiét és a fizikaiét. A fogalmi (konceptuális) szint kifejezés és annak magyarázata először az 1975-ben megjelent ANSI/SPARC adatbázis-architektúra leírásában látott napvilágot, ám ezt a szintet ritkán alkalmazzák az információs rendszertervezés gyakorlatában. Az adat- és feldolgozásvetület fent leírt absztrakciós szintjei, mint láttuk, egymásra épülnek: az alacsonyabb szintek a magasabbakból képezhetők le. A jó rendszerdokumentáció az össz felsorolt terv mellett tartalmazni fogja valamennyi korlát pontos leírását, a felmerült változatokat, az érveket és ellenérveket a változatokkal kapcsolatban, valamint az elfogadott változatot és az indokokat minden egyes leképzésre nézve. Másszóval ez azt jelenti, hogy a rendszertervező bölcsen teszi, ha lejegyzí hogyan is jutott el az elfogadott változat kiválasztásáig: melyek voltak a korlátozó körülmények, a megfogalmazott változatok és végül az okok, amelyek a döntéshez vezettek.

A vállalat ügyvitelének fejlesztéséhez, a mobilitáshoz, vagyis az ügyvitelbonyolítás felgyorsításához szükségszerű, hogy az információs rendszer fejlesztésére, az adatfeldolgozásra és az információs rendszer új feltételekhez való idomítására szánt idő folyamatosan rövidüljön. A fejlett informatikai országokban a változáskezelésre (az információs rendszer új feltételekhez való alakítására) is komolyan odafigyelnek, fejlesztésnek tekintik és külön névvel is illeték: change management.

Az információs rendszer fejlesztése gyorsabb, ha abszós szintek szerint történik, hiszen az egyes szintek tervei gyorsabban elkészülnek és ellenőrizhetők, mert nem keverednek bennük a valós világ elemei, a felhasználók igényei és az eszközök korlátai. A változások is gyorsabban átvezethetők a rendszerben a megrendelő-felhasználó nagy örömére. Ha csak a géppark (hardver), vagy az adatkezelő és feldolgozó szoftver változik, a fogalmi és logikai szintű tervek változatlanok maradnak: csak a fizikai adat- és feldolgozásterveket kell a változásokhoz igazítani (a logikai tervek alapján). Ez azt jelenti, hogy változni fog a tervdokumentáció azon része, amely korlátokra, a logikai-fizikai leképzésekre, a hozzájuk tartozó változatokra, az elfogadott változatra és az indokokra vonatkozik. Ha a felhasználói igények módosulnak, akkor a

változások a logikai és fizikai terveket fogják érinteni, és csak a fogalmi szintű tervek nem változnak. Újrafogalmazódnak a fogalmi-logikai és logikai-fizikai leképezések is.

A klasszikus rendszertervezésnél csak egy terv készül. Benne átszövődnek a valós világ elemei a felhasználók információs kéréseivel és a hardver-szoftver eszközök korlátaival. Így mindenfajta változás az információs rendszer teljes újratervezését indítja el. Emellett rendszerint a tervben a korlátok meg sem jelennek egyértelműen (explicit módon), csupán az egyetlen megoldás: változatok és mindenfajta indoklás nélkül. A klasszikus rendszerterv dokumentációja nem tartalmazza a valós rendszer leírását, így bármilyen változás megvalósítása érdekében a valós rendszer megismerésével kezdődik a munka. Pedig pontosan a valós világ megismerése az információs rendszerfejlesztés legbonyolultabb és legidőigényesebb vállalkozása. A háromszintű tervezés megkönnyíti és felgyorsítja a fejlesztés (tervezés) és változáskezelés folyamatát növelve ezzel a vállalatok ügyvitelének mobilitását.

5. TECHNIKAI ESZKÖZÖK – HARDVER ÉS SZOFTVER

A fejlődés alapvető mozgatórugója a profit, illetve a pénz¹ valamint a piacgazdaság. A profitszerzés érdekében a cégek kénytelenek jobb minőségű termékeket piacra dobni, mint amit a konkurencia árul, és emellett igényes reklámkampánnyal hirdetni a termékeket. Számtalan példa bizonyítja az ismert szólást, hogy “Jó bornak is kell a cégér”: jóminőségű termékek “fuccsoltak be” a piacon a gyér marketing-támogatás következtében. A mai sikeres vállalatok előkelő helyzetüket a tudományosan megalapozott “megérzéseiknek” és futuroológusaiknak (jövőkutatóiknak) köszönhetik. Manapság nem elég csupán jóminőségű és keresett árut termelni és forgalmazni. Ki kell találni a vásárlók ízlését és igényeit, mi több, irányítani kell az ízlésüket és igényeiket. Az utóbbi időben kitapintható az irányelv, miszerint a termékeket szolgáltatásokkal “gazdagítva” kínálják a piacon a vásárlók nagyobb kényelmét szolgálva, de nem utolsósorban a nagyobb profit reményében.

A vállalkozások léte és fejlődése mindinkább informáltsági fokuktól függ, vagyis attól, hogy mennyire képesek a külső és belső környezetükben keletkező adatokat, információkat begyűjteni, feldolgozni, és azoknak megfelelően cselekedni. A nagyobbfokú informáltság elérése gyorsabb adatfelvételt követel, és intelligensebb szoftvereket az adatok minél gyorsabb feldolgozására. Mindez persze a gyorsabb döntéshozatal érdekében fontos. A mobilitás tehát közvetve gyorsabb hardvert és intelligensebb (bonyolultabb) szoftvert igényel.

A hardver és a szoftver fejlődésirányát a fent említett elvárások határozzák meg. Rá kell mutatni, azonban, a hardver és szoftver fejlődésében kitapintható kettősségre. A hardver terén egyrészt a nagyteljesítményű, hibatűrő rendszerek fejlesztése jellemző (ahol kizárólag a feldolgozás sebessége – a throughput – a mérce), másrészt a feldolgozási egységek s velük együtt a komponensek miniaturizálása és energiaigényük lecsökkentése (ezzel párhuzamosan a használatuk és karbantartásuk leegyszerűsítése) az elsődleges cél. Az első fejlesztési vonal az üzleti világ asztali (helyhez kötött) gépeit célozza meg, amelyek helyi hálózatokban szerverként vagy egyszerű hálózati számítógépként, nagyobb számítógépek terminálhálózatának irányítóiként (front-end machine), háttérben működő adatfeldolgozó számítógépként (back-end machine, database engine), valamint a szuperszámítógépeket, amelyeket kutatási célokra és matematikai modellezésre használnak. A hardverfejlesztés második ága a hordozható, mobil eszközökre terjed ki: a laptopokra, notebookokra, a zseb- és kéziszámitógépekre valamint a mobiltelefonokra. Ide tartoznak még bizonyos szempontból az egyszerű Internet-gépek: az NC-k (NetworkComputer) és NetPC-k, amelyek, de nem a méreteiknek, hanem inkább egyszerű

felépítésüknek (nincs hajlékony- és merevlemezük, sem CD- vagy DVD-egységük), használatuknak és karbantartásuknak köszönve.

A szoftver-fejlesztés terén ugyancsak két irányzat ismerhető fel, amelyek a hardver-fejlesztés irányait követik. A helyhez kötött gépek számára készülő szoftver folyamatosan nagyobb tárigényűvé és bonyolultabbá válik. Az egyre bonyolultabbá váló szoftvertermékek kezelése azonban nem bonyolódik, sőt, inkább egyszerűsödik a felhasználói felületek áttekinthetőségének és az interaktív súgóknak köszönve. A második fejlesztési irány szoftver terén az egyszerűsítés felé hajlik, hiszen a mobil eszközök szűkülnek a processzor teljesítőképességben, központi és külső tárhelykapacitásban, be- és kimenet egységekben, stb. Itt csak a szoftverek alapfunkciói kapnak helyet (maradnak meg), amelyeknek hardveréhsége is visszafogottabb. A szoftver-világ egyéni sajátossággal is rendelkezik a hardverrel szemben: léteznek ingyenes teljesértékű (befejezett, tesztelt, jól működő) szoftverek. Ez a szoftver sajátosságos jellegéből ered.

A szoftver és hardver kölcsönhatása régóta ismert tény. A fejlődés folyamán kölcsönösen segítik egymást az előrehaladásban: az egyik téren megjelent újítások rendszerint gyutacsaként szolgálnak a fejlődés kirobbantására a másik téren. A hardver-fejlesztések (gyorsabb processzorok és memóriaegységek, a gyorsabb kommunikáció) nagyobb és bonyolultabb szoftvertermékek végrehajtását teszik lehetővé, a szoftverek újabb verziói pedig rendszerint igényesebb hardvert kérnek a régi feldolgozási sebességek megtartásához. Általános az a vélemény, hogy manapság a szoftveripar nagy lemaradást mutat a fejlődésben a hardverhez képest. Másszóval a szoftver krízise folytatódik. A szoftver krízise egyes vélemények szerint kb. harminc évvel ezelőtti keletű. Azóta csak két alkalommal történt meg, hogy a lemaradást a szoftverfejlesztés csökkentette volna: a hetvenes évek végén a strukturált tervezés és – programozás bevezetésekor, valamint a nyolcvanas évek végén egy hatvanas években felvetett ötlet, az objektumorientáltság (elemzés, tervezés és programozás) kialakításakor. Ma még nem teljesen egyeduralkodó az objektumorientált szemléletmód, és, habár új elgondolások jelentek már meg, még nem érezhető egy olyan új irányelv, paradigma, amely kihívást jelentene és elindíthatna egy jelentős hardverfejlesztési hullámot.

A kommunikáció (és itt nem a közvetlen ember-ember kapcsolatra gondolunk) ma már a hardver és szoftver eszközök együttműködésének eredménye. Komoly pénzeszközöket fektettek be az új kommunikációs médiumok felkutatására. Ennek eredményeként ma már többfajta lehetőség is kínálkozik a jelátvitelre, és ezeket a lehetőségeket egyre széleskörűbben alkalmazzák is. Róluk a későbbiekben több szó esik majd.

¹ Many makes the World's go around.

5.1. HARDVER

Ha a mobilitást szélesebb fogalomként értelmezzük (a környezet változására adott gyors válaszként), úgy a hardverfejlesztés első vonala is elősegítője a mobilitásnak. Amennyiben viszont a mobilitást csak a hordozható eszközök fejlesztésére szűkítjük, akkor csak a hardverfejlesztés második fajtája tekinthető a mobilitás támogatójának. A mobilitás tágabb fogalmát alapul véve rövid kitekintést teszünk a számítógép-architektúra fejlődés területére (nem kizárólag a PC-kre vonatkozóan), a feldolgozás sebességének növelése (a feldolgozási idő csökkenése) szempontjából.

5.1.1. Számítógép-architektúrák

A szokványos PC konfigurációk már jól ismertek a mai középiskolások számára is, hiszen megismerkednek velük már az általános iskolában. A PC-k hardver-felépítése, de főleg teljesítőképessége mindinkább a régi (húsz-harminc évvel ezelőtti) idők nagygépeinek (mainframe) és szuperszámítógépeinek nyomdokain halad. A PC-gépek teljesítményjavítására irányuló törekvések valószínűleg a régi nagygépek fejlesztésvonalát fogják követni: a párhuzamosított feldolgozást megvalósító masszívan párhuzamosan feldolgozó (MPP – Massively Parallel Processors) számítógépek, a szimmetrikus többprocesszoros rendszerek (SMP – Symmetric Multiprocessor Systems), a vektorarchitektúrák és mainframe számítógépek.

5.1.1.1. Masszívan párhuzamos feldolgozás – MPP számítógépek

Az MPP szuperszámítógép tulajdonságokkal rendelkező, speciális (egyedi) tervezésű, gyors működésű hálózatban működő többszáz vagy akár több ezer számítógép sajátos működését jelöli. Ez az MPP gépeknél alkalmazott hálózatépítési filozófia egyedi alkatrészek nélküli alkalmazása is honos és munkaállomás-fürt (Cluster of Workstations – COW) elnevezésen ismert az irodalomban.

Az MPP rendszerek csomópontjaiban legtöbbször általános célú RISC processzorok helyezkednek el saját (közvetlen csatlakozású) memóriával és I/O egységekkel, így a bővítés is aránylag egyszerű művelet: az újabb csomópontok egyszerű hozzáadásával történik (a csomópontokat összekötő hálózatnak természetesen bővíthetőnek kell lennie). Ez a megoldás

optimalizálja a feldolgozás folyamatát a gyártó cég által biztosított programkönyvtár segítségével. A nagyszámú csomópont jelenléte miatt nem létesül fizikai kapcsolat minden csomópont között, de a kapcsolatot az operációs rendszer szükség esetén szoftverileg biztosítani tudja (mindez persze transzparens – észrevétlen marad a felhasználó és programozó számára).

A lokális, csomópontokban elhelyezkedő memóriamodulok nem alkotnak egységes (folyamatos) memóriaterületet. A processzorok más csomópontok memóriaterületeihez (memória-címtartományaihoz) csak üzenetek segítségével (és nem memóriautasítások végrehajtásával) jutnak el. Az ilyen megoldást NORMA (No Remote Memory Access – NORMA) felépítésnek is nevezik, és megvan az az előnye, hogy nem igényel hardveres támogatást a memóriakonvizisztencia megőrzésére.

Az MPP rendszerek csomópontjaiban lévő számítógépek operációs rendszerei csupán a lokális feladatok ellátására képes (operációs) rendszer-magok, amelyeket a központi csomóponton futó operációs rendszer irányít és ütemez, emellett a feladatokat is elosztja a csomópontok között. A NORMA felépítés következtében a távoli memória elérése kifejezetten lassú művelet, ezért a feladat megfelelő tagolása és elosztása a számítási csomópontok között különös jelentőséggel bír a feldolgozási idő szempontjából. A gyakorlatban olyan fejlesztőeszközök használatosak ezeknél a rendszereknél, amelyek alkalmasak a feladat felépítését az MPP struktúrájához igazítani. Ezek közé tartozik a FORTRAN programnyelv egy új verziója, a HPF, amely a párhuzamosan végrehajtható kódrészek kijelölésére is alkalmas. A csomópontokban működő operációs rendszerek legtöbbször az UNIX család tagjai.

További érdekességek az MPP architektúráról az alábbi Internet címeken találhatók: www.top500.org (az ötszáz legjobb szuperszámítógép adatai találhatóak meg itt), www.computer.org/parascope/ (az IEEE számítógépes társaságának szuperszámítógépekkel foglalkozó lapja) i www.sandia.gov/ASCI/ASCI-links.htm (az amerikai Sandia laboratórium a nagyteljesítményű számítógépekkel kapcsolatos információk begyűjtésével, rendszerezésével és feldolgozásával foglalkozik).

5.1.1.2.Szimmetrikus multiprocesszoros rendszerek – SMP

A szimmetrikus multiprocesszoros rendszerek kettőtől néhány száz processzort tartalmazhatnak, vagyis az MPP rendszerektől kisebb számítógép-rendszerről van szó. Az SMP-ben a processzorok azonosak, és nem megkülönböztetett szerepűek. Egyenrangúan használják az összes rendelkezésre álló számítógép-erőforrást: a központi tárat (operatív memóriát), külső

tárakat és I/O egységeket. A processzorokon azonos operációs rendszerek futnak, de a közös használatú erőforrások miatt a szinkronizáció elengedhetetlen. A processzorok mindenemű célhardver és üzenetek nélkül hozzáférnek a teljes memória-címtartományhoz, a processzorközi kommunikációra osztott memóriaterületeket használnak. Az osztott memóriaterületek szokványos szoftver-szinkronizációja mellett szükség van a processzorok mellett elhelyezkedő cache-memóriák tartalmának aktualitásvizsgálatára (hardveres úton oldják meg általában). A processzorok cache-vezérlője figyeli a memóriabusz forgalmát (bus snooping), és szükség szerint (ha a központi tár része, amelyet az illető cache is tartalmaz megváltozik) megváltoztatja a cache állapotát (jelezve ezzel, hogy a cache már nem az aktuális memória-tartalmat őrzi).

A feldolgozások sebességét az SMP architektúrák esetében is párhuzamosítással tudják növelni. A feldolgozás sebességét a végrehajtandó program párhuzamos feldolgozásra alkalmatlan részének aránya (a program teljes címtartományához viszonyítva) korlátozza. Ezt a megállapítást gyakran Amdahl-törvénynek nevezik

Ha az SMP rendszerekben a processzorok száma jóval a tíz fölé emelkedik, akkor a közös erőforrások (különösen a buszrendszer) szűk kersztmetszetté válnak. Ilyen esetekben a memória nem egy, hanem több (általában négy) busz segítségével közelíthető meg. Az SMP megoldásokban nemkívánatosak (és veszélyesek) a hardver- és szoftverhibák. A közös erőforráshasználat miatt egy-egy egység meghibásodása komoly zavarokhoz vezethet a feldolgozásban. A programozás viszont egyszerűbb ennél a megoldásnál, mint az MPP rendszerben.

Az MPP és SMP filozófiát gyakran kombinálják, aminek hibridmegoldások az eredményei. Ilyen az IBM SP számítógép-családja, ahol az MPP architektúrájú rendszer csomópontjai SMP felépítésűek. Ha a feladat struktúrája megfelel az MPP rendszer felépítésének, akkor az SMP csomópontok gyorsabb feldolgozóképesége érezhetően kihat az egész feladat megoldására (ami a feldolgozásra szükséges idő csökkenésében nyilvánul meg). Ha azonban a feladat felépítése nagyban eltér az MPP struktúrájától, úgy intenzív kommunikáció bonyolódik a csomópontok között, és az SMP felépítésű csomópontok gyorsabb feldolgozóképesége nem gyorsíthat lényegesen a feldolgozás egészén (hiszen azt a nagyszámú üzenetváltás miatt a kommunikációs hálózat átviteli sebessége határozza meg).

5.1.1.3.Vektorarchitektúrák

Vektorprocesszorok alkalmazását a hetvenes évek közepétől lehet nyomonkövetni. Ezeknél a töbprocesszoros gépeknél speciális hardvertámogatás segítségével nagyszámú (akár több Kbyte méretű) adaton végezhető el egyidőben ugyanaz az elemi művelet). Egy-egy processzor általában nyolc egyenként 64-től 256 lebegőpontos elemet tartalmazó vektorregiszterrel rendelkezik. A vektorarchitektúrájú számítógépek a mátrix- és vektorműveletek elvégzésében verhetetlenek (innen származik a nevük is). A lineáris algebrai műveleteknél gyakran előfordul a vektorműveletek halmozódása, így a vektorregiszterek láncolása további gyorsítást eredményez a feldolgozásban. Ilyenkor az első vektorműveletből kilépő vektorelemek közvetlenül a második műveletet végző vektorpipeline-ba kerülnek, mielőtt még az első művelet teljesen befejeződne. A vektorprocesszor a vektorpipeline minden ütemének végrehajtásakor új adattal dolgozik, amit a memóriából kap meg. A memóriamodulok számának kiegyenlítése a vektorpipeline ütemszámával lehetővé teszi, hogy minden ütemvégrehajtáskor egy újabb memóriamodulhoz forduljon a processzor az adatokért, ami további feldolgozásgyorsulást jelent. A nagyszámú memóriamodul miatt gyorsítótárakat (cache) általában nem alkalmaznak vektorarchitektúrájú számítógépek esetében. A processzorok CISC típusúak és többszörözött hardver elemek valamint művelettípusok szerepelnek a vektorutasítások gyors végrehajtására.

Sajátos felépítésük miatt a vektorarchitektúrájú szuperszámítógépek csak a mátrix- és vektorműveletek gyors elvégzésében jeleskednek, de csupán abban az esetben, ha a programok felépítésüknek megfelelően íródtak. A programok szerkesztése nem olyan bonyolult, mint az MPP rendszerek esetében. A skalárfeldolgozások vektorszámítógéppel való végrehajtása mindenképpen drágább, mintha azt egyszerű számítógépen végeznénk.

Sok érdekesség található a vektorarchitektúrákról a következő Internet címen: www.cray.com (a legismertebb szuperszámítógépgyártó cég vektorszámítógépeket is gyárt), de a következő cégek palettáján is megtalálhatók a mátrix- és vektorfeldolgozásokat kedvelő gépek www.nec.com , www.fujitsu.com i www.hitachi.com .

5.1.1.4.Mainframek

A Mainframe számítógépek erős, robusztus, gyors számítógéprendszerek, amelyek központosított (centralizált) adatfeldolgozásra készülnek, de egyidőben többszáz vagy akár

többször intelligencia nélküli terminál kiszolgálására is alkalmasak (képesek). Igaz ugyan, hogy tíz évvel ezelőtt a szakemberek és futurológusok a központosított adatfeldolgozás teljes megszűnését jósolták, ezek a gépek most is jelen vannak a piacon, sőt, az érdeklődés évről évre növekszik irántuk. A mainframek tranzakció-beállítottságuk úgy szoftver, mint hardver téren. A tranzakció olyan teljességében elvégzendő műveletsorozat, melynek sikeres befejezése után eredményei elmentődnek. A részlegesen végrehajtott (megszakított) tranzakciók a végrehajtásuk előtti állapot visszaállítását váltják ki. A tranzakciós feldolgozás mindig konzistens állapotban állapodik meg: vagy a sikeres tranzakció utáni, vagy a megszakított tranzakció következtében visszagörgetett előző állapotban. A hardver szintjén is tranzakciókhoz hasonló megoldásokat alkalmaznak: ha egy hardver-egység meghibásodik, úgy azt kiegészítő, tartalékegységgel pótolja automatikusan a rendszer. Az IBM S/390 rendszereknél processzorhiba (téves működés) esetén a feladatot egy tartalék processzor veszi át és fejezi be a feldolgozást (természetesen a felhasználó tudta és beavatkozása nélkül).

A mainframe számítógépek gazdag utasításkészlettel rendelkező CISC processzorokat tartalmaznak a legrégebben (húsz-harminc éve) gyártott és működő számítógépekkel biztosítandó kompatibilitás miatt. A leginkább SMP (szimmetrikus) működésmódban dolgozó mainframe számítógépek processzorszámuk ritkán haladja meg a 14-et (az Amdahl Millennium 16 processzorig bővíthető). A rendszer indításakor történik a "szerepek kiosztása" a processzorok között: az adatfeldolgozást végző, klasszikus értelemben vett CPU-k mellett I/O kiszolgáló processzorok (SAP – System Assist Processor) is működnek, néhány processzor pedig tartalékként (SPU – Spare Processing Unit) készenléti állapotban marad. Az IBM S/390 processzora két, egymástól teljesen független utasításvégrehajtó egységgel dolgozik, és minden utasítás végrehajtása után belső ellenőrzés következik. Ha az eredmények különböznek, a feldolgozást a hibázó processzor még egyszer megismétli, de ha a következő ellenőrzés is hibát jelez, úgy a feldolgozást egy, a CPU-k által közösen kijelölt SPU – tartalékprocesszor veszi át. A feldolgozás és felhasználó számára mindez észrevétlen marad.

Az említett S/390-es processzorok utasításkészlete egyedi hierarchiával bír: a szokványos mikrokód mellett egy újabb, „köztes szinttel is rendelkezik, ez pedig a millikód. A millikód-utasítások több mikrokód utasítást felölelő, durván az assembly nyelvek makróinak megfelelő szintű egységek. A processzorok (processzor-modulok), természetesen működés közben is cserélhetők.

Az egyedi tervezésű és kivitelezésű, mainframe rendszerekben használatos processzorok nem hasonlíthatóak össze asorozatban gyártott, és a kereskedelemben kapható CTOS (CTOS – Commercial Off-The-Shelf) processzorokkal. A mainframe gépek többi alkatrésze is egyedi

tervezésű, és mindemellett az állandó rendelkezésre állás (nagyfokú megbízhatóság) elérése érdekében az alkatrészek számát többszörözik. A számítógép működésében közreműködő alkatrészek munkáját folyamatosan ellenőrzik, és a hibás alkatrészt feldolgozás közben lecserélik a tartalékok egyikével. Ez történik úgy a memória, mint a gyorsítótár esetében is. Mainframek esetében gyakran találkozhatunk egyedi tervezésű ellenőrző rendszerekkel is annak érdekében, hogy a meghibásodások között eltelt idő – MTBF (MTBF – Mean Time Between Failure) várható értéke minél kisebb legyen. Így elérhető egy írigylésre méltó évi 5-10 perces állásidő is. A hardver és szoftver átkonfigurálása a hibás alkatrészek felfedezése esetén munka közben történik, a rendszer leállása nélkül.

Az adatbiztonság és adatvédelem biztosítása érdekében az IBM S/390 mainframekhez kriptográfiai koprocesszort is vásárolható. Külső vektoregységek is csatlakoztathatók a rendszerhez, bár a mátrix- és vektorszámítások sebessége elmarad a valódi vektorprocesszorokétól, hiszen a nagyszámú modulra tagolt központi tár fogalma ismeretlen a mainframeknél. E rövid áttekintést a következő WEB-oldalakon lehet elmélyíteni: www.ibm.com - az IBM a legnagyobb mainframe-gyártó, a mainframe-klónok legismertebb előállítói-forgalmazói pedig az Amdahl és Hitachi: www.amdahl.com és www.hitachi.com a Fujitsu (www.fujitsu.com) is ismert IBM kompatibilis mainframeket gyártó cég (PCM – Plug-Compatible Manufacturer – a klóngyártó cégek közös neve).

5.1.1.5. Egyedi (Specifikus) architektúrák

Az egyedi architektúrák keretében lássuk miből épül fel az Eötvös Lóránd Tudományegyetemen (ELTE) “kifőzött” SZESZ (Szegény Ember Szuperszámítógépe – PMCP – Poor Man’s Supercomputer). Felépítését illetően gépfürtről van szó (munkaállomás-fürt – Cluster of Workstations – COW), a tervezett csomópontok száma 128. Valamennyi csomópont egy 450 MHz-en dolgozó AMD K6-os processzort tartalmaz, mellette 120 MB központi tárat és 2,1 GB-os merevlemezt. A gépfürtökre jellemző Ethernet gyűrű mellett azonban a SZESZ rendelkezik egy egyedi tervezésű, az MPP architektúrájánál alkalmazott hálózatra hasonlító gyors kommunikációs átviteli rendszerre. A jelenleg 32 csomópontot számláló SZESZ 20.000 US\$ értékű alkatrészt foglal magába, és az egymillió dollárt érő Cray szuperszámítógép teljesítményének 16 %-át éri el. A csomópontokban szokványos PC alkatrészek dolgoznak a Linux operációs rendszer felügyelete alatt. A feladatok Fortran és C nyelven kerülnek a gépre

meoldás, feldolgozás céljából. A gyors kommunikációs hálózat munkáját külön processzor irányítja egy, erre a célra kifejlesztett protokoll szerint. Bővebb információ a SZESZ-ről a <http://xxx.lanl.gov.aps/hep-lat/9912059> Internet-címen található.

A franciaországi Atomenergia Bizottság (CEA) a Compaq cégtől egy 5 teraflop kapacitású vagy sebességű (ötbillió lebegőpontos művelet másodpercenként) szuperszámítógépet rendelt meg. Ez a rendszer hétszer gyorsabb az Európában most működő leggyorsabb számítógépeknél. A számítógép MPP architektúrája a csomópontokban SMP felépítésű lesz, pontosabban a csomópontokban a Compaq Alpha Server SC Supercomputer jelzésű szuperszámítógépei helyezkednek majd el, 64 bites Alpha (RISC) processzorokkal. A csomópontok operációs rendszere a hatvannégybites Tru64 Unix, a csomópontokat összekötő hálózat pedig egyedi tervezésű lesz. A több mint 2000 processzort működtető szuperszámítógép várhatóan 2001 végén kezdi meg feladatai elvégzését (www.compaq.com).

Az IBM 2000. elején jelentette meg adatait a Blue Gene (Kék Gén) névre keresztelt szuperszámítógépről, amelyet a tervek szerint öt évig építenek majd (2005. végére készül el). A rendszer több mint egymillió processzort fog egybe, teljesítménye pedig a tervek szerint 1 petaflop (10^{15} flop) lesz. A számítógéprendszer új, SMASH (Simple, Many and Self-Healing) filozófia vagy architektúra szerint épül, ami a processzorok drasztikus tehermentesítését fogja biztosítani (ez viszont a méret és energiafölvétel csökkenését jelenti). Növekszik a processzorok egyidejű (párhuzamos) működésének lehetősége (a mai párhuzamosan végrehajtható szálak száma ötezer, a Blue Gene nyolcmillió szálal tud majd párhuzamosan és egyidőben feldolgozni), és a rendszer önstabilizáló és öngyógyító algoritmusokkal fog rendelkezni, amelyek kiiktatják és eltávolítják a meghibásodott rendszerkomponenseket a szakemberek külső beavatkozása nélkül (www.ibm.com).

5.1.2. Alaplapok

A leggyakrabban alkalmazott nem PC-s rendszerekarchitektúrák rövid ismertetője után visszatérünk a PC-k világába. Ezeket a számítógépeket alkalmazzák ugyanis leginkább manapság az információs rendszerek fejlesztésére és adatok feldolgozására egyaránt.

A számítógépek kiválasztásánál legtöbbször csak a processzorok sebességét (teljesítményét) és/vagy típusát, a központi tárterület nagyságát, a grafikus kártya típusát/lehetőségeit és a merevlemez(ek) összkapacitását veszik figyelembe. Az alaplapra

kevesen fordítanak egyáltalán figyelmet, annak ellenére, hogy általában tőle függ a rendszer stabilitása és bővítési lehetősége.

Az alaplapok kiépítésében útmutató irányelveként ma az integráltság fokozását jelölhetnénk meg. Egyre nagyobb számú vezérlőegység kerül az alaplagra felváltva az eddig használt vezérlőkártyákat. Ilyenek az IDE (Integrated Device Equipment) és EIDE (Enhanced Integrated Device Equipment) vezérlők a lemezegységek, a hang- és videokártyák számára, a hálózati kártyák, modemek stb. Az alaplagra integrált vezérlőegységek esetében nem jelentkeznek kompatibilitási problémák (azokat már az alaplap gyártók kiszűrték).

Az alaplap működését egy áramkör-család irányítja amit ma egyszerűen csak chipsetnek vagy chipkészletnek neveznek. Minden chipkészletnek megvannak a maga sajátosságai, és általában a piacra kerülő új processzorok új chipkészletet igényelnek. A legnagyobb chipkészlet-gyártók: az (www.intel.com), a Via (www.viahardware.com) és az Ali (www.ali.com). A legismertebb és leggyakrabban használt chipkészletek a következők: Intel 82440BX, Intel 82440ZX, Intel 82810E (i810E), Intel 82815E (i815E), Camino (i820), i840, i850 (a Pentium IV processzor számára), Via Apollo KX133, Via Apollo Pro 133, Via Apollo Pro Plus, Ali Aladdin TNT2 GFX Pro, AMD 750.

Az alaplap biztosítja az órajelet (Real Time Clock) a processzor számára. A processzor működési sebessége az alaplap órajelétől (amely egyben a processzor bemeneti órajelét képezi és FSB - Front Side Bus jelölést kapott) és a processzoron belüli órajelszorzótól függ. A szabványos FSB frekvenciák a következők: 66, 100 és 133 MHz. A legtöbb alaplapon a szabványostól eltérő FSB frekvenciák is beállíthatók a 66 és 150 (200) MHz-es határok között. Az FSB frekvencia kontinuális változtatása a BIOS segítségével valósítható meg, a diszkrét (meghatározott lépték szerinti) állítás pedig jumperek, DIP-kapcsolók és ugyancsak a BIOS alkalmazásával. A processzoron belüli órajelszorzó szintén megváltoztatható az arra előrelátott jumperek DIP-kapcsolók, esetleg a BIOS segítségével. A processzor bármilyen módon kieszközölt "túlhajtása" emeli annak hőmérsékletét, ami a processzor károsodásához (leégéséhez) vezethet!

Az alaplapok processzorfoglalatai határt szabnak az alaplapon elhelyezhető processzoroknak. Az Intel processzorai a következő három fajta foglalatot használják: a Slot 1-et, a Socket 370-et, és a Socket 423-at (a Pentium IV), az AMD pedig a Socket A-t (Athlon) és a Socket 462-t (Duron és Thunderbird). Egyes alaplapokon kétféle processzorfoglalat is található (az Acorp 6VIA96A-n – Slot 1 és Socket 370; bár a Slot 1 a SlotKet átalakítóval Socket 370-né alakítható). A processzor tápfeszültsége (magfeszültsége) valamennyi alaplapon jumper vagy a

BIOS segítségével állítható, így többféle (különböző feszültséget igénylő) processzor is alkalmazható, ha a foglalat is megfelelő.

A bővítési lehetőségek hordozói az alaplapi csatolók, amelyek különböző vezérlőkkel ellátott bővítőkártyák befogadására szolgálnak. A csatolók típusa az alaplapi lehetőségeit határozzák meg: az ISA (Industry Standard Architecture), az EISA (Extended Industry Standard Architecture) és a VLB (VESA Local Bus) a múlt emlékei (igaz ugyan, hogy egy ISA csatolót ma is megtalálhatunk bármelyik alaplapon a régi alkatrészekkel való kompatibilitás megtartása miatt). A legelterjedtebb csatolótípus ma a PCI (Peripheral Component Interconnect) csatolópajzsz (slot), amely a VLB-hez hasonlóan 32 bites, órajele pedig 20 és 33 MHz között a processzor órajeléhez illeszthető. Az alaplapi csatolók száma 3 és 6 között változik a konkrét alaplaptól függően. Valamennyi bővítőkártya PCI típusú csatlakozóval. A grafikus lehetőségeket (nagy sávszélességet) igénylő alkalmazások számára a PCI busz túlságosan lassú, ezért fejlesztették ki az AGP (Accelerated Graphics Port) buszt egyszeres, kétszeres és négyszeres sebességű üzemmódjával (AGP, AGP2x és AGP4x). Segítségével a videovezérlő közvetlenül elérheti a központi tárat, és azt megosztva használhatja a programokkal. Az újabb chipkészletet tartalmazó alaplapon AMR (Audio/Modem Riser) csatoló is előfordul, amely a hangvezérlő és a modem csatlakoztatására szolgál (ezek a vezérlők akár teljességükben is az alaplapi lehetnek integrálva). A vezeték nélküli kapcsolatok megvalósítására is gondoltak az újabb alaplapi tervezői az IrDA (Infrared Data Association) szabványt és a bluetooth technológiát támogató egységek csatlakozóinak biztosításával.

A szokványos vezérlők mellett az alaplapi tartalmazhatnak különleges vezérlőegységeket is, amelyek megbízhatóbbak, stabilabbak és nagyobb átviteli sebességet biztosítanak az előbbieknél. A különleges vezérlők közé sorolhatók az SCSI (Small Computer System Interface), a RAID (Redundant Array of Inexpensive/Independent Disks), az UDMA (Ultra Direct Memory Access), az USB hubok, vagy az IEEE 1394 adatátviteli csatolók. Az SCSI-re nemcsak merevlemezek kapcsolhatók, hanem minden olyan más egység is amelyek képesek csatlakozni erre a fajta vezérlőre. A merevlemezen tárolt adatok elvesztésének kockázatát nagymértékben csökkenti a RAID technológia, amely a biztonsági másolatok automatikus elkészítése mellett gyorsabb adathozzáférési lehetőséget is nyújt (két lemezegység esetén a páratlan adatblokkokat az egyikre, a párosakat a másikra írja). A mai leggyarabban használatos merevlemez-interfacek az UDMA-családból kerülnek ki: UDMA33, UDMA66 és UDMA100, amelynek tagjai a neveikben megtalálható jelzéseknek megfelelően 33, 66 és 100 Mbyte/s adatátviteli sebességet biztosítanak. Egyre nagyobb az érdeklődés a sokeres vezeték felváltó, vékonyabb kábelt használó gyors adatátvitel iránt. Két szabvány is létezik, amely vékonyabb

kábelt használ és gyors átvitelt biztosít: az: USB és az IEEE 1394 (Firewire). A két megoldás hasonlít egymásra. Az USB (Universal Serial Bus) soros adatátvitelt valósít meg az üzemmódtól függően: 1,5 Mbit/s illetve 12 Mbit/s sebességen (a legújabb verzió, az USB 2.0 kb. 40-szer gyorsabb az adatátvitelben az említetténél). Az alacsonyabb energiafelvételű USB-re kapcsolt egységek nem igényelnek külön tápegységet, mert az USB vezérlő 0,5 A áramot tud biztosítani, és legfeljebb 127 egység kapcsolható rá. Az IEEE 1394 hasonló karakterisztikájú, de nagyobb átviteli sebességű (200Mbit/s) vezérlő, maximálisan 64 eszköz (egység) kapcsolható rá, és az alacsony áramfelvételű egységeknek tápot is biztosít.

Az alaplap fontos integrált áramköre a BIOS (Basic Input Output System) chip, amely tartalmazza a rendszerkonfiguráció teszteléséhez és a számítógép indulásához szükséges programokat. Egyes alaplap-gyártók két BIOS-áramkört is elhelyeznek az alaplapon. Ez akkor jelent valamit, ha a BIOS kiadásának dátuma után megjelent processzort szeretnénk használni (ehhez persze új BIOS-frissítés kell), és a BIOS-frissítés Internetes letöltése valami okból megszakad. Hibás tartalmú BIOS-szal ugyanis a számítógép nem tud »felébredni«, és ekkor jól jöhet a tartalék BIOS. A legújabb BIOS-frissítések gyártók Web-helyén érhetők el. Közülük talán a következő kettő a leglátogatottabb: www.award.com és www.ami.com.

Az újabb alaplapok hűtőventilátor-csatlakozóval is rendelkeznek. A processzor a legjobban melegedő egysége a gépnek, a benne levő tranzisztorok nagy száma és az integráltság fokának folyamatos növekedése miatt. Az egyes alaplapokon áramkörök felügyelnek a processzor hőmérsékletére és vezérlik a hűtőventilátorok forgási sebességét is. A hűtőbordák (amelyekre a ventilátort szerelik) átlagos elfogadható hőmérséklete 30 és 45 °C között mozog.

Az ATX szabvány szerint gyártott alaplapok lehetőséget adnak a szabványt ismerő és tiszteletben tartó alkatrészek számára egy újabb lehetőséget: a felhasználó által beállítható passzívan eltöltött idő után ezek az egységek energiatakarékos üzemmódba kerülnek (pl. a merevlemez forgása leáll, a képernyő elsötétül). Ennek eredményeképp az energiatakarékosság és a melegedés mérséklése mellett az alkatrészek élettartama is növekszik. Az egységek »ébredtése« többféle módon is történhet, amit a BIOS tesz lehetővé (ott állítható).

Az alaplapok legismertebb Web-helyei a következők:

www.bxboards.com ,

www.motherboards.com i

www.slota.com .

A legnagyobb alaplapgyártók betekintést engednek termékeik dokumentációjába. A leglátogatottabb WEB-helyek:

www.abit.com.tw/english/download/manual.htm ,
www.asus.com/Products/Motherboard/manual.html ,
www.epox.com/support/database.html ,
www.support.intel.com/support/motherboards ,
www.soltek.com.tw/English/download/manuals.htm ,
www.soyousa.com/manuals.html ,
www.supermicro.com/PRODUCT/UsersManual.htm i
www.zida.com/support/manual.htm .

5.1.3. Processzorok

A régmúlt idők fegyverténye, hogy a gépi utasításokat kizárólag hardver segítségével hajtják végre a processzorok mára már eledésbe merült. A bonyolultabb gépi utasításokat a processzorokban ma szoftveresen, mikroprogram végrehatásával végeztetik el. A processzorok felépítése nagyon eltérő: a mainframe számítógépek processzorai szinte nem is hasonlíthatóak a PC-k mikroprocesszoraihoz. Általánosan elfogadott a CISC (Complex Instruction Set Computer) processzorok megkülönböztetése a RISC (Reduced Instruction Set Computer) processzoroktól. Az előzőek (CISC processzorok) nagyszámú gépi utasítással rendelkeznek, sok bonyolult utasítással (ilyenek a PC számítógépek processzorai is). Az utóbbiak (RISC processzorok) lényegesen kisebb utasításpalettával rendelkeznek, de ezeket hardveresen hajtják végre, így gyorsabbak és eredményesebbek a CISC processzoroknál, természetesen drágábbak is.

A PC számítógépek 32 bites mikroprocesszora a konfiguráció többi egységével a csatornán keresztül tartja a kapcsolatot. Sokan nem is egy, de három csatornáról (bus) beszélnek a jól elhatárolható feladatköreik miatt: adatcsatorna (data bus), címcsoport (address bus) és vezérlőcsatorna (control bus). Az adatcsatorna 8, 16, 32 vagy 64 bit egyidejű átvitelére képes (ez a processzor szóhosszával magyarázható, ezek alapján beszélünk 8, 16, 32 vagy 64 bites mikroprocesszorokról). Ez egy kétirányú forgalmat bonyolító csatorna. A processzor csak az operatív tárból tud utasításokat és adatokat közvetlenül lekérni és visszaírni, de ha egyedül végzi el a bemeneti-kimeneti utasításokat, úgy a külső tárral és bemeneti-kimeneti egységekkel is elvégzi a kétirányú forgalmat az adatcsatornán keresztül. Az adatok átviteli sebessége a processzor és az operatív tár között Gbyte/s egységekben fejezhető ki. A címcsoporton keresztül 32 vagy 36 bit egyidejű átvitele történik, amellyel legfeljebb 64 Gbyte (2^{36}) tárcím generálható. Ezen a csatornán keresztül, ahogy a neve is jelzi a címek átvitele

történik, a processzortól a tárig, vagy más egységig. A vezérlőcsatorna kétirányú forgalmat biztosít az ellenőrző és vezérlő jelek átvitelére a feldolgozás egyedüli aktív eleme (a processzor) és a többi egységek között. A vezérlőcsatorna sávszélessége (a bitek száma) a processzor felépítésétől függően változhat, és elérheti a 64 bitet is.

A processzorokat gyakran csak órajelük gyakorisága (sebességük, illetve munkafrekvenciájuk) alapján minősítik. Ez azonban nem teljesen tárgyilagos mérce, mert a feldolgozás sebessége más tényezőktől is függ (az adat- és címcsatorna sebességétől és szélességétől). A processzorok minőségének megítélésében a következő adatokat ajánlatos figyelembe venni és mérlegelni: regisztereinek száma, hossza és rendeltetése, az aritmetikai (fix- és lebegőpontos) műveleteket végrehajtó egységek száma és végrehajtási sebessége, az utasítás-dekódoló- és végrehajtó egységek száma, az utasítás-végrehajtó egység(ek) felépítése és működésmódja (**a.** szuperskalár felépítés – több egység egyidőben más-más utasítást hajt végre, mint valamiféle belső többprocesszoros feldolgozás, **b.** pipeline – superpipeline elvű működés – az utasítás végrehajtása fázisokban történik, a fázisok száma a pipeline mélységét jelenti), a belső (ha a processzor szerves része) és külső (ha különálló áramkörként valósítják meg) gyorsító tárak (cache) szintjeinek száma (elsődleges, másodlagos), nagysága és sebessége, valamint a végrehajtható gépi utasítások készlete. A processzorfoglatat típusa, a magfeszültség értéke és a kidolgozás technológiája (milyen anyagon – félvezetőn, milyen »vastagságú« vezetékekkel és hány tranzistor kapcsolódik egyáshoz a processzoron belül) másodrangú jelentőséggel bírnak, melyek csak a melegedés várható mértékére és az alaplapra való csatlakozásra utalnak.

A 2000-s év kiemelkedő processzor-újdonságai az AMD üdvöskéi: a Duron (700 MHz), az Athlon és a Thunderbird (1,2GHz), az Intel igyekezete, a Pentium 4 (1,5GHz) valamint az IBM újdonsága, a Power4 voltak. A Power4 új architektúrájú (felépítésű) processzor, két processzormaggal és háromszintű belső gyorsítótárral.

Megjelentek az első 64 bites processzorok is: az 1994. júniusában Merced néven (az Intel és a HP által) beharangozott ígéret Itanium néven látta meg a napvilágot. Az AMD nem sokáig maradt le a versenyben, mert röviddel az Itanium megjelenése után bejelentette Sledgehammer nevű processzorának elkészültét. A megjelenés évében a kereskedelemben még nem voltak kaphatók ezek a processzorok, de az állítások szerint nem kell sokáig várni rájuk, és az alkalmazásukkal sem lesz gond, mert hardver szintű kompatibilitást emlegettek az Intel előző processzoraival, valamint az általuk használt x86 jelű szabványos utasításkészlettel.

A processzorgyártásba új szintet lopott az újonnan alakult (alapított) Transmeta nevű cég, a Crusoe processzorcsaládjával. Szoftver szinten teljesen megfelel a mai PC-k processzorainak,

viszont a hardver megoldása egészen eltérő filozófián nyugszik. A Crusoe processzorok 128 bit hosszú, ún. VLIW (Very Large Instruction Word) utasításokkal dolgoznak, az x86 utasítások fordítását VLIW alakra a Code Morphing Software végzi. A processzor felépítése sokkal egyszerűbb a Pentiumokétól, kevesebb a tranzisztorok száma a processzorban, és ezzel a melegeedés is érezhetően kisebb mértékű.

A manchesteri Egyetemen kifejlesztették az első aszinkron processzort, és az ARM7 nevet adták neki. A megoldásról (a processzor felépítéséről) nem tájékoztatták a szakmai köröket, de nyilvánvaló, hogy azaszinkron működésmód energiatakarékos üzemeltetést tesz lehetővé, hiszen az ARM7 tranzisztorai csak akkor lépnek működésben, ha hasznos jelet észlelnek a bemeneteiken.

A valós idejű feldolgozás igénye (a kép-, és hangfeldolgozás valamint a vezetéknélküli, mobil-kommunikációk jelfeldolgozása) az utóbbi tíz-tizenöt évben folyamatosan bővül. Ilyen igények ielégítésére szinte kizárólagosan DSP (Digital Signal Processor) processzorokat használnak, amelyeknek a legnagyobb és legismertebb gyártója a Texas Instruments.

Bővebb ismereteket a következő WEB-helyek nyújttják:

www.intel.com ,

www.amd.com ,

www.cyrilx.com ,

www.ibm.com

www.transmeta.com ,

www.arm.com i

www.ti.com .

5.1.4. Táruk – memóriák

A számítógépekben többfajta memória is megtalálható, különböző memóriákból kialakított memória-hierarchia. Ennek az egyszerű magyarázata az, hogy a memória sebességének fejlődése nem tudott lépést tartani a processzorok működési sebességével. Már a processzor regiszterei is tárnak tekinthetők, de ezt senki sem hangsúlyozza. A memória-hierarchia a regiszterekből, a külső és/vagy belső gyorsítótár(ak)ból (cache), az operatív tárból, és külső tárukából épül fel. Az utóbbiak közé tartoznak a néhány év óta használatos ún. flash memóriák, a diszketták, a merevlemezek, kompakt lemezek (CD-ROM-, DVD-ROM- és a legújabb FVD-ROM-lemezek) valamint a mágnesszalagok. Sokféle egyéb külső tár is létezik, de azokat, a flash memóriához

hasonlóan, személyi számítógéphez nem alkalmazzák. Nagyobb vállalatok adathalmazainak őrzésére ma is mágnesszalagokat használnak, amelyeket PC számítógépek is kezelni tudnak (a mainframe gépeknél a mágnesszalag-egységek elmaradhatatlanok, mert nincs jobb és megbízhatóbb külső tár a szalagnál – egyenlőre), de az otthoni használatban levő személyi számítógépek esetében ez a fajta külső tár nem terjedt el.

A regiszterek a memória-hierarchia csúcsát foglalják el. A processzorban vannak, sebességük megegyezik a processzoréval. Számuk és nagyságuk (bitekben kifejezve) kicsiny (kapacitásuk általában az adatcsatorna szélességével korlátozott). Adatokat (számítási eredmények és részeredmények) csak rövid ideig tárolnak, és a számítógép kikapcsolásával tartalmuk törlődik.

A belső és külső gyorsítótárak (cache) a hierarchia második szintjét képezik. Ezek lassabbak a regisztereknél, de a processzor sebességétől nem sokkal maradnak le. A gyorsítótárak nagysága sokszorosa a regiszterek kapacitásának, drágának számítanak (a hierarchia többi elemével összehasonlítva), feladatuk pedig a processzor és operatív tár sebességkülönbségének áthidalása. Az operatív tárnak azt a részét tartalmazzák, amelyre a processzornak legnagyobb valószínűség szerint szüksége lesz az elkövetkező pillanatokban. A processzorezért először a gyorsítótárhoz fordul, hiszen onnan várakozás nélkül megkaphatja a kért adatot (utasítást). Az eddigi megoldások legfeljebb háromszintű gyorsítótár-rendszert alkalmaznak. L1-el jelölik az első szintet, amelyet általában a processzorba építenek (belső cache), így a processzor frekvenciáján dolgozik. Az L1 cache az Intel processzorainál akár 32 Kbyte nagyságú is lehet, az AMD processzorainál pedig elérheti a 128 Kbyte-ot is. A második szintű gyorsítótárak jele L2, és az utóbi időben ezt is a processzorban alakítják ki. A legújabb processzoroknál az L2 nagysága az Intel termékekben 128 Kbyte, az AMD-nél 256 Kbyte. Újabb időben megjelent a harmadik szintű gyorsítótár is, L3 jelöléssel. Elhelyezése még nem teljesen kialakult, mert a Micron nevű cég a chipkészletbe (Mamba) rejtette, az IBM pedig a processzorba gyömöszölte (IBM Power4).

Az operatív tár a memória-hierarchia harmadik szintjét képezi, áramköreinek működési sebessége durván egy nagyságrenddel marad le (tízszer lassabb) a processzoréitól. Az operatív tár közvetlen elérésű RAM (Random Access Memory) memória-áramkörökből áll. Ez a memóriatípus egyszerű technológiával állítható elő, elég nagy kapacitású chip formájában kerül forgalomba, és, ami fontos kedvező árfekvésű. A RAM típusú memóriáknak is többfajta változata használatos. A legrégebbi fajtája a DRAM (Dinamyc Random Access Memory), amely FET tranzisztorokból és kondenzátorokból áll, kicsit lassú (18 MHz a működési frekvenciája) és a memória-bitek folytonos frissítésre van szükség (refresh), mert a kondenzátorok gyorsan veszítik a töltésüket. A RAM típusú memóriák következő fajtája az EDORAM (Extended Data Out

RAM), amely ma is használatos, hiszen a munkafrekvenciája elérheti az 50 MHz-et is. A legelterjedtebb fajta az SDRAM (Synchronous DRAM) memória. Ez bonyolultabb vezérlőegységet követel (a tárterület két részre, ún. bank-re osztása miatt), de 100 MHz-es frekvencián képes dolgozni (sőt, az utóbbi időben 133 MHz-en is). A Rambus nevű cég DRDRAM (Direct Rambus DRAM) vagy RDRAM (Rambus DRAM) memóriafajtát kínál, amely ugyan mesés sebességeken (800 MHz, a legújabb típusok pedig 1066 MHz !) végzik feladatukat, de még kicsit drágák. A legnagyobb memóriagyártó, a Micron, a Rambus kihívására válaszolva DDR SDRAM (Double Data Rate SDRAM) típusú memóriát dobott a piacra, amelyek alig drágábbak az egyszerű SDRAM (SDR SDRAM – Single Data Rate SDRAM) memóriáknál, de kétszer gyorsabbak. Az IBM kutatói olyan statikus RAM memória kidolgozásán fáradoznak, amelyeknél a memória-bitek helyigénye atomméretű lenne. Az angliai Keele Egyetemen szabadalmaztattak egy olyan fajta statikus RAM memóriát, amely 86 Gbyte/cm² –es kapacitást tud megvalósítani. Ez azt jelenti hogy egy bankkártya nagyságú felületen (85x55x15 mm) 10800 Gbyte (10Tbyte) mennyiségű-kapacitású adat számára van hely (és mindez a sorozatgyártás megkezdése után nem kerülne többé 50 US\$-nál). Újfajta VRAM (Video RAM) típusú RAM memóriát fejlesztettek ki a grafikus kártyák igényeinek megfelelően, amelynél egy bit a képernyő meghatározott pontjára vonatkozó adatot hordoz. Sok hasonlóság fedezhető fel a DRAM és a VRAM típusú memóriák működésében, kialakításuk (felépítésük) azonban különbözik: a VRAM két csatlakozóval, kapuval (Dual Port RAM) csatlakozik az adatsínre, az egyikre írás, a másikon olvasás bonyolítható egyidőben.

Az adatok mágneslemezre írása hosszú ideje csak jelsűrűség növelésére terjed ki. Ezt leginkább a lemez forgásának növelésével eszközölik ki, úgyhogy az eddig halgatólagosan elfogadott, szinte szbványának számító 3000-es percenkénti fordulatszámot 5400-ra, 7200-ra legújabban pedig 15000-ra (Seagate CheetahX15, ami 48 Mbyte/s-os átviteli adatsebességet eredményez) emelik. Újfajta író-olvasófejet szabadalmaztattak, amely GMR-nek keresztelt technológiát alkalmazva 128 Mbyte/s-os olvasási sebességet is elérhet, természetesen nagy forgási sebességű lemezcsomag esetén (www.readrite.com i www.guzik.com). Az anyagtechnológiai kutatások persze tovább folynak a nagyobb jelsűrűségek elérése céljából, és megfigyelhető, hogy az utóbbi időben 3-4 havonta újabb, nagyobb kapacitású merevlemez-modell jelenik meg a piacon. A Seagate-a (60 GB) és Maxtora (60 GB) termékei után megjelent az IBM Deskstar 75GXP öt lemezből kiépített merevlemez-egység 75 GB-os kapacitással(2000-es adatok). Három világhírű merevlemezgyártó, a Lucent, a Bell Laboratories és az Imation közös erővel vágott bele a holográfiai alapokon működő, cserélhető merevlemezek és –egységek fejlesztésébe, amelyeknek kapacitása várhatóan kb. 125 GB-os lesz (2002-re teszik a kereskedelmi forgalomba

bocsátásukat). Az IBM új adatrögzítési technológia kialakításán dolgozik, amely segítségével lehetővé válik a terabyte nagyságrendű adattömeg szokványos méretű lemezre történő írása. “Az eljárás háromdimenziós önformáló struktúrák alkalmazásán alapul, molekuláris szinten”²-jelentette ki az IBM képviselője.

A klasszikus kopaktlemez (CD lemez) 650 Mbyte, a DVD lemez 4,7 Gbyte (az Imation 5,2 Gbyte kapacitású DVD lemezeket is gyárt) kapacitású. A New York-i Constellation 3D nevű cég legutóbb beharangozott FMD – ROM jelű technológiája 140 Gbyte-os lemezek gyártását teszi lehetővé, A lemezen az adatokat 100 rétegben helyezik el, és egyidőben több réteg olvasása is végezhető. Becslések szerint a kopaktlemez-egység 100, egy lemez pedig csupán 2 US\$-ba fog kerülni (hogyan mikor, erről nem szól a hír).

A mágnesszalag ma is az egyik legmegbízhatóbb adathordozónak számít, és nagy adattömegek archiválására gyakran használják. Ha házi alkalmazásban nem is, de a nagyobb számítógép-konfigurációkban rendre megtalálható a mágnesszalag (illetve a mágnesszalag-egység). Az IBM és a Tandberg új rögzítési eljárások után kutat, és ennek eredménye a szabványos méretű mágneses kazetták kapacitásnövekedése: manapság már a 100 Gbyte-os kapacitásnál tartanak.

Nagyon sok hír és leírás található a Világhálón a tárolóegységekkel kapcsolatosan, ezek közül, mintegy kiindulópontként a következőket ajánljuk:

www.microm.com ,
www.rambus.com ,
www.cmruk.com/cmruKHD.html ,
www.kingston.com ,
www.hitachi.com ,
www.samsung.com ,
www.tandberg.com ,
www.seagate.com ,
www.lucent.com ,
www.maxtor.com ,
www.immation.com ,
www.ibm.com .

² Az idézet a PC Press 58., 2000 július-augusztusi kettős számából való.

5.1.5. Multimédia

A multimédia fogalma alatt a hang- és képfeldolgozással, valamint a grafikus alkalmazásokkal kapcsolatos egységeket, berendezéseket és szoftvereket értjük, amelyek segítségével a fejlesztésük és bemutatásuk (prezentációjuk) történik.

A számítógép képernyője kisszámú áramkörből épül fel, a könnyebb bővíthetőség érdekében. A képernyő vezérlőegysége külön lapra, az ún. grafikus kártyára kerül. A grafikus kártyán különböző vezérlők helyezkedtek el az idők (a fejlődés) folyamán, úgy mint az MDA (Monochrome Display Adapter, kétszínű, 4Kbyte saját tárhelykapacitása, 80x25 karakteres felbontású vezérlő, amelynél egy-egy karakter/jel 9x14 pontból/-pixelből épült fel), a CGA (Color Graphics Adapter 16 Kbyte saját tárhelykapacitással, és kétféle üzemmódú működéssel: szöveges üzemmódban a felbontása 80x25 karakter/jel – egy-egy jelet 8x8 pixelből alakítottak ki, grafikus üzemmódban kétfajta felbontást támogatott, a kétszínű 600x200 pixelűt és a négy színű 320x200 pontost, a Hercules (ugyancsak szöveges és grafikus működésű monokromatikus vezérlő, szöveges üzemmódban 80x25 jelfelbontással – egy jelet 9x14 pixel-mátrix ábrázolta, a grafikus üzemmódban pedig 720x348 pontra bontotta a képernyőt), az EGA (Enhanced Graphic Adapter 256 Kbyte saját memóriakapacitással, karakteres üzemmódban 80x43 karakteres felbontással – jegy jel 8x14 pontból épült, és 640x350 pont- és 16 színelbontással a grafikus üzemmódban), valamint a VGA és az SVGA.

A VGA (Video Graphics Array) az IBM fejlesztése, amit a PS/2 számítógépeire szánt. A szöveges (karakteres) üzemmód az EGA szabvány továbbfejlesztett változata, grafikus üzemmódja pedig 16 szín- és 640x480 pontfelbontással dolgozott. Minden VGA kártyának saját BIOS chipje volt (ez lapka fogadta a beállításokat, és bármilyen más vezérlőt tudott emulálni).

Az SVGA (Super VGA) ma is nagy népszerűségnek örvend a PC piacon. Az SVGA szabvány konkrét képernyőfelbontás-értékeket és színmélységeket (színárnyalatszámot) tartalmaz. Ezeknek megfelelően az SVGA kártya szükséges saját-tárhelykapacitása is kiszámítható (5.1.5.-1. Táblázat). A 16 bites ábrázolásmódot gyakran Hi-Color névvel, a 24 bites színmélységet pedig True-Color elnevezéssel illetik.

A grafikus kártya lelke, a gyorsító áramkör, mára a processzor kitüntetett címet érdemelte ki, és GPU (Graphical Processing Unit) rövidítéssel jelölik. Nemrég még mindenki a 3dfx cég gyártotta Voodoo család valamely tagjával ellátott grafikus kártyára esküdött (Voodoo1 – Voodoo5), a múlt (2000) év elejétől azonban a számítógép-sznobok csak az Nvidia GeForce család grafikus processzorai iránt érdeklődnek. Léteznek természetesen a GPU-nak más megoldásai (megvalósításai) is, pl. a Savage 2000 (S3 a gyártó cég) vagy a G800 (Matrox), de a minőség/ár arány egyértelműen az Nvidia elsőbbségét igazolja (sebességben és az alkalmazott

technológia fejlettségében is). A Világháló ebben a témakörben is bőséges adathalmazt rejteget a türelmes surfözőnek.

5.1.5.-1. Táblázat. A grafikus kártya tárhelykapacitása a felbontás és színárnyalat függvényében

Színmélység (bitekben)	4	8	16	24
Szín(árnyalat) száma	16	256	65536	$16,7 \cdot 10^6$
A képernyő felbontása				
640x480	256 KB	512 KB	1 MB	1 MB
800x600	256 KB	512 KB	1 MB	1,5 MB
1024x768	512 KB	1MB	2 MB	2,5 MB
1280x1024	1 MB	1,5 MB	3 MB	4 MB

GPU gyártók:

www.s3.com ,
www.nvidia.com ,
www.matrox.com ,
www.3dfx.com ,
www.ati.com ,

Grafikus kártya gyártói:

www.diamond.com ,
www.elsa.com ,
www.ct.com ,
www.asus.com ,
www.microstar.com .

A házi számítógépek első képernyői (monitorai) a televíziókészülékek voltak. Minőségük és a kétszeres jelátalakítás (a számítógép digitális jeleit először magasfrekvenciás jellé, majd újra visszaalakították a megjelenítés érdekében) okozta torzítások miatt azonban már a számítógép-alkalmazás elején elhagyták a TV készülékeket az architektúrából. A grafikus alkalmazások nagyfelbontású kártyákat és képernyőket igényelnek. A monokromatikus (egyszínű) képernyők ugyan már a múzeumban vannak, de a színeseknél is megmaradt a tőlük örökölt alapelv: katódcső képernyőjét soronként pásztázza végig az elektronsugár, amelyet egy módosított anód biztosít, és eltérítő elektródok vezérelnek. Színes képernyő (katódcső) esetén három elektronágyú (módosított anód), R, G és B jelöléssel állítja elő az elektronsugarat, amely a színes képernyő pontjait "letapogatja". Egy-egy pont a képernyőn három kisebb pontból (piciny háromszög sarkaiban elhelyezve – delta felépítésű katódcső, vagy három alapszín- és egy sötét csígyvonalban – in-line felépítésű katódcső) épül fel. A kép kirajzolása megvalósulhat

félképenként (először a páratlan sorokat rajzolja ki az elektronsugár – első félkép, majd a párosakat – második félkép), és akkor az interlaced minősítéssel illetik a képernyőt. Az újabb monitorok esetében a kép egy végigtapogatással valósul meg (non-interlaced monitor), a másodpercenkénti képszám pedig 70 körül alakul. A mai képernyők in-line, non-interlaced és multisync (változtatható a képrajzolás sebessége vagy frekvenciája) szabványú síkképernyők. Két széleskörűen elfogadott szabvány is létezik a képernyőkre, az MPRII és a DMPS. Az MPRII Svédországból ered, és az elektromágneses hullámok sugárzási energiájára vonatkozik a következő két frekvenciasávban: 5 Hz-től 2 kHz-ig, és 2 kHz-től 400 kHz-ig. A DMPS a monitorok négy működésmódját határozza meg behatárolva a villamos energiafogyasztás legmagasabb szintjét az egyes üzemmódokban.

A folyadékkristályos képernyők (LCD – Liquid Crystal Display) elsősorban a hordozható számítógépekben találtak alkalmazásra, de persze előfordulnak az asztali és egyéb számítógépek konfigurációiban is. A folyadékkristályos képernyők két alapvető típusát különböztetjük meg: a DSTN és a TFT típusokat. A DSTN (Dual-Scan Twisted Nematic) képernyő képrajzolása lassú (kb. 300 ms szükséges a kép teljes megjelenítésére, másszóval másodpercenként csupán 3 kép váltható rajta!) és az olcsóbb hordozható gépek (laptop, notebook, palmtop) A DSTN képernyő esetében a vezérlősinék végzik a folyadékkristályok átrendeződését, ezért lassú a kép kirajzolása. A TFT (Thin Film Transistor) képernyőknél FET (Field Effect Transistor) tranzisztorok szolgálnak a kristálmolekulák átrendezésére. A vezérlősinék gyorsan nyitják-csukják a FET-eket, így a képmegjelenítés is sokkal gyorsabb (a képrajzolásra kb. 25 ms kell, vagyis a másodpercenkénti képszám eléri a 40-et!).

Összevetve a klasszikus és LCD képernyők tulajdonságait, a következő megállapítása juthatunk: az LCD monitorok előnyösebbek ha a súlyukat, a képernyőméretet (a szokványos képernyők valódi mérete 1-1,5 inch-csel kisebb, mint a katódcső mérete), az energiafogyasztást, a fizikai méretet, a képfrissítést (az LCD képernyőknél a képernyőfrissítésre nincs szükség) és a sugárzást (a folyadékkristályosok nem sugároznak) hasonlítjuk össze. Egyforma minőségű a két képernyőtípus, ha a felbontást és a színárnyalatszámot szemléljük. A klasszikus képernyők előnye az árukban, a képélességben és a színerősségben domborodik ki.

A legnagyobb monitor (képernyő) gyártók kínálatai megtekinthetők a Világhálón a következő címeken:

www.acer.com ,

www.belinea.com ,

www.hansol-us.com ,

www.hitachi.com ,

www.nec.com ,
www.panasonic.com ,
www.sony-cp.com i
www.viewsonic.com .

A hangkártyákat talán leginkább a számítógépes játékok hívták életre. Mivel a számítógépen minden adat (információ) bináris alakban van (diszkrét értékekkel rendelkezik), így a hanganyag is, a hang felvételezésénél a folytonos (kontinuális) jelet át kell alakítani diszkrét (bináris) értékke, lejátszásnál pedig a leírt folyamat fordítottja játszódik le. A legismertebb a Creative Technology cég Sound Blaster hangkártya-sorozata: a Sound Blaster, a Sound Blaster Pro, a Sound Blaster 16, a Sound Blaster AWE 32, a Sound Blaster AWE 64, a Sound Blaster AudioPCI 64V, stb.). A régebbi típusok ISA gyűjtősínre csatlakoztak, a maiak már a PCI szabványt támogatják (és erről a nevük is árulkodik). Nagyon sok hangkártya-gyártó kínál különböző minőségű és árú terméket. A Terratec például a kimagasló tulajdonságokkal rendelkező EWS 88 MT típusú kártyái mellett közepes minőségű, EWS 64 XL típusú kártyákat is kínál. A számítógép hangfeldolgozási alrendszere részben az alaplapra integrált egységekből épül fel (a vezérlőegység mindenképpen ott helyezkedik el), ahogy ezt az Audio Codec csatlakozóra vonatkozó 1997-ben létrejött szabvány is megfogalmazza (AC'97). Az AMR (Audio/Modem Riser) jelölésű szabvány az AC'97-es előírás (ajánlás) kibővített változata: az AMR kártyát írja le (amelyen a modem, és annak csatlakozója is elhelyezkedik), valamint a kártya csatlakozását az ISA és PCI sínrendszerhez.

A mai, valósídejű kép- és hangfeldolgozás elengedhetetlen kellékei (de méginkább főszereplői) a digitális jelfeldolgozó processzorok, a DSP-k (DSP – Digital Signal Processor). A korlátozott számú utasításkészlet jelfeldolgozásra optimalizált, így a DSP-k átvállalják a kép- és hangfeldolgozást a PC mikroprocesszorától, tehermentesítve azt.

A hangkártyákról szóló ismertetések a következő Internet-címeken találhatók:

www.creativelabs.com ,
www.diamondmm.com ,
www.digitalaudio.com ,
www.dtk.com.tw ,
www.gravis.com i
www.terratec.de .

5.1.6. Bemeneti/kimeneti (I/O) egységek

Már vagy húszéves véleményt tesztek közzé, amikor egy amerikai tudós véleményét (a nevét már sajnos elfelejtettem) közlöm, miszerint Amerikában már (akkor, húsz évvel ezelőtt) minden elképzelhető berendezést, tárgyat összekötöttek a számítógéppel (talán még az ekéről sem feldkezték meg – fűzöm tovább a gondolatot). Mindez persze csak tréfa, de minden tréfában valami igazság is lapul. Azóta a helyzet csak “fokozódott”, úgyhogy az eddig bemeneti-kimeneti egységként használt egységek, berendezések és tárgyak felsorolására nem is vállalkoznánk. Itt bennünket csak a szabványos, megszokott I/O egységek érdekelnek.

A bemeneti egységek adatokat, méréseredményeket, képadatokat, hanganyagokat, stb. gyűjtenek a számítógép közeli vagy távoli környezetéből. A bemeneti egységek közé a következőket fogjuk sorolni: billentyűzet, egér, hanyattegér (trackball), digitalizálásra szolgáló tábla, lapszkener, mikrofon és kamera (a legszokványosabbakra szorítkozva).

A kimeneti egységeknek köszönhetően tud a számítógép hatni a környezetére. Ezek közé az egységek közé sorolható a képernyő (monitor), a nyomtató, a rajzgép (plotter) és a hangszórók (fülhallgatók).

A bemeneti-kimeneti egységek fogadni is tudnak adatokat, de megjelentetni, küldeni is. Ide tartoznak az érintésre érzékeny képernyők (touch screen), a modemek és hálózati kártyák.

A billentyűzet a konfiguráció intelligens egysége, mert mikroprocesszor felügyeli a munkáját, az adatokat pedig a soros szabvány előírásai szerint továbbítja a számítógép felé. Az XT és az AT billentyűzetek csak a jelek kódolásában különböznek, és bár az XT gépe a dinoszauruszok sorsára jutottak, a billentyűzetek egy része ma is támogatja mind az XT, mind az AT kódolási szabványt.

Az egér a számítógép nélkülözhetetlen részévé vált a grafikus felületű operációs rendszerek és alkalmazások miatt. A számítógép közelében élő legelterjedtebb egérfaj a golyós, optomechanikus egér, mert a legegyszerűbb felépítésű és legolcsóbb. Az optoelektromos egerek már megjelentek a piacon, és az elterjedésük is várható, hiszen megbízhatóbbak és kevésbé romlanak (nincs mozgó alkatrészük) mint az előbb említettek, de az áruk még egyenlőre magas. Vannak még a piezoelektromos jelenségre és ultrahangra épülő egér-megvalósítások is, de egyenlőre (elsősorban az árfekvésük miatt) nem keresettek a piacon. Az egerek RS232 soros csatlakozón is kommunikálhatnak a számítógéppel, de manapság szinte kizárólag a PS/2 szabványú egereket forgalmazzák (így egy soros csatlakozó/kapu felszabadul). A legtöbb egér két (bal és jobb) gombbal rendelkezik, de nem ritka már a háromgombos egér sem (az jobb alkalmazások és szoftvertermékek támogatják a harmadik gomb használatát (különböző műveletek elvégzését bízzák rá)).

A számítógép használatok letiltása az idegenek részére olyan problémakör, amely több ötletet is eredményezett. Ezek közé tartozik az egerbe vagy billentyűzetbe épített ujjlenyomat-olvasó. Ez nem más, mint egy parányi szkennel, amely a felhasználó ujjának képét, vagy ujjlenyomat-képének csak bizonyos számú jellegzetes pontját (40-től 60 jellegzetes pont alapján az ujjlenyomat visszaállítható-rekonstruálható) elküldi a számítógépnek az USB, vagy a párhuzamos csatlakozón keresztül beazonosítás céljából. Ezen az alapelven működik a Cherry G81-12000 jelzésű billentyűzet, illetve a Siemens és a Logitech egyes egérmodelljei.

A hanyattgér (trackball) tulajdonképpen egy hátára fektetett egér az összes egerlehetőségekkel, és az egeret helyszűke esetén felválthatja. Nála ugyanis nem az egység mozog, hanem csak a golyó. Az optomechanikus elven működő egységnek általában két gombja van.

A digitalizáló tábla a CAD programoknál jelent nagy segítséget. Ez egy különleges, műanyag tábla induktív érzékelőkkel a hátlapján. Az érzékelők mátrix-szervezésűek, az egységnyi felületre eső számuk (magátólértetődően) a tábla érzékenységet, felbontását határozza meg. Különböző méretben gyártják őket, és a CAD programokkal történő fejlesztések legfontosabb segédeszközeinek számítanak.

A lapolvasó a forrásdokumentumokról készít digitális "képmásolatot". Lényegében digitalizálja a dokumentum képét. A lapolvasó érzékelői a CCD (Charge Coupled Device) jelzésű fényérzékeny elemek, amelyek folyamatos (feszültség)jelkimenettel rendelkeznek. A CD által generált folyamatos jel ADC (Analog-Digital Converter) átalakítók segítségével binárisá alakul, amit aztán a számítógép már kezelni tud. A lapolvasók lehetnek kézi vagy asztali kivitelezésűek, a képmásolatot pedig egy- vagy többmenetes működésmódban valósíthatják meg.

A képkészítés négyféle eljárással készülhet a következők közül:

- a) Line art,
- b) Halftoning,
- c) Grayscale és
- d) True Colour.

A mai lapolvasók egy letapogatással (egy menetben) Grayscale vagy True Colour eljárással készítik a másolatokat, függetlenül attól, hogy az végtermék, a kép színes-e vagy fekete-fehér. A lapolvasó optikai felbontásának értéke a kapott képminőséget befolyásolja. A mai lapolvasók 600x1200 dpi (Dot per Inch) vagy ppi (Pixel per Inch) felbontással dolgoznak. Az ettől finomabb (nagyobb felbontású) képek általában matematikai eljárással támogatott interpoláció segítségével készülnek. Az interpoláció megvalósítása lehet szoftveres vagy hardveres. A képfelbontás (m)értéke nagyban befolyásolja (meghatározza) a lapolvasó árát. A szkennerek többfajta fájlformátumot támogatnak, amelyek közül a leggyakoribbak: a BMP

(Bitmap), a TIFF (Tagged Image File Format), a PCX (PC Paintbrush) és a JPEG (JPG). A JPEG adattár-formátum igényli a legkisebb tárfelületet a kép minőségének megőrzése mellett (True-Colour színmélységgel, vagyis 24 bittel dolgozik). A lapolvasók gyártói termékeik (szkenerek) mellé gyakran valamilyen jelfelismerő, OCR (Optical Character Recognition) szoftvert is csatolnak. Szövegekről készült képek alapján a karakterfelismerő OCR program szöveges adattárat hoz létre, amely sokkal kisebb tárigényű a lapolvasó végtermékétől, és emellett még módosítható is.

A mikrofonok, de főleg a kamerák (különösen a WEB – kamerák) a Világháló elterjedésével, és a jó minőségű hang- és képfeldolgozó szoftverek megjelenésével váltak keresettekké. Főleg otthoni szórakozásigényt elégítenek ki, de nagyon sok más lehetőség is rejlik bennük: video – konferenciák, távoktatás (tele-teaching) szervezése, képtelefon, stb.

Mivel már volt szó a monitorokról, itt csak megjegyeznénk, hogy a képernyők lehetnek kimeneti, de kimeneti-bemeneti egységek is. Az utóbbi szerepükkel ritkábban találkozhatunk, hiszen azt csak az érintésre, esetleg megvilágításra (kéz, illetve bármilyen tárgy, vagy sajtósárga eszköz, pl. fényceruza, stb.) reagálni tudó képernyőkkel lehet megvalósítani (ezek pedig igencsak drágák). Nagy erővel kutatnak viszont a szupervékony, hajlítható képernyők technológiájának kidolgozásán.

A feldolgozások eredményei még mai is leggyakrabban papíron alakulnak jelentésekké. A papíron való megjelenítés a nyomtatók feladata, amelyek nagyon sokféle elven és kivitelben készülnek valamint különböző minőségű szolgáltatást nyújtanak.

A karakternyomtató papírszélességnyi hengerén minden nyomtatható jelből található egy-egy sor a henger palástján. A henger addig nem fordul el, amíg az össz azonos jel meg nem jelenik az adott sorban, mert a kívánt jel előfordulási helyén a kalapács ráüti a papírt a hengerre (a nyomtatószalag – belt természetesen a henger és a papír között helyezkedik el), majd a kalapács a hengeren levő jel következő megjelenítendő helyére mozdul (pozicionál). Ha az össz szükséges jel az adott sorban megjelent, a henger elfordul, és egy másik jel kerül szembe a kalapáccsal. A kívánt helyeken (pozíciókon) “leüt” a kalapács (a leütések közben persze pozicionál), és így a sorban szükséges valamennyi jel megjelenik a papíron abból a fajtából, amelyik “kalapácsvonalba” került. Ha az összes jel az adott sorban le van nyomtatva, akkor sordobás következik (line feed), amikor a lap egy sorral feljebb gördül. Ezután az egész folyamat ismétlődik. Ez egy lassú, zajos nyomtatófajta, amelyik az említett hátrányai miatt kevésbé közkedvelt.

A sornyomtató egy továbbfejlesztett karakternyomtatóként fogható fel, mivel a henger helyett a nyomtatási betűszélességnek megfelelő számú, egy betűszélességű diszkek találhatók.

A diszkek palástján valamennyi nyomtatható betű megtalálható. A diszkek egyenként (külön-külön) elforgathatóak, így megfelelő elektromechanikus vezérléssel és kellő pontosságú időzítéssel a kalapácsok egyidejűleg ütnek le egy teljes sort. A kalapácsok itt is a papírt a nyomtatószalaggal együtt nyomják a diszkek palástján kiválasztott és beállított betűhöz. Aránylag lassú, de mindenképpen zajos megoldásról van szó. A kiszolgálószemélyzet csak zajsűrű fülvédők használatával maradhat az ilyen nyomtatók közelében. Emellett a nyomtató hátrányaként róható fel a szinkronizációt biztosító bonyolult elektromechanikus berendezése.

A tűs nyomtatók a nyomtatandó jeleket tűszúrások (tüleütések) segítségével alakítják ki. A tűk száma alapján kilenc-tűs és huszonnégtűs nyomtatókat különböztetünk meg. A festett és végtelenített nyomtatószalag (ribbon), amelynek segítségével a jellenyomatok a papíron láthatóvá válnak, kis, műanyagdobozkában helyezkedik el. Sokáig kedvelt nyomtatófajtának számított a tűs nyomtató, főleg kedvező árfekvése, de tartóssága és karbantartást nemigen igénylő működése miatt is. A nyomtatás minőségében és a sebességben nem számít bajnoknak.

A tintasugaras (Ink – Jet) nyomtatók kétféle alapelven működnek: az egyik megoldásnál a tinta szüntelenül kering a rendszerben (ami folyamatos energiafogyasztást jelent), a másiknál a tinta csa szükség esetén “mozdul meg”, illetve kezd mozgásba a tintavezető csatornában a fűvókák felé. Az első, “keringtetett tintás” megoldás nem nyert létjogosultságot a piacon. A második alapelvet kétféle technológiával oldják meg: buborékok, illetve piezoelektromos hatás segítségével. A buborékos tintasugaras nyomtatók a tintabuborékokat a nyomtatófejben elhelyezett fűtőtest segítségével állítják elő. A tinta 300 °C hőmérsékletre való melegítésekor buborékok (és ezzel párhuzamosan túlnyomás) jelentkeznek a nyomtatófejben. A túlnyomás miatt a buborékok tinta kíséretében távoznak a fejből (kifröccsennek): így keletkezik a kívánt jel a fejhez közel elhelyezkedő papíron. A piezoelektromos megoldásnál a tinta eltávolítását a fejből aprócska piezo-keramikus lapocskák végzik, amelyek áramimpulzus hatására meggörbülnek, így teremtve meg a túlnyomást. A tintasugaras nyomtatók jó minőségű színes lenyomatot biztosítanak, csendes és gyors működésűek, de egyenlőre még drágák: no nem a nyomtatókészülékek maguk, hanem a tinta-utántöltések.

Manapság a PC számítógépek leggyakoribb nyomtatófajtája a lézernyomtató. A lézernyomtatók általában A4-es méretűek, és az oldalt folyamatosan nyomtatják le. A nyomtató hengerét többnyire szelénréteggel vonják be, amelyet kisintenzitású lézersugárral elektromossá tesznek azokon a helyeken, ahol festéklenyomat esedékes. Az elektromossá tett helyek ugyanis magukhoz vonzzák a festékszemcséket a festékdobozból, majd a hozzájukcsimuló papírra nyomják. A festéket ezután rögzítik a papíron: 200 °C hőmérsékleten ráégetik a papírra. A lézernyomtatók ismérve a felbontási finomság, melynek értéke 300-tól 600 dpi(dot per inch)-ig

terjed, a nyomtatási sebesség pedig 4-től 20 lap percenként. Vannak természetesen színes lézernyomtatók is, de ezek egyenlőre még drágák.

A termikus nyomtatók az ár szempontjából a végeket képviselik, ugyanis az alkalmazott nyomtatási technológiától függően vagy a legolcsóbb, vagy a legdrágább nyomtatók közé sorolhatók. Az alacsony árfekvésű termikus nyomtatók hőelemekkel ellátott nyomtatófeje a nyomtatáshoz hőérzékeny papír alkalmazását igényli. A lenyomat létrejöttét a nyomtatófej felfűtött hőelemének papírral való érintkezése váltja ki. A hőérzékeny papír azonban az őt érő fenyhatás következtében idővel veszít az élességén, a kontraszton, egyszóval elsötétedik, ami a kinyomtatott jelek eltűnéséhez, elvesztéséhez vezet. Az árlétra felső régióiba tartozó termikus nyomtatók nem igénylik a drága és fényérzékeny (természetesen mindenképpen hőérzékeny) papírt, mert a jeleket egy, a tintának és egy fajta speciális viasznak a sajátos keverékével ellátott szalag jeleníti meg. A szalagon lévő keverék a megfelelő jelet tartalmazó felmelegített hőelem hatására felolvad, és a közelében elhelyezkedő papírra ragad. Az olcsóbb nyomtatók mozgó nyomtatófejjel, a drágábbak több ezer hőelemből (termisztor) álló mozdulatlan nyomtatófej segítségével valósítják meg a nyomtatást. Ez utóbbiakat a színes nyomtatásban is (leginkább a négy színű nyomásoknál) használják.

A szublimációs nyomtatók, annak ellenére, hogy külön nyomtatófajtaként emlegetik őket, tulajdonképpen termikus nyomtatók. Már a nevükből kikövetkeztethető a működési elvük: a lenyomat a szilárd halmazállapotban levő festékanyag hirtelen és gyors melegítésével jön létre mégpedig oly módon, hogy a szilárd festékből közvetlenül gázhalmazállapotú anyag keletkezik. A sajátos eljárás mellett különleges, ún. polimerizált papírt igényelnek a szublimációs nyomtatók, ami nagyban megdrágítja a nyomtatást, mindemellett a nyomtatási folyamat szinte elkésztően hosszú (jó munkához idő kell...). El kell ismerni, hogy a nyomtatás minősége kiváló: mint a fénykép (tudomásom szerint főként képek nyomtatására használják). Egy A4-es oldal True Color minőségű (24 bites szikódolás, ami 16,7 millió színárnyalatot jelent) nyomtatása röpké tíz perc alatt készül el!

A rajzgépek, vagy közkeletű idegen szóval plotterek a CAD programok leggyakoribb segítő társai. A rajzlap nagysága szabványosított, standardizált: A4, A3, A2, A1 és A0. A rajzlap nagysága alapján végzik a plotterek egyik fajta csoportosítását. A rajzon megjelenő (a rajzgépen a valóságban is elhelyezkedő) különböző vastagságú tollak nyomait a rajzgép irányítását végző szoftver optimalizálja a gyorsabb kirajzolás érdekében. Az A4-es rajzlap-nagyságnál a rajzgépek szerepét teljességgel átvették a lézernyomtatók.

A hangszórók és fülhallgatók manapság különösen az ifjúság körében közkedveltek, elsősorban a multimédiás betétekkel tarkított játékok és alkalmazások miatt, de ne feledkezzünk

meg a zenei CD-kről, a házi összeállítású tömörített (MPEG3) zenei CD-kről, a DVD-kről és a házi változatokról, a DivX-ekről.

A nyilvános Internet szolgáltatót a felhasználók többsége (dial up) telefonvonalon éri el. A számítógép által létrehozott bináris (digitális) jelek a telefonvonalakon való "utazás" közben a felismerhetetlenségig eltorzulnak. Ezt megelőzendő, a számítógép bináris jeleit egy modemen keresztül kapcsolják csak a telefonvonalra, amely berendezés a bináris jeleket analóg jelekké alakítja, MODulálja, ellenállóbbá teszi a telefonvonalakon jelentkező zajjal szemben. A vételi oldalon természetesen az analóg jeleket vissza kell alakítani binárisra (hogyan tudja a számítógép felismerni), DEModulálni kell. A küldő- és a fogadószerep pillanatról pillanatra változik ugyanarra a számítógépre vonatkozóan is, így mindkét feladatot ugyanaz a berendezés végzi el (innen a neve is: MODEM). A bináris jel védelmét leggyakrabban háromféle módon végzik, így megkülönböztetünk amplitúdó, frekvencia- és fázismodulációt. A modemek hivatalos jele (rövidítése) DCE (Data Circuit – Terminating Equipment). A belső modemek valójában modemkártyák, amelyek ma PCI (régebben ISA) típusú csatornára (bus) kapcsolódnak az erre a célra létrehozott csatlakozóhelyeken (slot). Mindez persze a számítógépházon belül helyezkedik el. A külső modemek saját házzal, dobozzal rendelkeznek és a gépház soros csatlakozóhelyéhez (serial port) kapcsolódnak. Ma az 56 Kbit/s-os sebességű modemek a legelterjedtebbek.

A hálózati kártyák iránti igény régóta jelen van az egymástól kis távolságra telepített számítógépek esetében, hiszen az összekapcsolt gépek összes (gyakran költséges) hardver- és szoftver-erőforrása mindenki számára (még ha szelektáltan és korlátozottan is, de) elérhető: a közös adatbázis, a drága hardver-eszköz (mondjuk az egyetlen, közös, színes, lézernyomtató). A hálózat-topográfia fogalma a számítógépek fizikai elrendezését jelenti, a hálózat-topológia pedig számítógépek összekapcsolásának módját foglalja magába. Bennünket inkább a topológia kérdésköre izgat, hiszen a topográfia "csak" a számítógépek közötti távolságok pontos meghatározása szempontjából érdekes számunkra. A topológia megvalósításában kétfajta irányzat él: a) pont-pont kapcsolat (point to point topology), és a többpontú kapcsolat (multipoint topology), amikor az üzenet minden számítógéphez »egyidőben« jut el. Az első típusú topológia-alapelem segítségével kiépített hálózatok a következők: csillag, gyűrű, esetleg fastruktúra. Ennél a hálózatonál az üzenet érkezésekor a számítógép ellenőrzi, vajon ő-e a címzett, és ha igen, az üzenetet feldolgozza, ha nem, továbbküldi a másik számítógépnek amelyikkel még kapcsolata van (és nem amelyiktől kapta). A multipoint topológiájú hálózatoknál a számítógépek valamennyien a közös üzenetcsatornára kapcsolódnak. Az üzenetcsatorna lehet nyitott vagy zárt, és ez utóbbira mondják, hogy gyűrű alakú, az előbbire pedig, hogy gyűjtősínes multipoint topológiájú hálózat. Itt az elküldött üzenetet minden számítógép egyszerre kapja meg, egyszerre

ellenőrzik a címzettet, és csak a címben megjelölt számítógép (csoportos címzés – multicasting – esetén a csoport összes számítógépe) dolgozza fel az üzenetet. A számítógépek összekapcsolása a topológiától függetlenül három fajta vezetéktípussal történik:

- a) sodort vezetékkel (árnyékolásmentes – UTP – Unshielded Twisted Pair, vagy árnyékolt – STP – Shielded Twisted Pair kivitelben),
- b) koaxiális kábelrel, és
- c) optikai kábelrel.

A hálózati kártyák a megvalósított fizikai hálózatra való csatlakozást teszik lehetővé. A helyi jellegű hálózatoknál – LAN (Local Area Network) UTP típusú vezetékkel vagy koaxiális kábelrel találkozunk. Az újabb hálózati kártyáknak kétfajta csatlakozó-típusuk is van: az RJ-45 jelzésű az UTP vezetékek számára és a BNC (Bayonet Neil-Councilman) jelölésű, a vékony koaxiális kábelek csatlakozására. A számítógépek közötti nagyobb távolságok áthidalására vastag koaxiális kábelek szolgálnak, mert sokkal kisebb a jelgyengülésük, a jelgyengítésük. Ezeket azonban a szokványos hálózat kártyák nem tudják befogadni, ezért a számítógépek rákapcsolásához illesztők használata szükséges. A hálózati kártyák csupán a fizikai kapcsolatot biztosítják a gépek között, a számítógép-hálózat működéséhez elengedhetetlen a megfelelő hálózati (kommunikációs) szoftver jelenléte és működése.

5.1.7. Kommunikáció

Az üzenetküldést, adatcserét megvalósító folyamat, amelyet kommunikációnak hívunk, a hardver- és szoftvereszközök együttműködésének szép példája. A mai modern kommunikáció szélessávú és gyors adatátvitelt igényel. Az átvitelre szánt adatmennyiség folyamatosan növekszik az utóbbi évek folyamán, és mivel az átviteli sebesség véges (durván a fény terjedési sebessége a vákuumban), az átvitel növelését az átviteli sáv szélesség bővítésével oldják meg. A nagy átviteli sebességű hálózat vagy vezetéken vagy elektromágneses hullámok segítségével valósítja meg a kommunikációt az igénylők számára. A vezetékes átvitelre a már meglévő, kiépített hálózatok a jelöltek. Mivel a villamos energia minden lakásban jelen van, talán az áramelosztó hálózata a legtermészetesebb alany a kommunikációs hálózat megvalósítására. Mindjárt a második “leglogikusabb” közeg a telefonhálózat, amely ugyancsak sok helyütt működik. A harmadik lehetőség a kábelTV hálózat. A drótnélküli (elektromágneses hullámokkal történő) adatátvitel közege természetesen a levegő.

Az elektromos vezetékeken keresztüli kommunikáció (PLC – PowerLine Communication) kell, hogy elviselje a legnagyobb zajszintet az összes vezetékes átvitelek közül, de a legutóbbi Németországi eredmények igen biztatóak. A legújabb, energetikai hálózatra kidolgozott illesztők segítségével 2 Mbit/s-os átviteli sebességet érnek el, ami kényelmes Internet-böngészést tesz lehetővé. A PLC-illesztők továbbfejlesztésével esetleg a fali aljzatokban is elhelyezhetőek majd (a mai méret a számítógép tápegységében való elhelyezést sugallja).

A telefonhálózatot az emberi hang átvitelére tervezték, de ma egyre gyakrabban használják adatátvitelre is. Az ISDN (Integrated Services Digital Network) adatátviteli szabvány a telefonhálózaton megvalósított kommunikáció előírásait tartalmazza, és 1984-ben, megalkotásakor két, 64 Kbit/s-os átviteli sebességű csatorna megvalósítását rögzítette. Ha a telefonvezeték (egy érpár) csak digitális adatátvitelre szolgál (tehát telefonálni nem lehet rajta), akkor digitális előfizetői vonalról beszélünk, és DSL (Digital Subscriber Line) a jele. A világon többfajta DSL változat működik, a legelterjedtebb az ADSL (a rövidítésben az A betű az asszimmetrikus szóból ered), amely 9 Mbit/s sebességgel küldi és 1 Mbit/s sebességgel fogadja az adatokat a telefonközpont 5 km-es körzetében. A leggyorsabb változat a VDSL (Very high-rate DSL) amelyiknél 25 Mbit/s küldési és 3 Mbit/s fogadási sebességet jegyeznek, ha a berendezés nincs távolabb 300 m-nél a telefonközponttól. Nálunk a DSL még ritkaság, de várhatóan (a gazdaság fejlődési ütemének függvényében) gyorsan el fog terjedni.

A kábelTV koaxiális kábelhálózata szimpatikus közeg, jobb minőségű és megbízhatóbb a telefonhálózattól. A most működő kábelTV-hálózat komoly hátránya, hogy a jel csak egy irányban utazik a hálózaton, így az erősítők is “egyirányúak”, és a felhasználók ugyanarra a vezetékre kapcsolódik. Az Internet szolgáltatásainak lehetővé tételéhez az egyirányú hálózatot csillagtopológiájúvá kell alakítani úgy, hogy minden felhasználó közvetlenül kapcsolódjon a szolgáltatói központhoz. A koaxiális kábelek többszörözése helyett optikai kábel-átvitelt kellene biztosítani a hálózat megfelelő pontjaiban, ami lényeges befektetést igényel. Az ilyen megoldásnál nem biztosítható rögzített átviteli adatsáv-szélesség, hiszen az minden pillanatban az aktív felhasználók számától függ. Becslések szerint a küldési adatátviteli sebesség 0,5 – 2,5 Mbit/s, a fogadási sebesség pedig 1,5 – 3 Mbit/s értékek között változhatna.

A szatelit-kommunikáció a következő a sorrendben, ami az átviteli sebességet illeti. Az átviteli jel frekvenciája gigahertz nagyságrendű ($10^9 - 10^{10}$ Hz). A felhasználók többsége számára a szatelit-kommunikáció még csak (adat)vételre, vagyis adatfogadásra jó, mert a leadó ára túlságosan magas. 2000-ben indult egy vállalkozás (Aerospace Corporation) a drága kommunikációs szatelitek kis, olcsó szatelitekkel való helyettesítésére. Az említett cég útjára bocsátotta az első »picasat« típusú, 20 dkg tömegű, 10x8x2,5 cm méretű szatelitet. Bár az első

„picasat” beváltotta a hozzá fűzött reményeket, a szakértők mégis úgy vélik, hogy az eljövendő kommunikációk jövőjét az ettől kicsit nagyobb, ún. „nanosat” típusú szatelitek fogják biztosítani.

Az optikai kábelek működési frekvenciája százezerszerese a szatelit-kommunikáció munkafrekvenciájának, vagyis $10^{14} - 10^{15}$ Hz. Mivel a magasabb frekvencia nagyobb adatmennyiség továbbítását teszi lehetővé egy időegység alatt, a leggyorsabb adatátvitel ebben a közegben zajlanak. A gondok a villamos (elektromos) jelek optikai jelekké való átalakításakor jelentkeznek: egyrészt nehézkes, másrészt drága ez a fajta átalakítás, hiszen az átalakító áramkörök sebességével szemben nagyok az elvárások. Éppen ezért olyan jelentős a Lucent Lambda Wavestar elnevezésű jelátalakító, amely kb. 25 %-kal olcsóbb a hozzá hasonló berendezésekhez. Az elmúlt, XX. század utolsó évében (2000-ben) egy új, opto-chip-nek elnevezett jelátalakítót szabadalmaztatott két amerikai (a Washingtoni és Dél_Kaliforniai Egyetem) egyetem, amelynek a jelátalakítási sebessége 100 Gbyte/s. A leggyorsabb adatátvitelt a Lucent Bell cég valósította meg, egy kísérleti, 300 km hosszú optikai vezetéken, az elért sebesség 3,28 Tbit/s (terrabit másodpercenként) volt. Ezt az átviteli sebességet frekvenciamegosztással sikerült csak elérniük, természetesen, a 82 különböző frekvencián az adatátviteli sebességek értéke 40 Gbit/s volt.

A számítógépek közötti kommunikáció megvalósítása nagyon bonyolult feladat. Az ISO (International Standard Organization) Szabványügyi Hivatal külön előírásokat (szabványt) határozott meg (a szabvány rövidítése OSI – teljes elnevezése pedig Open System Interconnect) a számítógépek közötti kommunikáció kialakítására. Az ajánlások (a szabvány) a kommunikációt rétegekre (layer) osztva írja le, az OSI által javasolt kommunikációs architektúra szerint. Az elképzelés szerint a kommunikáció lebonyolításakor a két számítógép azonos rétegei valósítják meg egymásközt az adatátvitelt. A kommunikációs rétegek (vagy szintek) az OSI architektúra szerint a következők:

1. Fizikai,
2. Adatkapcsolati,
3. Hálózati,
4. Átviteli,
5. Kommunikációs (viszony),
6. Megjelen(ít)ési és
7. Alkalmazói réteg.

A fizikai réteg felelős az adatokat alkotó bitek egyenkénti átviteléért (az átvittel kapcsolatos jelzések – szignalizáció, a bitek modulációjának és az átviteli folyamat

összehangolásának – szinkronizáció – végrehajtásáért). Itt kapnak helyet az ajánlások a konkrét átviteli közegfajtára tervezett illesztőáramkörök mechanikai- és villamos tulajdonságaival kapcsolatban. Az adatkapcsolati réteg funkcionális és eljárásszintű előírásokat (elvárásokat) tartalmaz a hálózatban összekötött két számítógép közötti kommunikáció felvételére (kiépítésére, elindítására), a meglévő kapcsolat fenntartására (ápolására) és megszüntetésére. A hálózati réteg az OSI elvárások szerint a számítógép hálózatra való kapcsolódását, hibaellenőrzést, adatáramlás-ellenőrzést és a célgéphez vezető fizikai útkijelölést végzi. Az átviteli réteg a hosszú üzenetek feldarabolását (szegmentálását), időbeni ütemezését (multiplexelés) és összeszerkesztését (összerakását) végzi az OSI előírások szerint, valamint biztosítja az alacsonyabb (hálózati) réteg számára a célgép hálózati címét. A kommunikációs réteg a gépek közötti párbeszédet ellenőrzi, és ellenőrzési pontokat (check point) alakít ki, amelyek az esetleges párbeszéd-megszakadás után annak (utolsó ellenőrzési ponttól való) folytatását teszik lehetővé. A megteleítési réteg feladata a különböző számítógépeken futó sajátos felhasználói programokban alkalmazott eltérő adatmegjelenítési eljárások közötti különbségek áthidalása. Ebben a folyamatban szükség szerinti adatátalakítások (kódátalakítások, képmódosítások) mennek végbe, valamint titkosítás, visszafejtés, adatsűrités (kompresszió) és – visszaállítás (dekompresszió). Az alkalmazói réteg a felhasználót, annak berendezéseit és az alkalmazásait felügyeli (pl. nyomtató ellenőrzése).

Az analóg jelátvitelben az átviteli közeg sávszélessége határozza meg az adatátvitel sebességét, a digitális jelátvitelnél az egységnyi idő alatt átjutó bitek száma (bit/s) a mérvadó az adatátvitel sebességét illetően. A digitális átvitelnél gyakran alkalmazzák a baud egységet az átviteli sebesség leírására. A baud nem más, mint az egységnyi idő alatt történt változások átvitele (vagyis, hogy mennyi volt a változások száma – 0-1 vagy 1-0 változásokról van szó – egységnyi idő alatt). A sebességnövelés a meglévő fizikai közeg felosztásával történik. Az egyik felosztási lehetőség a multiplexelés, ami egyidőben több átviteli kapcsolat (kommunikációs csatorna) működését jelenti ugyanazon a fizikai médiumon (közegen). Az idő- és frekvenciamultiplexelés a leginkább alkalmazott módszerek a gyakorlatban. Az időmultiplexelés esetében az átviteli közeg több időcsatornára bomlik, a frekvenciamultiplexelésnél pedig a fizikai médium (átviteli közeg) sávszélességét (átviteli frekvenciatartománya) több frekvenciacsatornára bontják, amelyeken egyidőben különböző hordozófrekvenciájú modulált jelet továbbítanak. A frekvenciamultiplexet analóg jelek átvitelénél, az időmultiplexet pedig a digitális jelátvitelnél alkalmazzák elsősorban. Az átviteli közeg jobb kihasználása más módon is elérhető. Az üzenetek kis csomagokra osztva, egyedi célravezetéssel nagyobb átviteli sebességet

biztosít. Az adatáramlás irányát vizsgálva a kommunikáció folyamán háromfajta adatátvitelt különböztetünk meg:

1. Egyirányú (simplex) – az átviteli médiumon csak egy irányban mozognak a bitek,
2. Váltakozó kétirányú (half-duplex) – egyidőben csak az egyik irányban mozognak az adatok (bitek), de mindkét irányú adatmozgás engedélyezett, és használatos is, és
3. Kétirányú (full-duplex) – itt egyidőben mindkét irányban történik adatáramlás.

Több szakembernek is az a véleménye, hogy az optikai kábelén keresztüli adatátvitelre is érvényes Moore törvénye, miszerint az átviteli sebesség másfél évenként megduplázódik.

5.2. SZOFTVER

Ismert és elismert tény, hogy a számítástechnika fejlődése elmarad a telekommunikáció (hírközlés) fejlődési üteme mögött az utolsó tizenöt-húsz évben (egyesekek bátran húsz-harmincat emlegetnek). Lássuk a számadatokat: a hetvenes években a gyors számítógépek egy utasítást átlagosan 100 ns alatt hajtottak végre, ma erre elég nekik mindössze 1 ns. A két érték aránya 100, vagyis százszoros a feldolgozás sebességnövekedése. Ha most a digitális adatátviteli sebességeket hasonlítjuk össze, amely a hetvenes években 56 kbit/s volt, manapság pedig, mondjuk 10 Gbit/s, az arány százötvenezeres! A játék a számokkal nagyon érdekes, új ismeretekhez vezethet: összehasonlítva az optikai kábeleken mért átviteli sebességeket (1985-ben, a médiumok megjelenésekor az átvitel sebesség 50 Mbit/s volt, 1998-ban pedig 10 Gbit/s), az arány “csupán” 200. Tudni kell, azonban, hogy ez az aránylag visszafogott átviteli sebességnövekedés elsősorban a jelátalakító elektronikai berendezések lassúságának a következménye (a Lucent Bell a 2000-es évben gyors jelátalakítókkal 3,2 Tbit/s-os átviteli sebességet ért el).

Több szakember is osztja azt a véleményt, hogy a hardver és szoftver fejlettségi szintjei között a különbség még nagyobb, mint a fenti példában említett különbség a hardver és a kommunikáció között (a szoftverfejlesztés, fejlettségi szintjét tekintve a hardver mögött kullog).

Az iménti bekezdés persze nem azt jelenti, hogy a szoftverfejlesztés területén nincsenek eredmények. Mára már az emberi aktivitás úgyszólván minden területére készült valamilyen szoftver-segédeszköz. A mai szoftverek napról napra bonyolultabbak, gyorsabb processzorokat,

nagyobb központi tárat és lemezterületeket, gyorsabb grafikus kártyákat, stb. igényelnek működésükhöz. A szoftverházak közötti harc az elsőbbségért, a túlélésért, bár többnyire a nyilvánosságtól távol, a kulisszák mögött folyik, nagyon kíméletlen. Ilyen feltételek mellett az új szoftvertermékek mielőbbi megjelentetése az elsődleges cél, ez viszont az alapos vagy alaposabb tesztelések elmaradását vonja maga után (amit a szoftver működése közben jelentkező hibák – bug-ok – és az egyre sűrűsödő javítgatások – patch-ek megjelenése igazol). Az újabb alkalmazások (szoftverek) bonyolultsága talán egy másik igyekezetnek a következménye, ez pedig a felhasználói interfész egyszerűsítése. Az általánosan jelenlevő szoftverbonyolítással ellentétes folyamat csupán a mobilkészülékeknek szánt alkalmazásoknál érezhető, de ez is csak átmeneti kényszerállapot: a miniatürizálásnak, a helyszűkének a következménye. Korlátozott térben nem lehet tetszőleges gyorsaságú processzort, kapacitású központi tárat vagy külső memóriát biztosítani (itt a kihívás a hardverfejlesztőknek).

5.2.1. Operációs rendszerek

Amennyiben csak a személyi (asztali – desktop) és mobil számítógépekre szorítkozunk, akkor a következő, leggyakrabban használt operációs rendszereket megemlítenünk: Windows, Windows NT, Linux, OS2/Warp és Solaris valamint (a mobilgépek esetében) Windows CE (Windows for Pocket PC), Epoc, BeOS, Linux és PalmOS. A legelterjedtebb hálózati operációs rendszer a nyolcvanas évek végétől használatos Novell NetWare.

A bonyolultság növekedése, az új szolgáltatások megjelenése, a használat egyszerűsödése az operációs rendszerek terén is kitapintható. A mai operációs rendszerek többsége számítógép-hálózat kezelésére is fel van készítve. A legtöbbet használt Windows eredetileg 16-bites operációs rendszernek készült, és csak később dolgozták át 32-bitessé (most már 32-bites OS-nek számít, de egyesek a gyakori bukását – kék halál – épp az átalakítás tökéletlenségével magyarázzák). Sokak szerint a Microsoft első 32-bites operációs rendszere a Windows NT, ami a stabilitásában (rendelkezésre állásában) is megmutatkozik. Nem hagyható figyelmen kívül az előzőekben említett operációs rendszerek legnagyobb ellenlábas, a Linux, amely nyilvánosan hozzáférhető forráskódú, olcsó, gyors, megbízható, kevésbé sebezhető (a hackerek nem gyártanak rá vírust – hacsak a Microsoft el nem kezdi), aránylag egyszerű, sok szolgáltatással rendelkező, de aprólékosan skálázható rendszer. Egyedüli komoly hátránya, hogy egyenlőre még nagyon kevés alkalmazás (általános felhasználói program) készült rá.

A telepített operációs rendszerek számát és eloszlását nagyon nehéz megbecsülni úgy a PC-k, mint a mobilszámítógépek esetén. Úgy tartják, hogy a mobilgépeknél a két legtöbbet alkalmazott operációs rendszer a Windows CE (és a feljavított változata, a Windows for Pocket PC) és az Epoc. A Windows CE működése megegyezik a nagyobb testvér működésével (ez volt a piacradobás legfontosabb fegyverténye: nem kell egy új operációs rendszer működését megtanulni a mobilgép vásárlásakor). A Windows for Pocket PC átdolgozott grafikai elemekkel és gyorsabb működéssel büszkélkedik. Az Epoc mobilkészülékekre kifejlesztett operációs rendszer. A PSION cég azonos nevű mobil számítógépcsaládja működik ezzel az operációs rendszerrel, de megtalálható még telefonokban (smartphone) és kommunikátorokban is. Az Epoc operációs rendszer nagyon eltérő a Microsoft hasonló típusú termékeitől, az együttműködésük kielégítőnek, sőt, jónak mondható (a desktop PC-kel az adatsere terén kiválóan együttműködik).

A hálózati operációs rendszerek fejlesztése terén az első komoly mérföldkő a Novell nevéhez fűződik. Az 1994-ben megjelent Novell NetWare 4.0 tartalmazta először a NDS (NetWare Directory Service, később Novell Directory Service névre keresztelt) szolgáltatást, amely az összes lokális hálózatban található berendezés (objektum) felügyeletét lehetővé tette (szerverek, nyomtatók, lemezegységek, stb.). Az NDS filozófia később túlnőtte a Novell NetWare kereteit, más hálózati operációs rendszerre is elkészítették. A 2000-es év végén a következő operációs rendszerek rendelkeztek NDS szolgáltatással: NetWare, Windows NT, Windows 2000, Linux, Solaris 6, 7 és 8. Időözben az NDS-t összehangolták az általánosan elfogadott LDAP (Lightweight Directory Access Protocol) előírásokkal, amely természetesen a TCP/IP kommunikációs ajánlásokat támogatta. Az NDS így két elkülönült szolgáltatássá vált: az eDirectory Internetes környezetre szánt segédlet, a Corporate Edition pedig megmaradt (az NDS eredeti elképzelését megtartva) a klasszikus lokális hálózatok adminisztrátorának.

5.2.2. Segédprogramok

A segédprogramok azokat a felhasználók számára szükséges és hasznos szolgáltatásokat nyújtják, amelyekkel az operációs rendszerek nem rendelkeznek. A PC-k használatának kezdeti szakaszában könnyű volt megnevezni ezeket a programokat. Akkoriban a DOS operációs rendszert két segédprogram segítette: a PC Tools és a Norton Utilities. Manaság nehéz megnevezni a segédprogramokat a következő két ok miatt. Az első, hogy a segédprogram fogalma nem tisztázott és nem egyértelmű, a második ok pedig az, hogy az egyes gépeken, az operációs rendszer mellett működő segdprogramok a másik gépen futó opeációs rendszer szerves

részét képezik. Példa lehet erre az Internet böngészőprogramja vagy a vírusátadások elleni program.

A sikeresen működő mai szoftverházak kivétel nélkül az támogatják az e – commerce típusú (Interneten, illetve Internetről is futtatható) alkalmazásokat, illetve az ilyen fajta alkalmazások fejlesztését lehetővé tevő fejlesztőeszközöket forgalmazznak. Általánosnak tekinthető ugyanis az a vélemény, hogy az a cég, amelyik nem nyit az Internet felé, elveszett. E-commerce alkalmazásfejlesztőket sok cég kínál: az IBM, a Compaq, a Sun, az Oracle, a Netscape, a Computer Associates, a Novell, stb. Érdekes bepillantani a Novell stratégiaépítkezésébe. Először a NDS szolgáltatással állt elő a NetWare hálózati operációs rendszer szolgáltatásaként, majd ezt a szolgáltatást több más hálózati operációs rendszerre is átdolgozta. Újabb lépésként két irányban fejlesztette tovább (eDirectory és Corporate Edition), majd további szolgáltatásokkal bővítve egy közös One Net elnevezésű égisz alatt ismét összefogta. Az One Net a felhasználói szolgáltatások csoportját NSS-t (Net Services Software) jelöli: a Net Management Services-t (objektumok karbantartását – felhasználók, hardver-berendezések, szoftver-termékek és alkalmazások, stb. – végzi), a Net Content Services-t (a hálózatok adatelosztásának és adatrögzítés-eloszlásának irányítását teszi lehetővé) és a Net Portal Services-t (a felhasználók azonosítása és az e – commerce funkciók megvalósítása a feladata). Az egyes szolgáltatásokhoz különféle termékek tartoznak, amelyek közül talán a ZENworks (Zero Effort Networks) a legismertebb. A ZENworks a kiszolgálógépek (szerverek) és munkaállomások adminisztrációját végzi a hardverkiépítettség, a telepített szoftverek és a vírusvédelem terén.

5.2.3. Felhasználói programok

A felhasználói vagy applikatív programok (alkalmazások) valamilyen konkrét programnyelven íródnak, amelyhez manapság már törvényszerűen valamilyen fejlesztői környezet tartozik. Valamennyi fejlesztőkörnyezet adatbáziskezelésre is alkalmas alkalmazások fejlesztését teszi lehetővé. A ma használatos objektumorientált programnyelvek termékei, az alkalmazások, a következő három adatbázis-kezelési architektúra egyikét (némelyik mindhármát) valósítják meg:

1. fájl szerver (File server),
2. ügyfél/kiszolgáló (Client/Server) és
3. többretegű (Multi-tier).

Minden adatbáziskezelést is tartalmazó alkalmazás a következő három funkcionális egységből épül fel:

1. Közvetlen adatkezelés (Data processing),
2. Alkalmazáslogika (helyenként az üzleti logika – Business logic – kifejezést használják), és
3. Felhasználói felület (User Interface).

A közvetlen adatkezelés alatt a fizikai feldolgozást értjük: az adattárak nyitását, zárását, az adatok adattárakban való rögzítését (a felvitelt), a rögzített adatok módosítását, törlését, az állományok indexelését, a lekérdezések optimalizálását és végrehajtását, a zárolási konfliktushelyzetek feloldását, stb.

Az alkalmazáslogika biztosítja a létrehozott szoftver (alkalmazás) biztonságos és helyes működését valamint kényelmes használatát. Gondoskodik az adatvédelemről (a felhasználók azonosításával és jogosultságaiknak megfelelő funkcióik végrehajtásával), az adatok integritásáról (elronthatatlanságáról: **értelmezésitartomány-integritás** (v. megszorítás) – *domain integrity* – az adatok típusának, hosszának és a tervező által megfogalmazott feltételeknek ellenőrzésével, **kulcs-megszorítás** – *key integrity* (sz.: integritet entiteta) – a kulcs értékeinek egyediségére vonatkozó ellenőrzés, ti. ezeknek egyértelműen kell azonosítaniuk az egyedelőfordulásokat, vagyis, hogy a kulcs illetve annak minden összetevője, ha összetett kulcsról van szó nem lehet ismeretlen értékű (NOT NULL), **hivatkozásiépség-megszorítás** – *referential integrity* – , amely a relációk közötti kapcsolatípus megszorítása, ti. a külső kulcs értéke csak akkor vihető be a táblába, ha az a kulcsérték az egyes oldalon levő táblában elsődleges kulcsértékként már létezik, vannak még egyéb általános az alkalmazás v. üzleti életdiktálta megszorítások is – ezekről majd másutt többet szólnunk), a feldolgozási adatműveletek csoportosításáról, a tranzakciófeldolgozás vezérléséről stb.

Az utóbbi időben kifejlesztett alkalmazások a felhasználóval sajátos módon, ablakon keresztül “érintkeznek”. A felhasználói felület ebben a bizonyos ablakban alakítható ki, az összes lehetőségeivel együtt (fő- és almenü rendszer, felbukkanó menük, adatbeviteli képernyők, sűgóablakok, stb), amelyeknek az elvégzendő feladat megkönnyítése a céljuk: könnyen és gyorsan áttekinthetőnek kell lenniük, érthetőnek és szimpatikusnak.

A fájl-szerver architektúra megvalósításánál, ha minden aktivitás egy számítógépen bonyolódik, egyfelhasználós alkalmazásról beszélünk. Hálózatban működő változatban, többfelhasználós alkalmazás esetében a fájl-szerver architektúra nehézkes működésű: minden felhasználó az adatfeldolgozáshoz a szerverhez fordul a számára szükséges (feldolgozandó)

adatokért, melyeket a hálózaton keresztül meg is kap, és csak ezek után dolgozhatja fel őket a saját számítógépén. Az egyes feldolgozásokhoz szükséges adatmennyiség több megabájtot is kitehet, így gyakran a hálózat válik a feldolgozás szűk keresztmetszetévé. Ezt a típusú architektúrát valósítja meg a dBase, a FoxPro, a Clipper, a Paradox, a Visual Basic, az Access, stb. Ezt az architektúrát nevezik néha egyrétegűnek is (single – tier).

Az ügyfél/kiszolgáló architektúra két, jól elkülönülő rétegre tagolódik, éppen emiatt gyakran hívják kétrétegűnek is (two – tier). A szerveren az adatbázis karbantartásával kapcsolatos feladatkörök bonyolódnak (adatbázisgép – back-end machine), valamint az alkalmazási logika dandárja (adatintegritási v. megszorítási ellenőrzések, tranzakciók feldolgozása) beépített triggerek és rögzített eljárások segítségével. Ezt a szoftvert hívják adatszervernek, adatbázis-szervernek, de adatbázisgépnek is. A második rétegen, a felhasználó számítógépén az alkalmazási logika szerveren nem telepített, maradék része helyezkedik el, és természetesen a felhasználói felületet legtöbbször grafikus kezelő – GUI-t (Graphical User Interface) működtető programok. A feldolgozás teljes egészében a szerveren bonyolódik, a hálózaton keresztül csak a felhasználó kérései (SQL – Structured Query Language – lekérdezései) utaznak a szerver irányában, valamint a feldolgozás eredményei az ellenkező irányban. Ilyen módon működnek a következő adatszerverek: Microsoft SQL Server, Oracle, Informix, Sybase, MySQL, InterBase, stb.

A többrétegű architektúra esetében az alkalmazás nem kettő, hanem több részre, rétegre oszlik. A leggyakoribbak a legegyszerűbb többrétegű architektúrák: a háromrétegű megvalósítások. Itt a szerveren, az első rétegen helyezkednek el a közös adatok és az adatbáziskezeléssel kapcsolatos szoftver, a középső rétegen az alkalmazási logikát megvalósító alkalmazásrész (pontosabban az alkalmazáslogikának az a része, amely nem építhető be a logikai adatstruktúrába, illetve az adatbázisba), a harmadik réteg csupán a grafikus felhasználói felület (GUI) megvalósításáért felelős, ezért gyakran nevezik „sovány” ügyfélnek/kliensnek (thin client).

A Microsoft 1999 közepén új típusú alkalmazásfejlesztő stratégiát jelentetett be .Net (dotnet) név alatt. Ezt az új elképzelést valósítják meg a bejelentés után piacra dobott Visual Studio környezetbe foglalt fejlesztőrendszerek is: a Visual Basic, a Visual C ++, Visual FoxPro, Visual InterDev, a Visual J++ és a Visual SourceSafe). Az új fejlesztési elv első változata 2000 júliusában öltött testet .Net Framework néven. Ez képezi az alapját az újab, most már Visual Studio.NET-nek elnevezett fejlesztőrendszer-családnak. A mai gyors fejlesztést eredményesen végző fejlesztőkörnyezetek, mint a Visual C++ nagy szakértelmet és odafigyelést igényelnek a programozótól még a nagyon apró részletek megvalósításánál is (pl. kézi tárterület- és egyéb

erőforrás-felszabadítás), ami az eredményesség (termelékenység) rovására természetesen negatív értelemben hat ki. Az új erőfeszítések az ilyen gyöngeségek kiküszöbölésére irányulnak, gyorsítva a rendszermegvalósítás (programozás) ütemét (a programozónak az alkalmazási logika megalkotása lesz a legfőbb gondja).

Az elsőszámú (vagy rendű) újdonság az, hogy a .Net Framework közös végrehajtási modult lát elő a Visual Studio.Net számára, közelítve ezzel egymáshoz a fejlesztői környezeteket (a közös végrehajtási mag segítségével). A másik nagy újdonság a különböző fejlesztési környezetben felkínált (biztosított) osztályok szabványosítása, és ezeknek közös API (Application Programming Interface) könyvtárban aló elhelyezése. Ez a lépés az állítások szerint átjárhatóságot (az osztályok örökléthezességét) biztosít majd a fejlesztőrendszerek között, ami közös hibakezelők és nyomkövetős tesztelők (debug) kidolgozását, stb. teszi lehetővé. A harmadik újdonság a .NET Framework-ben az ASP+ (Active Server Pages+), ami olyan környezetet biztosít, amelyik a Web-alkalmazások egyszerű és gyors fejlesztését segíti. A COM (Component Object Model) stratégia továbbra is a komponensalapú automatizált fejlesztés fő mozgatórugója marad, mert a .NET Framework átvállalja a kidolgozott és letesztelt részek (komponensek) rendszerbe való iktatását (regisztrációját), a felhasználói felület leírását (dokumentációját) és elérhetőségeiknek létrehozását. A COM az objektumok eljárásaima (metódusaira) és felhasználói felületére vonatkozó szabvány a kizárólagos használatuk megkönnyítése érdekében. A COM objektumok hálózati használatát a DCOM (Distributed COM) előírások tartalmazzák. A Microsoft egy COM+ technológia bevezetésén fáradozik, amely a COM, az MTS (Microsoft Transaction Server) és a DCOM szabványokat ötvözné, és az alkalmazásfejlesztés középső rétegének (middle – tier) legfontosabb szolgáltatása lenne.

A Microsoft új, .NET stratégiájára több (de legalább két) okból is illik odafigyelni. Egyrészt a jelek azt mutatják, hogy sok pénzt szántak rá, és sok embert állítottak rá a fejlesztésére, másrészt viszont el kell ismerni, hogy a szoftverfejlesztés stratégiáját (tetszik-e ez nekünk vagy sem) még ma is a Microsoft alakítja.

6. AZ INFORMÁCIÓS RENDSZER FEJLESZTÉSE

Az előzőekből tudjuk, hogy az információs rendszer az inhomogén rendszerek családjába tartozik, elemeinek (tényezőinek) és a tényezők közötti kapcsolatainak eltérő természete miatt. A különmemű tényezők közötti kapcsolatok felderítése kemény munkát követel a tervezőktől, akik a valóság, illetve az információs rendszer megközelítésének módját (irányát) változtatva próbálják megtalálni a lehető legjobb utat az élet bonyolultságának feloldására. Egyesek (tévesen) az eszközöket tartják elsődlegesnek, mások a felhasználók által megfogalmazott igények, feldolgozások összességét tekintik mérvadónak (ugyancsak tévesen, mert a felhasználók gyakran szeszélyesek, maguk sem tudják mindig pontosan, hogy mit akarnak), és vannak akik (teljesen igazoltan, a mai ismereteink szerint) az adatokat (a modellezendő üzleti szféra fogalmait) részesítik előnyben.

Más szempontból az is tisztázott dolog, hogy az információs rendszer csupán másodlagos eredendőségi szempontból az elsődleges valós rendszer (a fizikai valóság) mögött. Az információs rendszer feladata a valóság tükrözése (hű visszaadása), a „tükörkép” pedig akár többszörös is lehet (a visszatükrözés tökéletlensége folytán). Még a valóságkép egyedisége és egyszerűsége mellett is számolni kell a megjelenítési formák különbözőségével (másképp jelennek meg az adatok és bonyolódnak az üzleti élet eseményei a formanyomtatványokon, vagyis a valós életben, és másképp az tervező képzeletében, ismét másképp a számítógépen). Az információs rendszer létrehozásakor ezért általánosítások segítik a valós világ információs rendszerrel való megfogalmazását.

Elv, eljárás, technika, folyamat Az információs rendszer fejlesztése különféle konkrét elvek, eljárások, technikák, folyamatok és eszközök alkalmazását követeli meg a fejlesztési eljárás vagy az életciklus folyamán. Az információs rendszer életciklusa és fejlesztése természetesen különböző fogalmakat takarnak. Sokan vitatják a fogalmak (kifejezések) jelentését, úgy elméleti mind gyakorlati szinten, sőt szinonimaként válogatva is használják ugyanazzal a jelentéssel. Az egyértelműség érdekében fogadjuk el a következő nézőpontot (meghatározást) az említett fogalmakkal kapcsolatban. Az információs rendszerek életciklusa három fázisból áll:

1. kialakítás,
2. működtetés és
3. változás.

A másik fogalom, a rendszerfejlesztés a következő aktivitásokat foglalja magába:

1. elemzés,

2. tervezés,
3. megvalósítás,
4. működtetés – felügyelet,
5. változáskezelés – change management.

Az információs rendszer életciklus-fázisainak a rendszerfejlesztés aktivitásaival való összevetéséből az derül ki, hogy az életciklus első fázisa az elemzést, a tervezést, és a megvalósítást fedi le. Felvetődik a kérdés, vajon a működtetés – felügyeleti aktivitás fejlesztésnek tekinthető-e egyáltalán! Amennyiben a működtetés felügyelettel nem párosul, a válasz nemleges, ha viszont a működtetést felügyelet kíséri, akkor a rendszerfejlesztés szerves részének tekinthető.

Az információs rendszerek fejlesztési paradigmája alatt olyan általánosan érvényes és elfogadott elvek, szemléletek, fejlesztési megközelítések gyűjteményét értjük, amely lehetővé teszi a valós rendszer (elvárási szempontokból fontos) általános tulajdonságainak és viszonyainak valamint törvényszerűségeinek a hű leképzését (tükrözését).

A megrendelő – felhasználó minőségi munkát vár a fejlesztőktől úgy a kialakítás (hasznos és naprakész dokumentáció, minőségi és tartalmas, közös – tervezői és felhasználói részvételű – munkamegbeszélések), mint az üzemeltetés (az információs rendszer pontosan, teljességében és csakis a közösen megfogalmazott elvárásokat végezze el¹, nagyfokú rendelkezésre állással rendelkezzen – ne „bukjon le” – , felhasználóbarát – könnyen áttekinthető, érthető, és a felhasználót sugóval segítő – legyen), valamint a változás-karbantartás (az új elvárások, változtatások könnyen véghezvihetők legyenek, vagyis a change management aktivitásai minél rövidebb ideig tartsanak) folyamán. Ezt a szempontot figyelembe véve megfogalmazható egy fejlesztéstechnológiai fejlődés-vonal, amelyet a későbbiekben ismertetünk.

A fejezet első pontjában az információs rendszerek fejlesztéstechnológiai fejlődését kísérvük végig, nagy vonalakban, az első gépektől a mai napig, majd a várható felődési irányokat villantjuk fel. Mindenképpen hálátlan dolog ezen a gyorsan fejlődő műszaki-tudományos területen jóslatokba bocsátkozni, mert G.Winberg szavaival élve: „A jövővel az a legnagyobb problémánk, hogy semmi sem olyan, mint amilyen lenni szokott!”². Ezért szorítunk csak a lehető fejlődési irányzatokra.

A gyors és eredményes információs rendszerfejlesztés feltételezi a fejlesztői csoport tagjainak összehangolt működését és fejlesztési elképzeléseit, az információs rendszerrel és annak tényezőivel, fejlesztési sorrendjükkel (ütemezésükkel), az ezeket biztosító aktivitásokkal

¹ Ahogy erre Dr Halassy Béla felhívta a figyelmet: ugyanúgy, mint az amerikai törvény által tanúk számára előírt felesketési szövegben: truth, all the truth, and only the truth.

² Forrás: [22.], 674. oldal

az elemzés és tervezés viszonyával, a rendszerdokumentációval, a modellezési és minőségellenőrzési kritériumokkal kapcsolatosan. Ezzel foglalkozik a fejezet többi része.

A fejezet második pontja a rendszerfejlesztési elveket ismerteti, azok közül is csak az általános rendszerfejlesztési elveket taglalja: az ütemterv kidolgozását, az elemzés és tervezés viszonyát, a rendszertaglalási szempontokat, a fejlesztési lépések fokozatos, iteratív végrehajtását. A fejlesztés folyamán az elvonatkoztatási készség elengedhetetlen a rendszermodell megfogalmazásához, de a tervezőcsoport egy pillanatra sem feledkezhet meg arról a tényről, hogy a megrendelő számára készülő információs rendszert megvalósító szoftverterméket (adatkezelő, -feldolgozó és –megjelenítő részekkel) tesztelten és jól (általánosan ismert és elfogadott szaknyelven) dokumentáltan kell leszállítani. Erről is szó lesz majd a második pontban.

A harmadik pont a módszer és módszertan fogalmak különbségét világítja meg. E két fogalmat az informatikában gyakran tévesen szinonimaként használják. A metodológia a módszerek szerkezetét és felépítését tanulmányozza, ezért a módszerek módszerének is tekinthető.

6.1.A RENDSZERFEJLESZTÉSI TECHNOLÓGIA FEJLŐDÉSE

A számítógépesítés hőskorában a szoftver fogalma csupán az operációs rendszereket, a programnyelveket, a különböző adathozzáférési módokat megvalósító programokat és a segédprogramokat ölelte fel. A szoftverfejlesztés teljes mértékben hardverfüggő volt, a számítógépeket csak a hozzáírt szoftverekkel együtt árusították. A szoftver hordozhatósága akkoriban még ismeretlen fogalom (és elvárás) volt: a konkrét géptípusra kidolgozott szoftver más gépeken nem volt futtatható. Saját alkalmazást (információs rendszert) azidőtájt csak a legnagyobb felhalmozást megvalósító cégek fejleszthettek ki: a nagy bankok és biztosító társaságok. Elsősorban (és szinte kizárólag) anyagi okokból. A korabeli információs rendszerek elsődleges ismérve a testreszabottság volt, hiszen a megrendelő igényei szerint készültek, kulcsrakész megvalósításban. Később a megrendelő cégek saját fejlesztési, működtetési és karbantartási csoportot hoztak létre az információs rendszer olcsóbb és jobb, hatékonyabb működtetésének reményében.

A hardverkomponensek árszintjének esése a nagyobb sorozatban gyártott számítógépek megjelenéséhez kötődik, különösképpen pedig a hetvenes évek elején piacra dobott mikroprocesszorokéhoz. A nagyszámú számítógép-felhasználó több közös új igénnyel jelent

meg a szoftverpiacon, ami a hetvenes évek folyamán a számítógépgyáraktól független szoftverházak megjelenéséhez vezetett. A szoftverházak szélesebb felhasználói rétegek által igényelt problémák megoldására szolgáló termékeket gyártottak, és szoftvereiket rendre nagy példányszámban tudták értékesíteni.

A szoftverfejlesztés mérnöki szigorát elsőként Fritz Bauer szorgalmazta, még 1968-ban, a Garmisban tartott NATO Tudományos Konferencián. Ettől kezdve alkalmazzák a szoftverfejlesztési technológia, rendszerfejlesztési technológia és szoftverfejlesztési tevékenység (Software Engineering) fogalmakat. Itt fogalmazódott meg az a nézet, amely ma már teljesen természetes: a szoftvert más, ipari termékhez hasonlóan mérnöki módon kell fejleszteni.

“A szoftvertermékek ipari fejlesztésének elméleti alapjait Brad Cox fektette le a nyolcvanas évek közepén. Nézete szerint a bonyolult szoftvertermékek kiépítését egyszerűbb, előre kidolgozott objektumok, komponensek felhasználásával kell megvalósítani. A szoftver ilyen, ipari jellegű fejlesztése, a kisipari jellegű fejlesztés fejlődésével jön létre, nem megsemmisítve, csupán háttérbe szorítva a „kézműipart”³

A szoftverfejlesztési tevékenység (Software Engineering) egy olyan fejlesztési folyamat, amelyben gazdaságos és minőségi szoftvergyártás folyik a mérnöki szigor és fegyelem betartásával, tudományos és matematikai módszerek alkalmazásával egy valós probléma számítógépes megoldása érdekében.

A szakirodalom a szoftverfejlesztési tevékenységet a fejlesztés tárgyától, illetve a megoldandó probléma jellegétől függően három típusúra osztja:⁴

1. system engineering,
2. information engineering és
3. knowledge engineering.

A fenti felosztás nem általános érvényű, különböző forrásokban a felsorolt típusokat megnevező fogalmak más-más tartalmat takarnak (homonímia). A fogalmak különböző értelmezésére vegyük górcső alá a rendszerfejlesztés és a szoftverfejlesztés közötti különbséget leíró következő meghatározást (itt, helyesen a rendszerfejlesztés általánosabb volta a kidomborított lényeg):

“A “rendszer-mérnökség” általánosabb fogalom, mint a “szoftver-mérnökség”, mert magába foglalja a rendszerfejlesztést, a rendszerdokumentációt, a megvalósítást, a tesztelést, a telepítést és a rendszerkarbantartást. A “rendszer-mérnökség” több, nem szorosan informatikai témakörhöz, hanem a működtetéshez elengedhetetlen elemet is tartalmaz. A rendszertervezés interdiszciplináris folyamat, melynek lebonyolításához a tervezőcsoport(ok) különböző

³ Dr Pere Tumbas: ibidem.

képesítésű (szakterületű) és szerepű tagjai együttesen (közös felelősséggel) alkotják meg a rendszert. A már kialakított rendszert csak nagyon indokolt esetben ajánlatos újratervezni (redesign), mert az utólagos változtatások kivételesen bonyolultak, drágák és időigényesek.”⁵

A Software Engineering a maga „alváltzataival” az irodalom szerint [22], az információs rendszerek ipari fejlesztését is lefedi (az általunk elfogadott információs rendszer definíció alapján a helyzet éppen fordítottnak lehet igaz). A változatok (system engineering, information engineering és knowledge engineering) úgy az összetettségben, mint az adatfeldolgozás mikéntjében és technikai megoldásában is különböznek egymástól. Az információs rendszerek fejlesztése szabványok nélkül komoly gondokkal küzd, ám az ilyen gyorsan fejlődő ágazatnál éppen a szabványok kidolgozása a legnagyobb probléma. A meglévő fejlesztési szabványokat általában egy, vagy a szoftverházak kis csoportja hozta meg az egyes szakterületeken. Ezek alapján kellene kidolgozni a nemzetközi szabvány-előírásokat (amelyek idáig felelőtlen késedelemmel jelentek meg).

Az információs rendszerek fejlesztési technológiájának fejlődésére vitathatatlanul a számítógépes támogatás gyakorolta a legnagyobb hatást (legfőképpen a fejlesztési fázisok – lépések – ideje alatt keletkező adatok rögzítése és rendezése területén). A tervezés folyamán keletkező különféle adatok a ma repository-nak nevezett fejlesztési adatbázisba kerülnek (régebben a hasonló célból biztosított lehetőséget/tárterületet adatszótárnak – data dictionary-nak – nevezték). A repository az információs rendszerterv több változatát is kezelheti, és a felhasználók hozzáférését is felügyel(het)i. Az utóbbi időben több olyan szoftvertermék is megjelent, amelyek az információs rendszerek fejlesztésének teljes folyamatát, vagy a fejlesztés csak bizonyos fázisait automatizálják (elvégzik, átvállalják, a dokumentáció-generálással egyetemben – itt a fáradtságos, rutinfeladatok számítógéppel való elvégzéséről van szó). Ezeket a szoftvertermékeket CASE eszközöknek nevezzük. A betűszó kuriozitásához a kétféle felodása tartozik: Computer – Aided Software Engineering, és Computer – Aided Systems Engineering.

Pillantsunk be most egy kicsit a futurológusok (jövendőmondók) boszorkánykonyhájába a hardver és szoftver jövőbeli fejlődésével (fejlődési arányaikkal, paritásával) kapcsolatosan, illetve a szoftverkereslet és –kínálat viszonyát illetően.:

“A számítógépek minőség/ár aránya milliószorosára növekedett az elmúlt húsz esztendőben, és Gordon Moore megfigyelései szerint ez a jelenség az elkövetkező negyven évre is érvényes marad. Utána az említett arány eléri növekedési telítettséget, más szóval az aráynövekedés gyors ütemben elenyészik. Moore törvénye (melyet az alábbiakban ismertetünk)

⁴ Forrás: [22.], 157-158 oldal.

⁵ Dr Pere Tumbas: ibidem.

alapján megállapították az alapvető szoftvertörvényeket, amelyek a szoftverfejlődés irányvonalait körvonalazzák:

- a szoftver gázként terjed évi 33,9%-os növekedéssel,
- a szoftver terjedését Moore törvénye korlátozza,
- a szoftverfejlesztés növekedése Moore törvényének hardverre való hatását erősíti (biztosítja, meghosszabbítja az érvényességi idejét),
- a szoftver fejlődésének egyedül az emberi faj lehetőségei és igényessége szab korlátot, és
- a szoftver „elnyúlhatatlan” termék (nem kopik, nem törik, nem szakad, nem megy tönkre), amely a korlátlan hosszú idejű használatát teszi lehetővé (sokak szerint ennek a következménye az Y2K – vagyis a 2000-es év – problémája, senki sem gondolta, hogy a dátum rögzítésének szoftver-megoldása ilyen hosszú ideig fennmarad).

Az előzőekből leszűrhető következmény a szoftverkrízis folytatódására mutat, és arra, hogy a 21. század legfontosabb iparága a szoftverfejlesztés lesz.”⁶

Moore törvénye a számítógépek alapelemeinek számító integrált áramkörök fejlődési görbéjére és azoknak áralakulására vonatkozik. Gordon Moore megfigyelte, hogy a

1. processzorok sebessége, valamint a sebességük és áruk aránya,
2. az operatív tár egy bitre vonatkozó ára, valamint az egy chipbe tömöríthető bitek száma, és
3. a háttértár-lemez egy bitre vonatkozó ára, valamint a lemezen tárolható bájtok száma tizennyolc hónaponként megduplázódik.⁷

6.2.INFORMÁCIÓS RENDSZERFEJLESZTÉSI ELVEK

Az információs rendszerek gyorsabb fejlesztése érdekében a fejlesztőcsoport tagjainak ismerniük kell a rendszerfejlesztési elveket. Az információs rendszerfejlesztési elveket a szakemberek többsége a következő három csoportra osztja:

1. általános rendszerfejlesztési elvek,
2. a fejlesztési lépések megvalósításának elvei (fejlesztési modellek is nevezik őket), és
3. a rendszerfejlesztés megközelítésének elvei (fejlesztési filozófiák néven is ismertek).

⁶ Dr Pere Tumbas: ibidem.

⁷ Forrás: [32], 50. oldal.

Ez az egyszemeszteres kurzus csupán az általános rendszertervezési elvek ismertetését teszi lehetővé, a fejlesztési modellek és a fejlesztési filozófiák ismertetése egy második, magasabb fokú kurzus témája lesz.

6.2.1. Általános rendszerfejlesztési elvek

Az általános rendszerfejlesztési elvek az információs rendszerfejlesztés általános “ökölszabályait”, szemléletét, normáit, a fejlesztőcsoport együttműködésének és döntéshozatalainak szabályait.

6.2.1.1. A háromszoros és háromszintű tervezés

Az információs rendszer tervezésének az eredménye a rendszerterv, amelyet meg is kell valósítani. A fejlesztést (a rendszertervezést) ugyancsak tervezni kell, ami a projektmenedzsment feladatkörébe tartozó probléma, azzal a megjegyzéssel, hogy az információs rendszerfejlesztés sajátos jegekkel bír, mint a többi tervfajták is, általában. Az információs rendszer terve, a többi projekttervhez hasonlóan, három tervből áll össze:

1. átfogó (globális) terv (plan),
2. részletes terv (design), és
3. kivitelezési- vagy ütemterev (schedule, project-plan).

A projektmenedzsment nagyon dinamikusan fejlődő tudományterületnek számít az utóbbi időben, így a három fajta terv részleteibe nem bonyolódunk bele. A téma után érdeklődők az információs rendszerfejlesztések menedzselésével kapcsolatban a nagyszámú, rendelkezésre álló irodalomból elégíthetik ki kíváncsiságukat: [3], [21]. Itt csak a tervek nagyvonalú ismertetésére szorítkozunk. A tervek kidolgozásában való jártasság (különösen vonatkozik ez az átfogó tervre) nagy előnyöket jelenthet a nyilvánosan kiírt pályázatok (tender) elbírálásánál és elfogadásánál.

Az információs rendszer fejlesztési folyamatának beindítása előtt a fejlesztés egy elsőleges, durva becslése kerül papírra az átfogó terv keretében. Itt a cél (esetleg célhierarchia), a nagyvonalú tartalom, a szerkezet (architektúra), a működtetés hardver és szoftverkönyezete, a várható költségek és a fejlesztéshez szükséges idő kap helyet.

A részletes terv csak a valós rendszer alaposabb ismerete alapján készíthető el. Az előkészületben levő információs rendszer által kezelt adatok, az általa támogatott események és

az események által kiváltott tevékenységek ismeretében már nagyobb pontossággal becsülhetők az erőforrások: az idő, a pénz és a technikai eszközök (hardver, szoftver, kommunikáció, stb.). Az előző mondatban a hangsúly a „nagyobb pontossággal” szavakon van, ugyanis teljes, abszolút pontosság nem létezik. Az informatikailag fejlett országokban már meghonosodott a „csúszótervezés” fogalma, amelyet az erőforrásigények (különösen a fejlesztési idő és a - költségek) többlépéses meghatározására használnak. Ezek a becslések az erőforrások meghatározásában folytonos finomításokat, pontosításokat jelentenek, és a részletes tervdokumentáció kiegészítését alkotják.

A rendszerfejlesztés kivitelezési- vagy ütemterve a munkák, a rendelkezésre álló munkaeszközök és munkavégzők számának ismeretében készül. Itt a feladatok elosztásának és ütemezésének becslése történik munkavégzőkre és munkaeszközökre a végrehajtáshoz szükséges idővel együtt.

A projekt fogalmának megvilágítására elfogadjuk a következő meghatározást: a “projekt minden olyan tevékenység, amely egy szervezet számára olyan egyszeri és komplex feladatot jelent, amelynek teljesítési időtartama (kezdés és befejezés), valamint teljesítésének költségei (erőforrások) meghatározottak és egy definiált cél elérésére irányul”.⁸

Az információs rendszer fejlesztéséhez biztosítani kell a megfelelő erőforrásokat, és a feladatok elvégzésének összehangolását. A fenti meghatározás szerint ez az ütemterv feladata, amelynek kidolgozásában többféle képzettségű szakemberek vesznek részt: informatikusok, közgazdászok, kommunikációs szakemberek, számítástechnikusok, jogászok.

A tervezés folyamán nem szabad szem elől téveszteni az információs rendszerek három vetületét (adat-, feldolgozás- és környezetvetület), és három absztrakciós szintjét (fogalmi, logikai és fizikai szint).

“Tervezési-folyamatnak (design-process), azaz tervezésnek hívjuk azt az időben lezajló tevékenység-sorozatot, amelynek során az információs rendszer adott vetületekbe és szintekbe sorolt elemeit meghatározzuk illetve azokat összehangoljuk.”⁹

Az információs rendszer tervezése a rendszerfejlesztésnek (pontosabban a rendszerkialakításnak) csak része, hiszen a fejlesztéshez egyéb, más tevékenységek is tartoznak, úgy mint a megvalósítás (programozás, tesztelés), a felügyelet és az új követelményekhez való alakítás (változtatás – change management).

A terv vagy terv-termék (design-product) a tervezési folyamat végterméke, de a fejlesztés nem egyetlen produktuma. A fejlesztés (kialakítás) termékei például a programok, vagy a dokumentáció.

⁸ Forrás: [8.], 14. oldal.

6.2.1.2. *A rendszerelemzés és a rendszertervezés viszonya*

Az információs rendszerfejlesztés két fontos tevékenységsorozata (tevékenységgör, aktivitása) a rendszerelemzés (analysis) és a rendszertervezés (design). A két tevékenységgör viszonyáról még ma is megoszlanak a vélemények. Az egyik tábor azt vallja, hogy az elemzésnek és tervezésnek nincsenek közös (átfedő) pontjaik (semmi közük egymáshoz). Ezt igazolják azok a rendszerfejlesztési filozófiák, amelyeknél az elemzési és tervezési fázis teljesen különálló. A másik oldalon az ezzel ellekezőt bizonygatók hada, mondván, hogy a két fázis szétválaszthatatlan. Ez utóbbi tábornak is vannak érvei: a tervezési folyamat megoldási változatokat készít elő, és a legmegfelelőbb változatot dolgozza ki részletesen, de a kiválasztáshoz nélkülözhetetlen a változatok összehasonlítása (elemzése). No, akkor most kettő az igazság vagy pedig egy? Hihetetlenül hangozhat a válasz, de mindkét fél érvelése helyes.

Az ellentétek feloldására közelítsünk az elemzéshez két különböző oldalról. Tekintsük az elemzést egyrészt tevékenységgörnek, de a másik oldalról felfogható (az élet ezt igazolja is) egy munkafázis (tevékenységgör) részttevékenységének is. Ez utóbbi nézőpontot a tervezési fázisnak, a tervváltozatok közötti választást megelőző elemzése igazolja.

A fejlesztési igényt két féle jelenség (állapot) válthatja ki. Az egyik negatív állapotot tükröz vissza: valami nem megy a rendjén az információs rendszer működtetése folyamán, a másik egy jónak tűnő ötlet a valós világ tökéletesebb tükrözésére az információs rendszer segítségével (az információs rendszer valamelyik tényezőjével kapcsolatos új, jobb megoldás). A működési rendellenesség első lépésben a tényfeltárást (fact finding) indítja el, az új ötlet pedig a kiértékelést (evaluation). Bármilyen célból is indul el a fejlesztés, szükségszerű a cél (célhierarchia) megfogalmazása. A fejlesztésnek azt a fázisát hívjuk elemzésnek, amelyik a fejlesztés célját megfogalmazza, és a kitűzött cél szükségességét és helyességét (értelmét, megvalósíthatóságát) vizsgálja.

A rendszerelemzés két szinten fogható meg:

1. absztrakt és
2. konkrét szinten.

Az absztrakt szinten a rendszerelemzés folyamatának elvi lényegével és felépítésével foglalkozunk, azokkal a lépésekkel (vizsgálati szempontokkal), amelyekkel a feltárás/kiértékelés folyamán számolnunk kell (ki kell dolgozni). Ez a problémakör a módszertanok tárgyához tartozik. A rendszerelemzés konkrét megvalósításánál a következő lépések (vizsgálati szempontok) kerülnek végrehajtásra:

⁹ Forrás: [11.], 122. oldal.

1. az információs rendszer céljának, illetve célhierarchiájának megfogalmazása,
2. a céltól elválaszthatatlan tartalom meghatározása (a tartalom az információs rendszerrel támogatni kívánt valós rendszer fogalmaiban rejlik),
3. az információs rendszer struktúrájának/statikájának (tényezői statikus elrendezésének) vizsgálata és leírása,
4. a rendszerdinamika (a tényezők dinamikus viszonyának) felderítése és dokumentálása (a kivételek felkutatása és feljegyzése a működésben és a statikában/dinamikában néhol külön lépésként szerepel, mi ehhez a ponthoz csatoljuk), és
5. az előző négy ponttal kapcsolatos fejlődési irányok feltérképezése és rögzítése a valós rendszerben, valamint ezeknek az információs rendszerre gyakorolt hatásának leírása.

A rendszerelemzés konkrét szintjén egy kiválasztott rendszer megismerését, feltárását és kiértékelését értjük a fent megfogalmazott vizsgálati szempontok szerint, és erre a szintre (tevékenységgörre) használjuk a rendszerelemzés (systems analysis) fogalmát.

Az előzőekben leírt feltárási-kiértékelési fázis után a fejlesztésben a rendszerelemzés és a rendszertervezés kölcsönösen egybefonódnak.

A fejlesztés folyamán a két említett fázis (elemzés és tervezés) problémamegközelítése is eltérő. Általánosan elfogadott elv, hogy a fejlesztés elején (az elemzési fázisban) az egésztől közelítünk a részletek felé (dekompozíció segítségével), és ezt a megközelítést hívják top – down típusúnak. A tervezés problémamegközelítési iránya pontosan fordított: az elemi tényezők segítségével építjük ki az információs rendszert. Ennek a megközelítésnek bottom – up a neve.

6.2.1.3. *A strukturálási és iterációs elvek*

Ismert és elfogadott az a tény, miszerint az összetett dolgok megismeréséhez az alkotórészeikre való bontásukon keresztül visz az út. A már kicsit is összetettebb (valós) rendszerek esetében lehetetlen egyszerre átfogni az összes fogalmat, eseményt és tevékenységet. Bizonyított, azonban, az is, hogy a rendszertagolás (részekre bontás) módja kihat a tervezésben lévő információs rendszer majdani működésének sikerességére (hatékonyságára). A rendszerek tagolása többfajta elv szerint végezhető, ezek közül a következő öttel ismerkedünk meg:

1. földrajzi elv,
2. eszköz elv (hardver- és szoftvereszközökre kell gondolni),
3. szervezeti elv,
4. funkcionális elv és

5. ismereti elv (adatok szerinti tagolás elve).

A földrajzi elv a több telephellyel rendelkező szervezetek lehetséges tagolási elve, amikor a különböző szintű telephelyi funkciók nem egyetlen rendszer részeként (elosztottan, disztribuíltan), hanem külön rendszerekként (szétszórtan) valósulnak meg, elveszítve ezzel a rendszer egységességét és teljességét.

A rendszerstrukturálás eszköz-elve hardver szempontjából az információs rendszer nagygépes, PC-s, munkaállomásos, stb. felosztását szorgalmazza, a szoftver szerinti tagolás pedig az információs rendszert működtető különböző fajta (esetleg különböző verziójú) szoftverek szerinti részekre bontást. Mivel a hardver és szoftver-eszközök aránylag sűrűn változnak, ez minden ilyen változáskor a rendszer újratagolását váltja (váltaná) ki.

A szervezeti elv szerinti rendszerfelosztásnak legalább értékelhető érvei vannak, de a szervezetek gyakori átszervezése (a jelentkező problémák egy részét először átszervezéssel próbálják megoldani) ennek a tagolási elvnek a hiányosságait és hátrányait hangsúlyozza.

A két utolsó tagolási elvet tartják ma a legmegfelelőbb szempontnak a rendszer felosztására. A funkció fogalma alatt itt adott feladat elvégzéséhez szükséges aktivitássorozatot (tevékenységsort) kell érteni. Mivel, azonban a funkciók változékonyabbak az üzleti élet (valós rendszer) fogalmaitól (adataitól) – az események és tevékenységek mindenképp adatfüggők – így a rendszertagolást legcélszerűbb az ismereti elv szerint végrehajtani.

A valós rendszer (az üzleti élet egy behatárolt területe) megismerése lépésenként (iteratív módon) történik az előzőleg felismert és dokumentált tényezők folyamatos finomítása közepette. Ezt a tényt tartja tiszteletben a rendszeranalízis is a valós világ mind pontosabb megismerése érdekében, de a tervezési fázist is az iterativitás jellemzi: a tervezés során több tervváltozat készül, amelyeket elemzéssel és tervezéssel egybekötött lépéssorozatok folyamán javítanak fel, hogy a végén a legjobb megoldást fogadják el rendszertervként.

6.2.1.4. Felhasználó-orientáltság

A fejlesztés folyamán sohasem szabad megfeledkezni arról a mozzanatról, hogy az információs rendszer a megrendelőnek/felhasználónak készül, tehát az ő igényeit a lehetőség szerint ki kell elégíteni. A fejlesztés ideje alatt fontos a szoros együttműködés megvalósítása a fejlesztőcsoport és a megrendelő(k)/felhasználó(k) között. Ezt a gyakori közös munkaértekezletekkel lehet megvalósítani, ahol a fejlesztők ismertetik a fejlesztés menetét, és meghallgatják a megrendelő(k)/felhasználó(k) véleményét, megjegyzéseit és elvárásait.

A felhasználó nem tudja megfogalmazni az igényeit az információs rendszerek összes tényezőivel kapcsolatban, mivel nem ismeri valamennyit. Ezért változatok kidolgozásával segíteni kell őt elvárásai megfogalmazásában. A kialakított információs rendszer, és az azt megvalósító szoftver felhasználó-orientáltsága ma már alapkövetelménynek számít (könnyen áttekinthető, egyszerűen kezelhető, súgóval ellátott, jól dokumentált és toleráns – legyen türelmes a hibázó felhasználókkal, és tegye lehetővé a hibáik javítását) a megbízhatóság és hibátlan működés mellett.

A fejlesztőcsoport tagjainak azonban nem kell feltétel (és főleg gondolkodás) nélkül minden felhasználói igényt kielégíteniük. A szakma becsületét nem szabad Csáki szalmájaként őrizetlenül hagyni. A megrendelő szeszélyes igényeit, amelyek az információs rendszer működésének minőségi romlásához vezetnének nem ajánlatos teljesíteni. A felhasználók gyakran egymásnak ellentmondó elvárásokkal állnak elő, így a fejlesztés folyamán fogant döntések feladata a tervezett rendszer szempontjából legjobbnak ítélt kompromisszum kialakítása, megtalálása.

6.2.1.5. Modellezés, absztrakció, modularizálás

Az információs rendszerek tervezését, a fejlesztés (még pontosabban a kialakítás) fontos fázisát ma gyakran nevezik modellezésnek. Mert az most modern: modellezni a tervezés helyett. Ezért ma mindenki inkább modellezzik, mert az jobban hangzik. De akkor legalább a modelleknek nevezett rendszertervek jobbak lennének a régieknél...

Az információs rendszerek három tényezője (az adat, az esemény és a tevékenység) elvonatkoztatás, absztrakció. Mi több, ezeket a tényezőket az információs rendszerek tervezésekor a három vetület egyikében el kell helyezni és három szinten megfogalmazni. A dokumentációban pedig mind az absztrakciós szintek közötti leképezéseket, mind a leképezések okait indokolni kell. Az elvonatkoztatás (absztrakció) a rendszerelemző azon képességét veszi igénybe, melynek segítségével ki tudja szűrni valamely jelenség általános jellemzőjét a rendelkezésre álló sajátságos, konkrét jellemzők alapján¹⁰. A tervezés nélkülözhetetlen kelléke az absztrakció, hasonlóan, mint az előbbi projektek folyamán bevált, jó megoldások (sablonok) újfenti alkalmazása. A jó informatikus megoldás-változatokat (prototípusokat) dolgoz ki, összehasonlítja, mérlegeli és értékeli őket.

¹⁰ In pluribus unum.

Mai értelmezés szerint modellnek csak azok az információs rendszerprojektek tekinthetők, amelyek megfelelnek az előzőleg kialakított minőségi mércéknek (kritériumoknak). A modellezés tehát olyan folyamat, amely a legnagyobb mértékben megfelel az előre kiszabott minőségi kritériumoknak. A modellezés folyamatában tehát kikerülhetetlenül találkozunk a tervezés az elemzéssel, mivel a kidolgozott megoldás-változatokat össze kell vetni a minőségi előírásokkal, és szükség szerint újabb iteratív (finomító) lépést kell elindítani a vizsgált változat minőségének javítása érdekében. Automatikusan nem tekinthető minden projekt modellnek (ha nem felel meg a minőségi mércéknek), ugyanakkor, természetesen minden modell projekt is egyben. A minőségi kritériumok leírása a 6.3.4. pontban található.

A modularizálás fogalma alatt a rendszer olyan, kisebb részekre (modulokra) való tagolását értjük, amelynél a modulok közötti kapcsolatok is megfogalmazottak a modulok egy későbbi, többszörös alkalmazásának érdekében.

Komponenseknek az olyan, moduloktól nagyobb egységeket (részeket) nevezzük, amelyek meghatározott tevékenységsort ölelnek fel egy funkción belül, esetleg teljes funkciót valósítanak meg. Egyre terjed a komponensalapú rendszerfejlesztés, bár a szabványok hiánya komoly problémákat gördít a fejlődés útjába. Az különböző komponens-fejlesztőeszközök által létrehozott termékek (komponensek) más fejlesztőeszközökön nem használhatók (ezt hívják szoftverkompatibilitási problémának). A fejlesztési és használati környezet (főleg operációs rendszer) különbözősége ugyancsak gátat vet az ilyen típusú fejlesztésnek. Ezeket a különbségeket igyekeznek eltörölni (vagy legalábbis minimalizálni) a Microsoft az új .Net Framework filozófiájával. Erről az elképzelésről (a közös API – Application Programming Interface – ról, ASP+-ról és COM+ technológiáról) az előző fejezetben volt szó bővebben. Az ellenőrzött komponensek többszörös alkalmazása mindenesetre jelentősen meggyorsíthatná az információs rendszerek kialakítását. A komponensalapú fejlesztésről szóló irodalom szerteágazó: [Raffai – 22], [Tumbas] és a Microsoft Web-oldala, www.microsoft.com.

6.2.1.6. Szabványosítás, dokumentáció és minőségellenőrzés

Az újra (vagy többszörös) használhatóság és a hordozhatóság elérése céljából szükségszerű az erre a célra szabványosított eszközök és technológiák használat. Ez elsősorban a hardverelemekre, a hálózati szabványokra (protokollokra), a kommunikációra és a rendszerszoftverre (operációs rendszer-környezetre) vonatkozik. Amint ez a fenti felsorolásból

látszik az ajánlások elsősorban az információs rendszer környezetvetületére, és a rendszer megvalósítására (leprogramozására) vonatkoznak.

A szabvány szerepét és jelentőségét már az előzőekben megismertük. Itt csak a dokumentálással kapcsolatos szabvány-ajánlásokat emelnénk ki, ezekről ugyanis eddig nem volt szó. Az információs rendszer tervének írott formájú megvalósítását általában dokumentációnak hívják. A dokumentáció szöveges megfogalmazásokat, listákat, diagramokat, rajzokat, stb. felölelő rendszerleírás jelent. A rendszerterv (rendszerdokumentáció) az információs rendszer végleges alakját (a tényezők ajánlott változataiból az optimálisakat) tartalmazza, de csak gazdagítja a tervet a többi tényező-változat hozzácsatolása, és az optimális tényező kiválasztás megindoklása. A rendszerdokumentáció nem a tervezők magánügye és magántulajdona, így azt a legyártott szoftvertermékkel együtt kötelesek leadni a megrendelőnek. A dokumentáció szerepe sokrétű:

1. a fejlesztőcsoport tagjai a dokumentáció segítségével kommunikálnak egymással,
2. a dokumentáció alapján a projektvezető megbecsülheti a fejlesztőcsoport tagjainak eredményességét,
3. új felhasználók (és fejlesztőcsoport-tagok) a dokumentáció segítségével ismerkedhetnek meg a rendszerrel, de a megrendelővel való érdekegyeztetések alapjául is szolgálhat,
4. a közös etalon szerepét is be kell töltsé,
5. a megoldás újraelemzésének és a változó természetű elemzés alapját is képezheti,
6. a rendszerfejlesztéshez szükséges erőforrások ellenőrzésének (ujrabecslésének) lehetőségét is biztosítja.

A régi rendszerfejlesztési módszerek tervezési és specifikálási fázist különböztettek meg. A tervezési fázis, eredményként a rendszerdokumentációt biztosította, amelyben az információs rendszer általános tényezői kaptak helyet, míg a specifikáció azokat a részleteket tartalmazta, amelyek az információs rendszer megvalósításához voltak nélkülözhetetlenek: a tervező/elemező és a programtervező/kódoló papírra rögzített kommunikációját (leírásait, okfejtéseit, magyarázatait, jegyzeteit, kérdéseit, tesztadatait, stb.). Ma már nem használják ezeket a fogalmakat, ma már nem globális/fontos és részletes/kevésbé fontos tényezőkre osztják az információs rendszer tényezőit. Helyettük három fogalom körül “forog a világ”: a feladat, a megoldás és az eszköz (vagyis a fogalmi/konceptuális, logikai és fizikai szint) körül. Manapság az információs rendszert nem dokumentálják és specifikálják, hanem terveket (projekteket), modelleket dolgoznak ki rájuk, mégpedig az adat- a feldolgozás- és környezetvetületre. A

környezetvetületet természetéből adódóan (csak fizikai szintje van) csak egy terv írja le, a másik két vetületre azonban három terv készül: fogalmi, logikai és fizikai terv.

A dokumentáció nagy erenye lehet az, ha a megoldások mellett a megoldások indoklásait is tartalmazza (miért éppen olyan a megoldás, amilyen, vagy mi alapján, milyen szempontok szerint történt a választás a változatok közül). Az adat- és feldolgozásvetület három absztrakciós szintje között leképzések jelentik a kapcsolatot. A jó rendszerdokumentáció raktározza a leképzések okait is, nemcsak magukat a különböző absztrakciós szintű terveket.

A dokumentáció tartalmazhatja az adott fejlesztéssel kapcsolatos ismereteket (adatokat) is, amelyeket a fejlesztési tevékenységgel kapcsolatos események keltenek életre, de természetesen a dokumentációnak (az átmenetinek, de a végsőnek is) felhasználói is vannak szép számban, akik maguk is hozzájárulnak (felelősek) a dokumentáció létrejöttéhez valamilyen hardver/szoftver eszközön való “közreműködésükkel”. Újraolvasva az előző mondatot arra kell rájönnünk, hogy a dokumentáció ugyancsak információs rendszer, amely magáról a fejlesztési fázisban “vergődő” információs rendszerről raktároz adatokat (ismereteket tartalmaz az ismeretekről), de az adatkezelési eljárásokat már nem tartalmazza, ezért gyakran metaadatbázisnak nevezik.

A jó dokumentáció biztosítja az információs rendszer minősítéséhez szükséges “nyersanyagot”. A rendszerfejlesztés mérnöki megközelítése azonban azt követeli, hogy az információs rendszert más ipari termékekhez hasonlóan részekre, szerelvényekre bontottan alakítsa ki, szemlélje és vizsgálja. A kívánt végtermék tulajdonságai alapján közelítenek az “alkatrészek” felé, melyeknek szabványosított az előállítása (kialakítása): előkészítés, kidolgozás (termelés) és minőségellenőrzés. Ezt a megközelítési módszert alkalmazzák az információs rendszerek fejlesztésénél is: a fejlesztési folyamat fázisokra és lépésekre oszlik. Pontosan behatárolt a fejlesztési fázisok feladatköre és kapcsolódásuk az előző tevékenységekhez (lépésekhez és fázisokhoz, amelyek időben megelőzik és feltételezik a fejlesztés folytathatóságát). Ismertek a fázisok és lépések végtermékeire vonatkozó minőségi elvárások is, amelyeket a lépések és fázisok zárótevékenységeként kötelező ellenőrizni. Az információs rendszer tervezésének folyamán általában két fajta tesztelést végeznek: a fázistesztet és a rendszertesztet.

Mint tudjuk a fejlesztés fogalma szélesebb a tervezés fogalmánál, mert a tervezés mellett felöleli az információs rendszer megvalósítását is. A kifejlesztett szoftvertermékek külön tesztelés tárgyát képezik. A szoftvereket két szempontból kell tesztelni (vizsgálni), ezért két fajta vizsgálati tevékenységet különböztetünk meg:

1. verifikáció (verification) és
2. validitásvizsgálat (validation).

A verifikáció során a tesztelő meggyőződik a szoftver működésének helyességéről (dolgozik-e minden „ág” a szoftverben, és azt csinálja-e a számítógép, amit az utasítások megfogalmaznak), hibamentességéről, „lefagyásmentességéről”. A validitásvizsgálat azt vizsgálja, vajon a szoftver csakugyan azt csinálja-e, ami a feladata (azt, mindazt és csakis azt, ami a feladata). A rendszerszintű validitásvizsgálat a felhasználói elvárások kielégítésének mértékét becsüli meg. A verifikációs teszt számára már a tervezés idején készítenek tesztadatokat a rendszer működése szempontjából kritikusnak tekinthető pontok ellenőrzésére. Nem létezik egyetlen olyan bonyolultabb szoftvertermék sem, amelyiket teljesen (százszázalékosan) letesztelték volna.

Mivel a tesztelés tevékenységeköre a verifikációval kezdődik, és a validitásvizsgálattal folytatódik (fejeződik be), lássuk a verifikációs teszt két végrehajtási szintjét:

1. modulteszt (unit teszt) és
2. integrációs teszt.

A modulteszt az egyes programmodulok önálló működését (a működés helyességét) ellenőrzi a következő négy, jól elkülöníthető (bár szorosan összetartozó) működési területen:

- a. algoritmus,
- b. művelet-, kifejezés- és függvényhasználat,
- c. adatkezelés és
- d. kapcsolatbiztosítás (interfész, interface).

Az integrációs teszt a modulteszten túljutott programkomponensek együttműködésének helyességét ellenőrzi. Az együttműködési hibák a kapcsolatbiztosítási modulrészek összehangolatlanságából eredhetnek (nem, vagy nem jól történik meg a paraméterátadás). A paraméterek kétfélék lehetnek: vezérlőjelek és adatok. Ha a hívó és hívott modul eltérően értelmezi a paramétereket (akár csak egyet is, legyen az vezérlőjel vagy adat), fennáll a hibás működés lehetősége. Emiatt az integrációs tesztet mindkét irányból ajánlatos elvégezni, vagyis megkülönböztetünk:

- a. top – down integrációt és
- b. bottom – up integrációt.

A validitásvizsgálat nem a modulokra vagy végrehajtható egységekre vonatkozó ellenőrzés, hanem a szoftvert szerves egésznek tekintve vizsgálja, hogy vajon megfelel-e a felhasználó elvárásainak (pontosabban a rendszertervben/rendszerdokumentációban megfogalmazott felhasználói követelményeknek). A validitásvizsgálat leggyakrabban “élő adatok” segítségével történik. Két fajtáját különböztetjük meg:

1. alfa teszt és
2. béta teszt.

Az alfa teszt a fejlesztőcsoport (vagy annak szoftverházán belül kialakított team) által végrehajtott validitási vizsgálat, a béta tesztet viszont a fejlesztett szoftver potenciális jövőbeni felhasználói végzik el, saját hardver-szoftver-környezetben, a fejlesztőcsoporttól teljesen függetlenül.

6.3. AZ INFORMÁCIÓS RENDSZEREK FEJLESZTÉSÉNEK MÓDSZERTANA

Az információs rendszer nagyszámú és eltérő jellegű tényező együttese. A tényezők jellegének nagyfokú eltérése, és a köztük fennálló összetett kapcsolatháló az ilyen rendszerek fejlesztéséhez konkrét módszer alkalmazását igényli. Az információs rendszer túlságosan bonyolult ahhoz, hogy konkrét módszer nélkül megvalósuljon (létrejöjjön).

Az információs rendszerek fejlesztési módszerei a technikák és eljárások szervezett együttesét biztosítják a rendszer programozott (algoritmusszerű) megoldásához. Mivel a fejlesztési módszerek szervezett tényezőegyüttest (technikák és eljárások szervezett együttesét) jelentenek, ők maguk is rendszereknek tekinthetők. A módszertan az egyes (konkrét) módszerek elemzésével és felépítésével foglalkozik, vagyis magasabb absztrakciós szintű fogalom a módszer fogalmánál.

A fejlesztési folyamatban annak tárgyától, és az alkalmazott módszertől függetlenül két elválaszthatatlan szál mentén folyik a fejlesztés: az egyik szál a tartalmi (tárgy szerinti, és a mit kérdésre gombolyodik ki), a másik az irányítási (vezérlési, és a hogyan kérdésre halad előre). A konkrét módszerekben szinte sohasem választják őket ketté

Az információs rendszerek fejlesztési módszerei (mint rendszerek) a következő módszertani kellékekkel kell, hogy ékeskedjenek:

1. alapkoncepció,
2. struktúra és terminológia,
3. megfogalmazási mód,
4. kritériumrendszer,
5. tervezési algoritmus,
6. dokumentáció-kezelés és
7. technikai támogatás.

6.3.1. Alapkoncepció

A koncepció egy alaposan átgondolt problémamegoldítási hozzáállás, letisztult szemléletmód. Az alapkoncepció a fejlesztés tárgyához igazodó, letisztult, megalapozott, egységes elképzelések szerinti általános szemléletmód. Az információs rendszerek alapkoncepciójának megfogalmazásakor az első és legfontosabb (talán egyetlen lényeges) kérdése a következő: mi is az információs rendszer. Az információs rendszer fogalma a negyedik fejezetben ismerhető meg, ahol megtudhatjuk, hogy az információs rendszer nem más, mint a tényezőinek, és az azok közötti kapcsolatok szervezett együttese. Az információs rendszer tényezői közé pedig a következők tartoznak: az adatok, a velük kapcsolatos információs események, az adatokon végrehajtott információs tevékenységek, az előző három tényezőhöz és az információs rendszer működtetéséhez szükséges erőforrások, a felhasználók, és az előző tényezőket szabályozó, rendszerező szabványok és előírások.

Az alapkoncepció a tevékenység tárgyának értelmezése, és ezért olyan fontos a fenti meghatározásban az egységes jelző: a fejlesztőcsoport minden tagjának (de méginkább a fejlesztés minden résztvevőjének) el kell fogadnia (a közös, egységes) alapkonceptiót. Mert képzeljük csak el, mi lenne akkor, ha egyesek az információs rendszert programhalmaznak, mások jelentéshalmaznak, ismét mások pedig hardver és szoftver segítségével támogatott döntéshozatali segédeszköznek tekintenék...

6.3.2. Struktúra és terminológia

Az információs rendszer inhomogén tényezőegyüttes szervezett egysége: adat-, feldolgozás- és környezetvetületből (ahova a felhasználók és az információs rendszerek fejlesztéséhez és működéséhez szükséges erőforrások – idő, pénz és technikai eszközök – tartoznak) épül fel. Az adat- és feldolgozásvetületnek három absztrakció szintje van: fogalmi/konceptuális, logikai és fizikai szint. A tényezők közötti kapcsolatok szinte átláthatatlanul bonyolultak: adott vetület ugyanazon szintjén elhelyezkedő tényezők is kapcsolódnak egymáshoz, a vetületek szintjei között szükségszerű a kapcsolat, hiszen leképzés alapján származnak az alacsonyabb absztrakciós szintek a magasabbakból, de a különböző vetületek azonos absztrakciós szintjei között is kimutatható a kapcsolat. Az adatok egymással is (logikai) összefüggésben állnak, mint ahogy a feldolgozást alkotó tevékenységek is (a

feldolgozási logika szerint megelőző és követő tevékenységek kapcsolódnak a szemlélt tevékenységhez. Az adatok a feldolgozásokhoz, egyik is, másik is a felhasználóhoz kapcsolódik.

Az előző bekezdés arról árulkodik, hogy a információs rendszer struktúrája (belső felépítése) nagyon bonyolult hálónak felel meg.

A rendszerstruktúra a rendszer tényezőinek és azok kapcsolatainak a letisztult, megalapozott és egységes elképzelések szerinti általános szemléletmódja. Minden konkrét információs rendszernek van rendszerterve, amely annak konkrét tényezőit, és azoknak konkrét összefüggéseit (vagyis struktúráját) is tartalmazza. Ezt a konkrét tervet (ha még egyéb, más, a későbbiekben ismertetett rendszerkritériumokak is megfelel) nevezzük az információs rendszer modelljének (specifikus minta). Az információs rendszer struktúrájáról alkotott absztrakt képet, vagyis a specifikus minta absztrakt nézetét a modelllek modelljének: metamodellnek nevezik.

Az információs rendszerek tényezőinek és a közöttük lévő kapcsolatoknak konkrét fogalmakat, elnevezéseket kell adni. Ezek az elnevezések, és a mögöttük húzódó fogalom-tartalom képezik, jelentik a terminológiát. A fejlesztőcsoport tagjai csak akkor tudják kicserélni ismereteiket egymással, és közösen gyúrni, javítani a változat-ajánlatokat, ha egységes terminológiát használnak. Mivel az információs rendszer tényezői három absztrakciós szintbe tartoznak, az őket leíró elnevezések gyűjteménye, halmaza, vagyis a terminológia is strukturált.

6.3.3. Megfogalmazási mód

Az információs rendszer tervezése (a rendszerterv) maga is információs rendszernek tekinthető a negyedik fejezetben leírt információs rendszer definíciója alapján. A rendszerterv adatokat tartalmaz (a fejlesztett információs rendszerről), a fejlesztés folyamán olyan események is jelentkeznek, amelyek tervezési tevékenységeket váltanak ki, a rendszerterv kidolgozásához nélkülözhetetlenek a megfelelő erőforrások, és a fejlesztés folyamata egységesített (szabványosított) ütemben és formában bonyolódhat. Ha az előzőekben az információs rendszer absztrakt képét metamodellnek neveztük, akkor logikus, hogy az annak tervével kapcsolatos tényezők szervezett együttesét metainformációs rendszernek hívjuk.

A megfogalmazási mód fogalma alatt az információs rendszertervé rögzítésének, leírásának, közlésének, bemutatásának formáját értjük. Az eddigiek során három fajta megfogalmazási mód volt/van használatban:

1. tervezési nyelv alapú tervezés,
2. formaorientált tervezés és

3. ábraorientált tervezés.

Az első típusú megfogalmazási mód a formai (szintaktikai) és tartalmi (szemantikai) szabályokkal ellátott tervezési metanyelv alapján történő fejlesztést szorgalmazta, az akkor lendületes fejlődést vett programnyelvek sikerein felbuzdulva. Az angol nyelv szabályaira épülő metanyelv használata fárasztó, és (főleg a nem angol anyanyelvű fejlesztők körében) idegen volt, ezért a hetvenes évek közepén új megfogalmazási mód kezdett elterjedni. A formaorientált tervezésnél a megfogalmazási mód egyszerűsödött: a tervező a tervezést szabványosított tervezési nyomtatvány(sor)ok kitöltésével végezte (persze a megfogalmazási mód mögött mghúzódó terminológia megfelelő ismeretében). Ez a megfogalmazási mód meggyorsította a terv leírását (dokumentációját). Egy pillantás ezer szónál is többet ér...¹¹ A mérnöki tervezés CAD (Computer Aided Design) programjainak mintájára az információs rendszer tervezése is ábraorientálttá alakult.

Teljesen nyilvánvaló, hogy az ábrázolási mód is egy nyelv, magától a konkrét ábrázolási módtól, vagyis az ábrázolási mód konkrét formájától függetlenül. Mint minden más nyelvnek, neki is egységes szintaktikai és szemantikai szabályokat és terminológiát kell követnie.

A pont zárógondolataként hangsúlyozni kell, hogy az információs rendszerek fejlesztésénél legmegfelelőbb a megfogalmazási módok ötvözött, együttes használata a rendszer leírásához. A legfontosabb dolgokról (az alapkoncepcióról, a struktúráról és a terminológiáról), azonban, továbbra sem szabad megfélekedni („most a megfogalmazási móddal foglalkozom, más nem érdekel” alapon).

6.3.4. Kritériumrendszer

A fent említett gondolatot nem lehet elégszer hangsúlyozni: a fejlesztőcsoport a tervezés folyamán az alapkoncepció útmutatása alapján, az annak megfelelő struktúra és terminológia tiszteletben tartásával építi ki a rendszert. Ez azonban nem más, mint a „mit” és a „hogyan” megfogalmazása, és csupán a szükséges keretrendszert biztosítja. Egy jó tervezési módszernek rendelkeznie kell olyan kritériumokkal (kritériumrendszerrel), amelyekkel minősíteni lehet a kidolgozott tervet (még jobb, ha több tervváltozat is készül).

Az utóbbi zárójel mondatfoslánya megérdemel egy teljes gondolatot: ajánlatos lenne, ha a fejlesztés során több tervváltozat is készülne (legalább kettő-három), amelyek közül aztán választani lehetne, esetleg a változatok alapján egy újabb, „összekombinált” válhatna az

¹¹ a Slovenijales régen elfelejtett szállóigéje; eredetiben: jedan pogled vredi više od hiljadu reči...

optimálisan elfogadhatóvá (el kell ismernem, hogy a mai gyakorlatot figyelve ez idegennek, sőt eretnekségnek tűnik). Az információs rendszer terve, mint tudjuk, a valós rendszernek részben absztrakt leképzése, mert a fejlesztőcsoport intellektuális erőfeszítéseit is tartalmazza. A tervtermék (az információs rendszer terve és a megvalósított, működő információs rendszer) nem tekinthető az egyetlen lehetségesnek, hiszen még ugyanaz a fejlesztő sem tervezné meg kétszer ugyanúgy az információs rendszert, még ha teljesen azonos lenne is a feladat megfogalmazása.

A kritériumrendszer olyan objektív és egymással kapcsolatban álló minősítési mércék rendszere, amelyet a fejlesztői csoport tagjai alkalmaznak a rendszerterv, vagy tervváltozatok kiértékelésére és az optimális megoldás kialakítására.

A kritériumrendszer nagyszámú konkrét kritériumot is tartalmazhat (egyesek több száz kritériumot emlegetnek), amelyek tudatosan szervezettek, és egymással is kapcsolatban vannak, éppen ezért tekinthetők rendszernek. A sok konkrét kritérium közül mi csak öttel foglalkozunk.

6.3.4.1. Valósághűség

A valós (eredendőségi szempontból elsődleges) rendszer tényezőit és azok összefüggéseit a rendszerfejlesztés folyamán torzításmentesen kell leképezni az információs rendszer tényezőivé. Ezt mondja ki a valósághűség elve, illetve kritériuma.

Ezt a minőségi előírást (kritériumot) nem lehet formálisan ellenőrizni semmilyen automatizált eljárással, programmal. A valósághűség ellenőrzéséhez ugyanis megfigyelési, értelmezési, elemzési, absztrakciós, és ítéletmeghozási készségre van szükség, amely csak az emberi fajra jellemző (az eddigi ismereteink szerint).

A valósághűségi kritérium megsértése kétféle lehet: a túlságosan „elnagyolt”, vagy a túlon túl „elaprózott” tükrozés. A legmagasabb iskolai végzettségre az egyszeres adatbeviteli lehetőség (mit, mikor és hol végzett) biztosítása a valóság elferdítése (elnagyolt leképzése), mert a többdiplomások hazugságra kényszerülnek. Ennek az ellentéte a túlzott, és tegyük hozzá felesleges részletezés. Az új munkások nyilvántartása az előző munkahelyeikről is kimutatást vezet. Az egyszeres adatbevitel a fenti hibát eredményezi, a nem korlátozott pedig, bár elméletileg helyes, de gyakorlati szempontból helytelen, mert sok érdektelen adattal terheli az információs rendszert minden érdemleges haszon nélkül.

6.3.4.2. *Érthetőség*

Az érthetőségi mérce (kritérium) két szinten, absztrakt és konkrét szinten ellenőrizhető. Az absztrakt szinten ez az előírás a fejlesztési vagy tervezési fogalmak helyes használatát rögzíti. Konkrét szinten az érthetőség problémakörében ugyancsak a helytelenül használt kifejezések okoznak gondot.

Az érthetőségi kritériumnak megfelelni kívánó fejlesztők beszélő neveket használnak az információs rendszer tényezőinek elnevezésénél. Az elnevezések megválasztásánál kétféleképp sérthető meg az érthetőség kritériuma: túlzottan lakonikus, tömör névadással, vagy éppen ellenkezőleg, túlzó szószátyársággal, csacsogó névmeghatározásokkal.

6.3.4.3. *Egyértelműség*

Az egyértelműség mércéje szoros kapcsolatban van az érthetőség kritériumával, és azt írja elő, hogy ugyanarra a fogalomra mindig az eredetileg kiválasztott egyértelmű nevet használjuk. Ez más megfogalmazásban úgy hangzana, hogy kerüljük a homonimák és szinonimák használatát. No, de mit is jelent ez a két fogalom?

Szinonima jelenség akkor merül fel, ha ugyanannak a fogalomnak (lényegnek) két vagy többféle elnevezést adunk: a nyakleves és a pofon – szinonima. Homonima akkor lép fel, ha különböző lényegeknél (fogalmaknak) ugyanazt a nevet adjuk, pontosabban ha egy szó hallatán nem tudjuk eldönteni, hogy melyik fogalomról van szó: a király – homonima (nem tudni pontosan, hogy uralkodó szemlyről, vagy egyszerű helyesléről – elégedettség kifejezéséről van-e szó).

Az egyértelműség ellen vétkezni akkor is tudunk, ha vétkezünk az érthetőség kritériuma ellen: ha túl tömör, nem beszélő, de akkor is, ha túl hosszú, csacsogó neveket adunk az információs rendszer tényezőinek. A biztosítóintézetben különböző biztosítás- és gépkocsitípusok vannak. Ha csak „típus” nevet határozunk meg (kocsitípus és biztosítástípus helyett), homonima problémájával állunk szemben. A biztosítóintézetben élet-, vagyon-, jármű-stb. biztosítással kapcsolatos ügyeket intéznek. Ha a tervező „a biztosított személy személyi száma”, „a vagyon tulajdonosának személyi száma”, vagy „a jármű tulajdonosának személyi száma” elnevezéseket választja, szükségtelenül szinonimát idéz elő.

Az egyértelműség kritériuma matematikai és szemantikai eljárásokkal jól ellenőrizhető, viszont a homonima és szinonima kiszűréséhez a minimalitási és teljességi kritériumok alkalmazása is szükséges.

6.3.4.4. Minimalitás

A valós rendszer összetettsége határozza meg az információs rendszer bonyolultságát: a valós világ bonyolult szegletéről nem készülhet egyszerű információs rendszer. A minimalitás hajszolása nem lehet az egyedüli mérce: a fizikai valóság fontos tényezői nem hagyhatók figyelmen kívül az információs rendszerterv minimalizálása kedvéért.

Az értelmes minimalitás-elv alkalmazása az átfedések kiküszöbölését, vagyis a redundancia elkerülését jelenti a fejlesztés során. A tervezők idegenkednek, félnek a redundanciától, mert első közelítésben nem egyértelmű, hogy vajon az átfedések az egyértelműség, vagy a minimalitás kritériumának megsértése miatt jelentkeznek. Ha például az előzőekben már említett „típus” adatrév (tulajdonságtípusnév) több helyen is jelentkezik, nem egyértelmű, hogy az homonimáról vagy redundanciáról árulkodik-e.

Az információs rendszertervnek három vetülete (adat-, feldolgozás- és környezetvetület), és három absztrakciós szintje (fogalmi, logikai és fizikai szint) van. A legszigorúbb elvárások a minimalitási kritérium betartásával kapcsolatban a fogalmi/konceptuális szintre érvényesek: itt teljes átfedésmentesség az előírás. A logikai és fizikai szinteken már nem olyan kritikus a minimalításra való törekvés, sőt, az információs rendszert működtető szoftverek adatkezelő rendszere, és a hardver gyöngeségei rendszerint kényszerítő tényezőként jelentkeznek az átfedések tudatos bevezetésére. A tudatosan kezelt többszörösség egyébként a rendszerműködtetés finombehangolásának (tuning) az egyik legfontosabb eszköze.

A minimalitás kritériumának megsértése, az átfedések, a redundancia jelensége matematikai algoritmusok, eljárások segítségével ellenőrizhető. Mivel az átfedések homonimát is jelenthetnek, ami már formális módszerekkel nem ellenőrizhető, így a matematikai eljárás után emberi beavatkozásra van szükség a homonimák kikeresésére. A tisztán redundancia jellegű problémák ezután már könnyen orvosolhatók.

6.3.4.5. Teljesség

A teljesség mércéje sokkal összetettebb az előző kritériumnál, ezért automatizált ellenőrzése nem is lehetséges. Nem található (egyenlőre) olyan eszköz (rendszer), amely a valós világ (rendszer) elemeit (azok kapcsolatait, azoknak jellemzőit, az eseményeket, stb.) fel tudná ismerni, és az információs rendszerben való jelenlétüket ellenőrizni tudná (ezek már az alkotó gondolkodás, az absztrakció, az ítéletformálás ismérvei).

Amennyiben a rendszerterv nem teljes, úgy az nem az objektív valóság csorbításmentes (torzításmentes) képét türozi vissza fogalmi szinten. Ez egyben azt is jelenti, hogy a teljesség kritériumának a megsértése egyidőben egy másik, a valóságghűség kritériumának a megsértését is jelenti (legalábbis ami e kritérium alulteljesítését illeti). Ennek a kritériumnak a „túlteljesítése” sem kívánatos, mert olyankor ismétlések, átfedések jelentkeznek, amelyek két másik előírás elleni vétséget is jelentenek. Adott esetben a minimalitási és az egyértelműségi kritériumokra kell gondolni.

6.3.4.6. A kritériumok következetes, harmonikus alkalmazása

A fent ismertetett kritériumok csak ízelítőt adnak a minőségellenőrzési előírások világába, de már ebből is jól kivehető, hogy az egyes mércék erős kapcsolatban állnak másokkal. Az információs rendszer terve csak akkor lehet optimális, ha minden előre meghatározott, kiválasztott kritériumnak egyaránt megfelel. Először az egyértelműség, érthetőség és valóságghűség szerinti ellenőrzés történik, az alábbiaból az is kitűnik, hogy miért.

Az egyértelműség kerül legelőször górgcső alá. Ennél az ellenőrzésnél a homonimák és szinonimák felszámolása történik az érthetőségi kritérium figyelembe vételével és tiszteletben tartásával (ellenőrzésével), majd az érthetőségi mérce kerül a vizsgálódás középpontjába.

A szinonimák rejtett redundanciát képviselnek. A különböző elnevezések alkalmazása ugyanarra a fogalomra (szinonima-jelenség) arra a téves következtetésre vezethet, hogy a rendszerterv megfelel a minimalitási kritériumnak, ami valójában a felfedezetlen, de létező szinonimák miatt nem igaz. A szinonimák az információs rendszer adatainak tartalmi ellentmodásához, inkonzisztenciájához is vezethetnek. Ha nem ismert, hogy milyen elnevezések mely fogalmakra vonatkoznak (pl. ha a gépkocsitípus elnevezés ugyanazt a fogalmat képvislei mint a járműfajta), akkor a különböző elnevezésű adatok eltérő értékeket vehetnek fel akár a feltöltés, akár a használat folyamán (pl. ha ugyanannak a személygépkocsinak a gépkocsitípusa

Cordoba, a járműfajta adatának értéke pedig Seat, és az elnevezések ugyanannak a fogalomnak a szinonimái, akkor a rendszer inkonzisztens).

A homonima látszat-redundanciát okoz. Két vagy több különböző fogalomnak ugyanazon névvel való megjelölése arra enged következtetni, hogy a terv nem tesz eleget a minimalitási megkötésnek (kritériumnak), hiszen két vagy több azonos név jelenik meg a tervben. Ha a több azonos névből csak egyet hagyunk meg, és a többit eltávolítjuk, akkor a terv a teljesség elve ellen fog véteni. Itt is az egyértelműségi mérce ellenőrzésével kezdődik a járható út, amely aztán folytatódhat a minimalitás és a teljesség vizsgálatával.

Az előzőek alapján megfogalmazható egy kritériumellenőrzési sorrend, amely a helyes megoldáshoz vezethet, de tudni kell azt is, hogy az egyes kritérium-előírások egymásnak ellentétes elvárásokat szabnak meg, így az optimális terv kialakítása nem egy, hanem „többmenetes játszma”. Íme az ajánlott sorrend:

1. egyértelműség,
2. érthetőség,
3. valóságghűség
4. minimalitás
5. teljesség.

6.3.5. Tervezési algoritmus

Az információs rendszerek fejlesztésének folyamata és annak felépítése (fázisai, eljárásai, ütemezése, stb) a projektmenedzsment hatáskörébe tartozik. A fejlesztés folyamatának egyik fázisa a tervezési algoritmus.

Tervezési algoritmusnak azt a tevékenységcsoportot tekintjük, amelyben a tervezőcsoport az információs rendszer tényezőit megfogalmazza (kialakítja), és ezeket a tényezőket, a valós rendszernek megfelelően, egy előre meghatározott kritériumrendszer mércéi alapján összehangolja. Így aztán a tervezési algoritmus végterméke az információs rendszer modellje lesz.

A tervezési algoritmus lehet szigorúan előírt lépéseket tartalmazó merev szabályzat (algoritmus), ahogy a múlt század hetvenes éveiben képzeltek el a tervezést, de lehet teljesen flexibilis, rugalmas is, amilyennek a legtöbb mai automatizált tervezési eljárást megvalósítják. A manapság piacon kínált, számítógéppel támogatott fejlesztési segédletek tervezési algoritmusai annyira rugalmasak, hogy szinte már nem is tekinthetők algoritmusoknak (teljes szabadságot

biztosítanak a tervezőnek a tényezők kialakításában, a tényezők kialakításának sorrendjét sem korlátozzák, ami nem lenne baj, de a tényezők kapcsolatainak megfogalmazását sem kényszerítik ki – ami viszont már nagy hiányosság).

A BISAD (Business Information Systems Analysis and Design) fejlesztési módszer szigorúan kötött tervezési algoritmust látott elő a tervezési folyamatra, és a rendszerterv formájára is. A merev fejlesztési módszerek nem engedik kidomborodni a tervező kreativitását, ezért egyesek szívesen mellőzik az alkalmazásukat. Mások viszont, akiknél a merev fejlesztési módszereknek nincs mit gátolni, ma is sajnálják, hogy „kimentek a divatból”.

A teljesen ötlet- vagy ad hoc-szerű fejlesztési módszer (hozzaállás) a hódító az utóbbi időben, amit talán a próbálkozás és tévedés (trial and error) elnevezéssel határozhatnánk meg legpontosabban. Ez egy olyan igazi algoritmus nélküli fejlesztési módszer. A rendszertervet az első „ötletroham” alapján, minden komolyabb elemzés nélkül állítják össze, majd az alkalmazásgenerátorok segítségével nyomban elkészült programokat futtatják (próbálkozás), végül ellenőrzik a működésüket. Ha nem azt kapták amit vártak (tévedés), akkor módosító ötletekkel indítják el a következő (iterációs) lépést. A módszer magában véve nem elvetendő, csak nem az információs rendszerterv első változatának kialakításánál alkalmazva, hanem a már a kritériumrendszer szerint kialakított terv fizikai megvalósításának javítására. Ez utóbbi eljárást finombehangolásnak (tuning) nevezzük, és a jó megoldás tökéletesítését szolgálja.

A tervezési algoritmus második fajtája az előterv (előprojekt, pilot-project) ahol az elv megegyezik a próbálkozás/tévedés hozzáállással, csak a terv és a (fizikai) megvalósítás nem az egész információs rendszerre vonatkozik, hanem annak egy kiválasztott, jellemző, aránylag független rendszerrészére.

A prototípus-készítés ugyancsak a rendszer sajátosságos (jellemző) rendszerrészének letervezéséből és fizikai megvalósításából áll, de nem a próbálkozás/tévedés eljárással hanem a tényezők három vetületbe és három szintbe való elhelyezésével, a kapcsolataik feltárásával, a kritériumrendszer érvényesítésével, stb. Tehát a hozzáállás komoly, de csak a rendszer egy részére vonatkozik. Ezzel a tervezési algoritmussal akkor van baj, ha a tervezők megfeledkeznek az információs rendszer egységességéről, vagyis arról, hogy a rendszerterv nem a rendszerrészek terveinek halmaza. Fogalmi szinten mindenképpen az egész információs rendszert kell megfogni, és egy ellenőrzött rendszertervet kidolgozni rá. A logikai és fizikai szint megvalósítása részletekben is megvalósítható, ha a fogalmi szintű rendszerterv a megvalósítás alapja. A prototípus nem lehet végső megoldás (nem lehet a rendszerrészek prototípusaiból összeállítani az információs rendszert), de hasznos tapasztalatokat nyújthat a teljes rendszer kiépítéséhez.

Az elemzés és tervezés összefonódásáról már volt szó az előző részekben. Most a tervezési algoritmus szempontjából kell még egyszer hangsúlyoznunk ennek a fontosságát. A tervezési algoritmus három fontos tevékenység-csoportra épül: az elemzésre, a tervezésre és az iterációra. Az elemzési fázis után a tervezési fázisban már az elemzés és a (algoritmustól függetlenül különböző változatokat biztosító) tervezési folyamat átfedek egymást. A tervezés befejezése után a fejlesztőcsoport bemutatja és igazolja a kidolgozott terv helyességét, ami kikerülhetetlenül ismét csak elemzéshez vezet. Fejlesztés közben pedig a kritériumrendszer előírásaival való egyeztetés folyamán a javítások új iterációs ciklus(ok) indítását váltják ki, amelyekben változatok kidolgozása (tervezés) és azok értékelése valamint a legjobb változatok kiválasztása (elemzés) bonyolódik.

A tervezési folyamat kezdetén mindig a kiindulási pont dilemmája lebeg Damoklész kardjaként a fejlesztők feje felett: honnan kezdeni, mi legyen a „kályha” (ahonnan elindulhatnak). A rendszerfejlesztési módszerek fejlődése folyamán „kályhaként” a következők szerepeltek:

1. kimenetek (jelentések, output),
2. bemenetek (ürlapok, input),
3. feldolgozások (funkciók) és
4. adatok (adatmodell, adatbázis-terv).

A merev tervezési algoritmusok (természetükből fakadóan), szigorúan előírják, hogy mi legyen a „kályha”, a rugalmasak pedig szabad kezet, szabad választást biztosítanak az alkalmazónak. A hetvenes évek elején a kiindulási pontot a jelentések, a kimenet képezte, amiből aztán visszafejtették, az algoritmusok ismeretében, hogy milyen bemenetekre is van szükség. A szeszélyesen változtatható kimeneti igények hamarosan kiindulási-pont változást idéztek elő: a bemenetek váltak a „kályhává”, mivel minden tőlük függött. A hetvenes évek közepén jelent meg a strukturált programozás, és vele majdnem párhuzamosan a strukturált tervezés is. Ez a tervezési algoritmus az eseményekre és tevékenységekre, illetve a tevékenységek elemi szintig való bontására, egyszerűen a feldolgozásra épült. Itt tehát a feldolgozások képezték a fejlesztés kezdőpontját. A fekete doboz fogalom alkalmazása itt teljesedett ki először igazán. A strukturált rendszertev felső hierarchiaszintjein csak a funkciók lényege (volt) ismeretes, a tartalma nem. A fekete doboz feketesége (tartalma) az alacsonyabb hierarchiaszinteken világosodva vált fehérré (ismertté). A feldolgozásból való kiindulás a feldolgozásoknak a kimenetekhez hasonló változékonysága miatt csakhamar elvesztette egyeduralmát, mert szinte vele egyidejűleg elterjedt az adatmodellezés, az adatközpontú alkalmazásfejlesztés, másszóval az adatok váltak a „kályhává”.

Az információs rendszerrészek eltérő természete miatt ajánlatos mérlegelni az egyes rendszerrészek fejlesztéséhez leginkább elfogadható kiindulási pontokat, de fejlesztési algoritmusokat is. Néhány fejlesztési algoritmus többoldalú megközelítést tesz lehetővé, ennek megfelelően több kiindulási pontot lát elő a fejlesztés elkezdésére. A fejlesztési algoritmus nem az egyszeres megközelítést (kiindulási pontot) kell, hogy szorgalmazza, hanem az objektív (a valós) világ jó tükrözését a jó struktúra kialakításának lehetőségeivel, és a kritériumok elemzésének előrelátásával (biztosításával).

6.3.6. Dokumentációkezelés

Az információs rendszerek fejlesztése folyamán többféle dokumentáció is készül. Ha csak az információs rendszer tervdokumentációjára szorítkozunk, akkor is többfajta dokumentációval kell számolnunk. Az elfogadott rendszerterv mellett a tervváltozatokra is készül dokumentáció.

Az információs rendszer dokumentációja lényeges fontosságú az információs rendszer fejlesztése és működtetése szempontjából, tehát nemcsak a felhasználónak, de a tervezőcsoportnak is nélkülözhetetlen, mert:

1. a tervezőcsoport tagjai a fejlesztés folyamán a dokumentáció segítségével egyeztetik a már elkészült megoldásokat és megvalósítandó feladatokat (a dokumentáción keresztül kommunikálnak egymással),
2. a fejlesztésbe később bekapcsolódó munkatársak és felhasználók a dokumentáció segítségével ismerkedhetnek meg a fejlesztésben álló rendszerrel,
3. a dokumentációt szabványként kell kezelni és tisztelni az információs rendszer tényezői, azok összefüggései valamint az alkalmazott fogalmak és megfogalmazott eljárások szempontjából, és
4. a dokumentáció a rendszerterv kritériumrendszer szerinti ellenőrzését, és utólagos, a rendszer javítását célzó elemzéseket teszi lehetővé.

A kritériumrendszer egymással ellentétes kritériumokat is tartalmaz (erről már volt szó az előzőekben). A kritériumok egyszer megállapított súlyozása (fontossága) nem örökérvényű modell. Az egyes kritériumok fontosságának megváltoztatása (akár a megrendelő, akár a tervezőcsoport tagjainak ajánlatára) a terv minőségének újraértékelését vonja maga után, ami csak a dokumentáció segítségével végezhető el.

Sokan vallják, hogy a dokumentáció a megoldások (és ajánlott változatok) mellett a megoldásba kerülő változat választásának indokait is tartalmaznia kell. A dokumentáció

ugyancsak rendszernek tekinthető, mert az információs rendszerterv adatait tartalmazza, a fejlesztés folyamán (az információs rendszer vagy rendszerterv adataira vonatkozó) események váltják ki a szabványosított (vagy szabványosítható) fejlesztési tevékenységeket (a dokumentáció kidolgozásának lépéseit), amelyeket megfelelő erőforrások (idő, pénz, technikai eszközök) biztosításával lehet csak a tervezők és felhasználók együttes bevonásával elvégezni. Mivel a dokumentáció az ismeretekről (információs rendszerről) tartalmaz ismereteket metaadatbázisnak is hívják (a fejlesztői adatbázist is nevezik metaadatbázisnak, és itt semmilyen ellentmondás nincs, hiszen a fejlesztői adatbázis a fejlesztés minden lépését rögzíti – ki, mikor, melyik információs rendszertényezőt, mennyi ideig stb. tervezte).

A dokumentációkezelés az információs rendszerek adatainak szervezett és hatékony kezelését jelenti, pontosabban a dokumentációnak, mint rendszernek a tényezőit gondozza szervezeten és hatékonyan. A dokumentáció a valós világ (rendszer – ez esetben az információs rendszer) hű tükrözését hivatott megvalósítani: az információs rendszer alapkonceptióját, struktúráját, az elfogadott megfogalmazási módo(ka)t, az elvárasi kritériumoknak megfelelő változatokat, összhangban az alkalmazott rendszerfejlesztési módszerrel.

A dokumentáció, bármennyire furcsának is tűnik, a rendszerterv tartalmától függetlenül saját, önálló életet is él. Erre azok az „apró”, menet (működtetés) közbeni változtatások szolgálnak példaként, amelyek sohasem találják meg a dokumentációhoz vezető utat, vagy épp ellenkezőleg, azok a tervezők által javasolt, és a dokumentáción keresztülvezetett javítások, amelyek sohasem „kelnek életre”.

A dokumentációkezelés egyik legfontosabb feladata az volna, hogy biztosítsa a megtervezett, megvalósított és változás következtében módosított információs rendszertényezőknek a dokumentációban és a működésben való kölcsönös, élethű leképzésének megjelenését.

6.3.7. Technikai támogatás

Az informatika és számítástechnika mai fejlettségi fokán már természetes az az elvárás, hogy az információs rendszerek fejlesztése számítógéppel történjen, pontosabban, hogy a rendszerfejlesztéssel kapcsolatos feladatok elvégzését számítógépen futó alkalmazások segítsék.

A fejlesztést támogató szoftverek közös néven CASE (Computer Aided Systems Engineering) eszközök néven ismertek. Nagyszámú tervezést és fejlesztést segítő szoftvertermék található a piacon, amelyek közül a legtöbbjük csak a fejlesztés bizonyos fázisait (és nem a teljes

fejlesztést) segíti. El kell ismerni, hogy ezek a CASE szoftverek nagyon szimpatikus, általában Windows operációs rendszerre írt, grafikus környezetet megvalósító (tetszetős menürendszerrel és ikonhalmazzal elkápráztató), gyors működésű (amihez természetesen erős számítógépre van szükség) segédeszközök. Ezeknek a szoftvereknek a mai értékelése nem eszköz-, inkább módszerközpontú. Inkább azt kell tehát megvizsgálni, vajon milyen a CASE-zel megvalósított információs rendszer alapkonceptiója, struktúrája (megengedi-e, megkülönbözteti-e a háromvetületű és hárszintű tervezést), terminológiája, megfogalmazási módja (megfelelő-e a szimbólumkészlete a valós rendszer tükrözésére), tervezési algoritmus, a vele létrehozható dokumentáció és a minőség ellenőrzésére biztosított kritériumrendszere.

Az információs rendszer teljes fejlesztési ciklusát lefedő, hasznos, jó CASE segédeszközök ma még nincsenek (figyelemreméltó próbálkozások viszont igen). Az okát talán az általunk biztosított technikai lehetőségekben kell keresni (amit szükségtelenül túllícitáltak a fejlesztő cégek): a tetszetős alakok, ikonok, a villogás eltereli a figyelmet a felhasználói szempontból sokkal fontosabb lényegről, a módszerről.

6.3.8. Metaszabványok

Az információs rendszer fogalmának megvilágításánál már volt szó a szabványokról, de ott, azok, az információs rendszerek tényezőire vonatkoztak (nem jelzőszerkezetű főnévről – metaszabványokról – hanem csak egyszerűen szabványokról volt szó). A szabványok az információs rendszer ötödik tényezőjét képviselték, és megkülönböztetésül a metaszabványoktól talán felhasználói szabványoknak nevezhetnénk őket.

A felhasználói adatokon, eseményeken, tevékenységeken, erőforrásokon, stb. kívül, mint tudjuk, a fejlesztésre vonatkozó, az előbbieken felsorolt tényezőkkel is számolnunk kell. Ezeket pedig a fejlesztési szabványok írják elő, amelyeket, mivel az ismeretekről szóló ismeretekre vonatkoznak, gyakran metaszabványoknak nevezzük.

A metaszabványokat általában nem sorolják a rendszerszervezési metodológia tényezői közé, tehát egyetlen fejlesztési módszer sem tartalmazza kitétel szeit a metaszabványokat, de az előzőekből látható, hogy a fejlesztés/tervezés minden tényezőjét átszövi.

Az információs rendszer fejlesztésére nincsenek általánosan elfogadott metaszabványok a fejlesztés tényezőire, ezért is uralkodik olyan nagy káosz a fejlesztés terén.

7. AZ INFORMÁCIÓS RENDSZER ADATVETÜLETE

Az információs endszerek fejlesztésének célja a különböző működési területen gazdálkodó cégek, esetleg személyek információ iránti szükségleteinek kielégítése. A cégek is embereket takarnak, akik a cég sikeres irányításának érdekében minél gyorsabban és egyszerűbben szeretnék a számukra fontos adatokhoz (információkhoz) jutni. Az emberek is, az adatok is az információs rendszer tényezői közé tartoznak, így egymással is és a többi tényezővel is összefüggésben állnak.

Egy adott információs rendszeren belül maguk az ismeretek is rendszert (adatvetületet) alkotnak. Az adatrendszer egymagában is igen bonyolult. Az ismerettől út vezet az adatig, melynek során az „ismeret mesterséges képet ölt, és elveszti természetes képét”¹. Az adatot ismét csak egy intellektuális erőfeszítést igénylő úton érleli minden homo sapiens külön-külön információvá. Az információvá érlelhető adat a kommunikáció nyersanyaga. Az adattá vált ismeret több szempontból is vizsgálható. A leggyakoribb vizsgálódás az adatok fizikai megjelenítésével foglalkozik, pedig az adatok logikailag is összefüggnek, és a valós rendszer tükrözésére szolgálnak.

7.1. AZ ISMERET

Az emberi faj folyamatosan adatokat gyűjt az őt körülvevő világról. A gyűjtés alapját az ember érzékszervei által biztosított ismeretek (adatok) jelentik. Nem cáfolható az a tény, hogy akár egy mosoly vagy kellemes illat, esetleg ízletes, ínycsiklandó étel is jelentős ismeretet, információt hordozhat, de az emberek közötti kommunikáció (ismeretátadás) legtöbbször a beszéd vagy az írott beszéd, a nyelv segítségével történik. Már tudjuk, hogy az adat olyan ismeret, amelyet értelmezni lehet, de még nem értelmezett. Az értelmezett adat pedig csak annak a személynek a számára válik információvá, akinek új ismeretet hordoz.

A természetes beszédben a kijelentés (állítás, tényközlés) legegyszerűbb formája az alanyt és állítmányt tartalmazó tömondat. A hiányos mondatok is kijelentésnek értelmezhetők, ha a hiányzó rész (alany vagy állítmány) az előző kijelentésekből (mondatokból) kiérthető. Az alany a kijelentés (közlés) tárgyát (miről akarunk valamit kijelenteni) jelöli, az állítmány pedig magát a közlést (mit akarunk kijelenteni az alannal megjelölt tárgyról) tartalmazza.

¹ Forrás: [11], 175. oldal.

Mindinkább felismerik és elismerik úgy az elméleti mint a gyakorlati szakértők, hogy a fogalmi tervezés fontos lépése az információs rendszer fejlesztésének. A fogalmi tervezés három legfontosabb fogalmának az egyedet, a tulajdonságot és a kapcsolatot tekintik. Ezek bemutatása következik.

7.2.AZ EGYED

A valós rendszernek (világ) az információs rendszer segítségével ismeretekkel leírni kívánt jelenségeit fogalmi szinten egyedeknek nevezzük. Még pontosabban: az egyed a valós világ adatokkal leírni kívánt jelenségeinek tükröződése a fogalmi absztrakciós szinten. Az egyed fogalma alatt sokan a pillanatnyi megfigyelés tárgyát értik, ami lehet személy, tárgy, esemény, természeti vagy társadalmi jelenség, esetleg elvont fogalom.

Az ember folyamatosan értékeli és valamilyen természetes vagy mesterséges ismerv, jellemző, illetve tulajdonság alapján csoportokba osztja a valós világ jelenségeit. A valós világnak az információs rendszer segítségével ismeretekkel leírni kívánt jelenségcsoportjait (osztályait) egyedtípusnak nevezzük. Az egyedtípusok tulajdonképpen a valós rendszer jelenségcsoportjainak fogalmi szintű képét, megfogalmazását képviselik.

A jelenségek csoportokba osztása vagy osztályozása nagy mértékben függ a szándéktól, illetve az információs rendszer célkitűzéseitől (célhierarchiájától). Az egyedtípus megfogalmazásának (elnevezésének) pillanatától, a konkrét egyedeket egyed-előfordulásoknak tekintjük. Az egyedtípusnak egy időben változó előfordulás-halmazát alkotják az egyed-előfordulások (a konkrét egyedek). Az egyedtípus elvontabb (magasabb absztrakciós szintű) fogalom az egyed fogalmánál.

Nézzük a fenti fogalmak gyakorlati hasznát egy szemléltető példán keresztül. Tételezzük fel, hogy Feriről és Zoliról szándékozunk ismereteket tárolni, akik egyetemi (főiskolai) hallgatók. Feri és Zoli jelentik számunkra az egyedeket, mert ők képezik megfigyelésünk tárgyát. Megalkotva a HALLGATÓ egyedtípust Feri és Zoli megszűnnek egyedeknek lenni, egyed-előfordulássá válnak: a HALLGATÓ egyedtípus előfordulásaiává.

7.3.A TULAJDONSÁG

A valós jelenségeket jellemzőik, tulajdonságaik segítségével írjuk le, a valós jelenségek osztályait ugyancsak. A kiválasztott jelenségeknek, az információs rendszer szempontjából lényeges jellemzőit fogalmi síkon tulajdonságoknak (tulajdonságtípusoknak, attribútumoknak) nevezzük. Az egyedtípus leírását, jellemzését tehát a tulajdonságtípusok valósítják meg. A tulajdonságtípus az egyedtípushoz hasonlóan elvonatkoztatás, absztrakció, hiszen az ember rendeli hozzá az elnevezést a fogalmakhoz. Az ugyanazon egyedtípushoz tartozó egyed-előfordulások rendelkeznek legalább egy olyan közös vonással (tulajdonságtípussal), amely alapján közös egyedtípusba kerültek.

Lássuk rögtön a példát! Legyen az (absztrakt) egyedtípus a SZEMÉLYGÉPKOCSI, ekkor a lehetséges tulajdonságtípusok a következők: rendszám, alvázszám, motorszám, gyártó_cég, gépkocsitípus, származási_hely (gyártó_ország), gépkocsi_színe, stb.

Minden tulajdonságtípusnak van egy időben változó értékhalmaza, amelyből a tulajdonságtípus előfordulásai kikerülnek. A tulajdonságértékek, vagyis a tulajdonságtípusok előfordulásai tehát értékhalmazt alkotnak, amit értéktartománynak, esetleg egyszerűen csak tartománynak (domain) is neveznek. Itt is a tulajdonságtípus az elvontabb, a tulajdonságérték pedig a konkrétabb fogalom.

Az egyedtípus, az egyed-előfordulás, a tulajdonságtípus és tulajdonságtípus-előfordulás (tulajdonságérték) a valós világ jelenségeit és annak fizikai jellemzőit fejezik ki a fogalmi szinten. Tudni kell, azonban, azt is, hogy az egyedtípus (egyed) és a tulajdonságtípus (tulajdonság) relatív fogalmak. Nem mindig egyértelmű, hogy a megfigyelt dolog micsoda: egyed-e vagy tulajdonság. A “szín” elnevezésű fogalomra azt modjuk, hogy tulajdonságtípus, pl. gépkocsi_színe, és ez így rendben is volna. Ha azonban az autógyárban, a festőrézszegben keverik ki a sajátos színeket különböző árnyalatokban, akkor már a színeket jellemezni kell az alap- és egyéb, forgalomban lévő színeknek a százalékban kifejezett jelenlétével, a szükséges oldószert mennyiségével, a felviteli eljárás megnevezésével, az egységnyi felületre vagy gépkocsitípushoz szükséges mennyiségre, stb., vagyis a szín ekkor egyedtípussá válik.

Azt a tulajdonságtípust vagy tulajdonságtípus-csoportot, amelyik egy egyedtípuson belül minden egyed-előfordulásra különböző értéket vesz fel azonosítónak, elsődleges kulcsnak vagy kulcsjelöltnek hívjuk. A valóságban ritkán található egy olyan természetes jellemvonása, tulajdonsága a jelenségeknek, amely különböző értékeket vesz (venne) fel minden egyed-előfordulásra. Ilyenkor egy új, mesterségesen kialakított tulajdonságtípust vezetünk be, amely az előfordulások mindegyikére más-más értéket vesz fel, vagy a tulajdonságtípusok olyan csoportját

választjuk ki, amelyik már betöltheti az azonosítás szerepét (különböző értéket vesz fel minden előfordulásra).

Az azonosításnak tehát két fajtája ismeretes:

1. nominatív és
2. deskriptív.

A nominatív azonosítás esetén kiválasztható egy olyan tulajdonságtípus, amelyik alkalmas az előfordulások megkülönböztetésére. Példa: a megrendelésszám egyértelműen azonosítja a megrendeléseket, ugyanúgy a személyi szám (JMBG) az embereket.

A deskriptív, leíró azonosítás alkalmával a jelenséget jellemző tulajdonságtípusok között válogatva olyan tulajdonságcsoporthoz kerül kiválasztásra, amely alkalmas a jelenségek egyedi azonosítására (megkülönböztetésére). A SZEMÉLY egyedtípus leíró azonosítója például a következő tulajdonságtípusokat tartalmazhatja: Vezeték- és keresztnév, születési dátum, születési hely, anyja neve, lakcím.

Ritkán, de előfordulhat, hogy egy egyedtípusban több olyan tulajdonságtípus (tulajdonságtípus-csoport) is található, amelyik azonosítani tudja az előfordulásokat. Ezeket kulcsjelölteknek nevezzük. A kulcsjelöltek közül egyet elsődleges kulcsnak jelölünk ki.

7.4.A KAPCSOLAT

A mindennapi beszédben egyre ritkábban használunk teljes kijelentéseket. Mondataink (kijelentéseink) általában hiányosak, az előző kijelentés(ek)re épülnek, vagyis kijelentéseinket összefűzzük, összekapcsoljuk. A jelenségekre vonatkozó ismeretek hármasságát (egyesek vetületnek emlegetik, de ez zavart okozhat az információs rendszer ugyancsak hármasság vetülete miatt) ma már mindenki elismeri:

1. extenzionális,
2. intenzionális és
3. transzverzális jelleg.

Az egyedtípus extenzionális jellegét (vetületét) az előforduláshalmaz elemei, az egyedi előfordulások képezik. Az egyedtípus extenziója az előforduláshalmaz elemeinek felsorolásával adható meg.

Az intenzió az egyedtípust jellemző fogalmak összessége, tulajdonságtípusainak halmaza. Az egyedtípus intenzióját tulajdonságtípusainak megadásával hozzuk létre.

A transzverzió az ismeretek, jelenségek (pontosabban az egyed-előfordulások és egyedtipusok) közötti kapcsolatot jelenti. Az előző mondat már sugallja a transzverzió kettős jellegét: két absztrakciós szinten fogalmazható meg.

Az egyedek illetve egyed-előfordulások közötti konkrét összefüggések, viszonyok fogalmi szintű tükröződését kapcsolatoknak hívjuk. Az egyedtipusok között fennálló összefüggések fogalmi tükröződését pedig kapcsolattípusoknak nevezzük.

A kapcsolattípusnak többféle jellegzetessége van, ezek közül csak a kardinalitással (kapcsolat fokával) ismerkedünk meg. A kapcsolattípus három fajtáját különböztetjük meg a kapcsolati fok alapján:

1. 1:1,
2. 1:N és
3. M:N.

Ugyanannál a biztosítótársaságnál egy gépkocsinak csak egy kötelező biztosítása lehet (gondolom nincs olyan együgyű, aki három társaságnál is kötne a kocsijára kötelező biztosítást). Ez a viszony 1:1 típusú, mert fordítva is igaz az érvényben lévő előírások szerint: egy kötelező biztosítás csak egy gépkocsira vonatkozik. Az 1:N fokú kapcsolat a gépkocsi és annak tulajdonosa között jelentkezik, mert egy kocsinak csak egy tulajdonosa lehet, viszont a (fizikai és jogi) személyek által birtokolt gépkocsik száma nem korlátozott. Az M:N fokú kapcsolatra a közlekedési balesetet vesszük példának, mert egy balesetben több gépkocsi is „érintett” lehet, de ugyanaz a gépkocsi több balesetet is „átélhet” (remélhetőleg nem egyidőben) a használata folyamán.²

Vannak olyan álláspontok is, hogy minden kapcsolattípust, a kapcsolat fokától függetlenül adatokkal kell ellátni. Emlékezzünk azonban arra, hogy az adatokkal leírni kívánt jelenséget vagy jelenségcsoportot hívjuk egyednek illetve egyedtipusnak. Ha tehát a kapcsolattípust ismeretekkel látjuk el, akkor az egyedtipussá, ún. kapcsoló-egyedtipussá válik. Jó iskolapélda erre az árut és a megrendelést összekötő megrendeléstétel kapcsoló egyedtipus.

Az angolszász nyelvterület indítványa, hogy a két lényeg összekötéséből eredő egyetlen fizikai kapcsolatot két névvel is kell illetni, mert a kapcsolat eltérő jellegű, ha az egyik, illetve másik lényeg irányából figyeljük. Mivel egyetlen kapcsoltaról van szó, a két elnevezés redundanciát okozna (szinonima, egyértelműségi kritérium), de más alapvető kritériumoknak sem felelne meg (valóságghűség, minimalitás).

Az M:N fokú kapcsolatot nem is tekintjük valódi kapcsolatnak, mert az egyedek valójában nincsenek egymással semmilyenféle viszonyban. Két egyed csak akkor van kapcsolatban

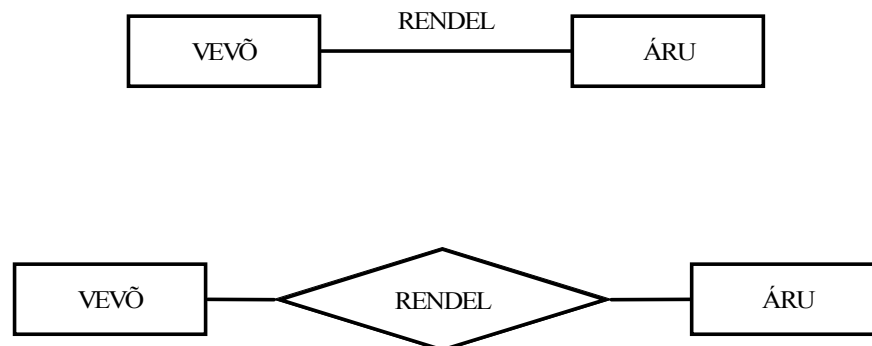
² Kölcsönvett példa. Forrás: [11.], 187. oldal.

egymással, ha az egyik egyed azonosítója (elsődleges kulcsa) megtalálható a másik egyed tulajdonságtípusai között. Az M:N fokú kapcsolat nem elfogadható volta az ilyen típusú kapcsolatok felbontását váltják ki: kapcsoló egyedtípus bevezetésével az M:N fokú kapcsolat két 1:N fokú kapcsolattá oldható fel.

A kapcsolat a legnehezebben felfogható fogalom az ismeretek világában, de érdemes megharcolni érte, mert ő a hatékony adatkezelés legfontosabb pillére.

7.5.AZ EGYEDTÍPUS KÜLSŐ SZERKEZETE

Az egyedtípusok külső szerkezetét az egyedtípusok, és a közöttük lévő kapcsolatok (kapcsolattípusok) alkotják. Az egyedtípusok külső szerkezetét gyakran hívják még logikai adatmodellnek (szinonima) is, nem téve különbséget a fogalmi és logikai absztrakciós szintek között. A logikai adatmodellt általában grafikus formában (megfogalmazási mód) dolgozzák ki. Az egyedtípus szimbóluma a téglalap, amelybe az egyedtípus neve kerül, a kapcsolattípusnak



7.1. ábra

pedig kétfajta jelölése is honos (7.1. ábra). A kapcsolatok alaposabb elemzése ahhoz a felismeréshez vezet el bennünket, hogy a kapcsolati viszony foka mellett a kapcsolat opcionálitása is fontos tényező. Az opcionálitás a kapcsolat mindkét oldaláról külön-külön elemzendő és megállapítandó tényező. Első közelítésben a kapcsolat csak opcionális vagy kötelező lehet az opcionálitását tekintve, de van finomított megközelítés is. Az opcionálitás az egyed és a hozzá tartozó konkrét kapcsolat létrehozásához vonatkozó megszorítás. Az opcionális kapcsolat esetében az egyed léte nem “követeli meg” a kapcsolat létezését, a kötelező opcionálitású kapcsolatnál pedig megköveteli. A RENDEL kapcsolattípus esetére az opcionálitás

minősége (értéke) a következő módon szűrhető le. Ha létezhet olyan konkrét vevő, aki még egyszer sem rendelt árut, akkor a kapcsolat felülről opcionális, ha nem, akkor felülről kötelező. Ha nyilvántartjuk az olyan árut is, amelyből még senki sem rendelt, akkor a kapcsolat alulról opcionális, ha nem, akkor alulról kötelező. A kapcsolatok foka és opcionalitása a kapcsolatok minimális és maximális kardinalitásával is kifejezhető, ahogy majd a következő pontban látni fogjuk.

7.5.1. A kardinalitás (kapcsolat-fok) szemantikája

Az inhomogén és homogén kapcsolatok általános tulajdonságainak pontosabb megismerése a következő, általános érvényű, kapcsolat-fokra vonatkozó kapcsolattípus-leírást eredményezte:

$$R(E_1(a_2, b_2):E_2(a_1, b_1))$$

Az előző kifejezésben az R a kapcsolattípus nevét jelöli, az E_1 és az E_2 az egyedeket (egyed típusokat) képviselik, az a_1 és a_2 a minimális kardinalitás-értéket, a b_1 és b_2 pedig a maximális kardinalitás-értéket ábrázolják.

Amennyiben csak általánosan kapcsolat-fokról beszélünk (mint a 7.4. pontban), rendszerint a maximális kardinalitás-értékekre gondolunk. A megismert kapcsolat-fokok (1:1, 1:N és M:N) alapján elemezzük végig a kapcsolattípusok összes megvalósítható fajtáit az opcionalitás értékeinek beleszövésével.

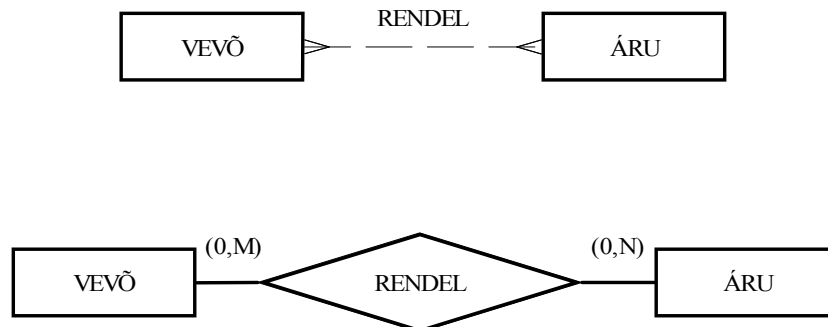
7.5.1.1. Az M:N típusú kapcsolat-fok szemantikája

Ennél a típusnál a maximális kardinalitás-értékek ismertek és rögzítettek: $b_1 = N$ i $b_2 = M$. A minimális kardinalitás-értékek tetszőleges értéket vehetnek fel. Négy alapállapot (értékállapot) kerül bemutatásra az alábbiakban, amely kimeríti a lehetséges kombinációk számát.

$$1. \quad a_1 = 0 \text{ és } a_2 = 0 \quad [R(E_1(0, M):E_2(0, N))]$$

A minimális kardinalitás-értékek azt mutatják, hogy mindkét egyed típus előforduláshalmazában lehetnek olyan egyed-előfordulások, amelyek egyetlen egyeddel sincsenek kapcsolatban a másik egyed típus előforduláshalmazából. Az sem kizárt, persze, hogy mindkét osztály (egyed típus) minden egyede kapcsolatban van a másik egyed típus

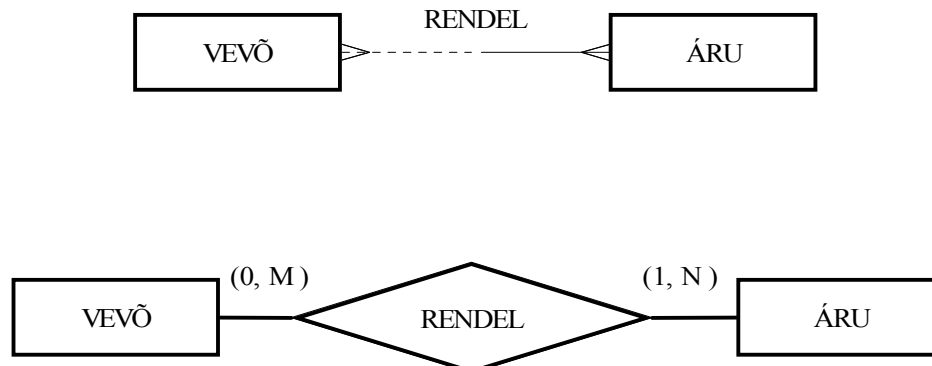
(osztály) valamelyik elemével. Ennek a kapcsolattípusnak a 7.2. ábra mutatja be a grafikus ábrázolását. Az E_1 egyedtípus (VEVŐ) egyedei közül egynek sem kell kapcsolatban lennie az E_2 egyedtípus (ÁRU) egyedeivel (vagy egyedei közül akár eggyel is), de eggyel, vagy akár többel is kapcsolatban lehetnek. Mindez természetesen fordítva is érvényes: az E_2 egyedtípus (ÁRU) egyedei egy vagy több egyeddel létesíthetnek kapcsolatot az E_1 egyedtípus (VEVŐ) egyedei közül, de akár eggyel sem.



7.2. ábra.

2. $a_1=1$ és $a_2=0$ $[R(E_1(0,M):E_2(1,N))]$

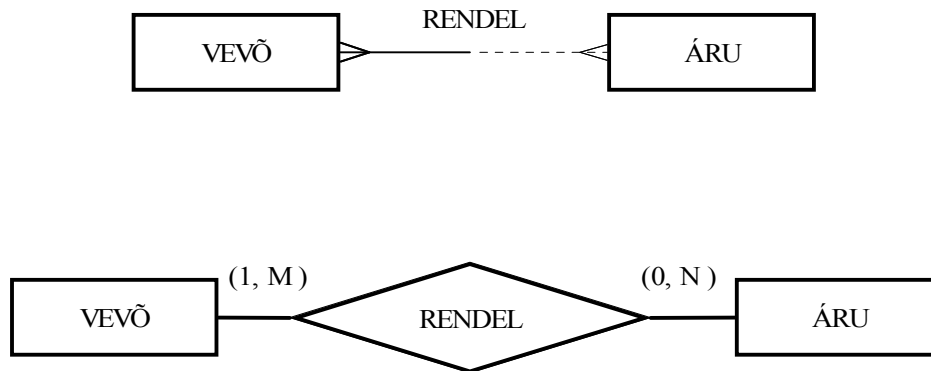
Ebben az esetben az E_2 egyedtípus (ÁRU) egyedeinek az E_1 egyedtípus (VEVŐ) egyedei közül legalább eggyel (de akár többel is) kapcsolatban kell lenniük. Tehát az E_2 egyedtípus (ÁRU) egyedeinek léte a kapcsolat meglétének (létezésének) a függvénye. Az E_1 egyedtípus (VEVŐ) egyedei kapcsolat nélkül is létezhetnek, de egy, sőt több kapcsolatot is létesíthetnek az E_2 egyedtípus (ÁRU) egyedeivel. A grafikus ábrázolás a 7.3. ábrán látható.



7.3. ábra.

3. $a_1=0$ és $a_2=1$ $[R(E_1(1, M):E_2(0, N))]$

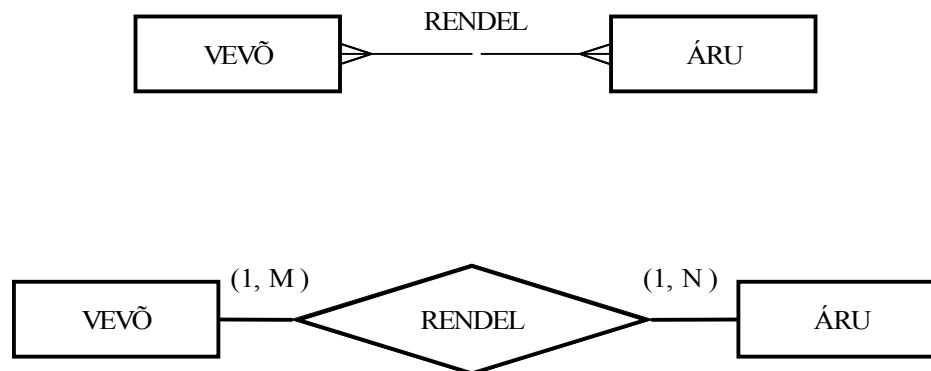
A mostani alapállapot az előző ellentettje: az E_1 egyedtípus (VEVŐ) egyedeinek az E_2 egyedtípus (ÁRU) egyedei közül legalább eggyel (de akár többel is) kapcsolatban kell lenniük (lehetnek). Tehát az E_1 egyedtípus (VEVŐ) egyedeinek léte a kapcsolat meglétének (létezésének) a függvénye. Az E_2 egyedtípus (ÁRU) egyedei kapcsolat nélkül is létezhetnek, de egy, sőt több kapcsolatot is létesíthetnek az E_1 egyedtípus (VEVŐ) egyedeivel. A grafikus ábrázolás a 7.4. ábrán látható.



7.4. ábra.

4. $a_1=1$ és $a_2=1$ $[R(E_1(1, M):E_2(1, N))]$

A negyedik kapcsolattípus szemantikája a következő. Mindkét egyedtípus minden egyed-előfordulása kapcsolatban kell, hogy legyen a másik egyedtípus legalább egy egyed-előfordulásával. Mind a két egyedtípusban megjelenő egyed-előfordulásnak legalább egy létező konkrét kapcsolat az előfeltétele. Ez nagyon szigorú megszorításnak számít. Az E_1 egyedtípus (VEVŐ) minden egyedének az E_2 egyedtípus (ÁRU) legalább egy egyedével (de akár többel is) kapcsolatban kell lenniük (lehetnek), és mindez fordítva is igaz. A grafikai megfogalmazás 7.5. ábrán látható.



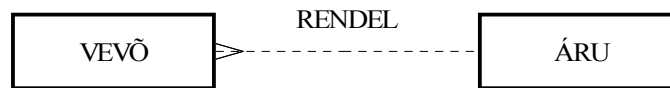
7.5. ábra.

7.5.1.2. Az 1:N típusú kapcsolat-fok szemantikája

Erre a kapcsolat-fok típusra a rögzített maximális kardinalitások értéke $b_1 = N$ i $b_2 = 1$. A minimális kardinalitás-értékek itt is négy kombinációt adnak.

1. $a_1 = 0$ és $a_2 = 0$ $[R(E_1(0,1):E_2(0,N))]$

Az E_1 egyedtípus (VEVŐ) egyedeinek nem kell kapcsolatot létesíteniük, de ha megteszik, úgy legfeljebb egy egyeddel állhatnak kapcsolatban az E_2 egyedtípus (ÁRU) egyedei közül. Az E_2 egyedtípus (ÁRU) egyedei egy vagy több egyeddel létesíthetnek kapcsolatot az E_1 egyedtípus (VEVŐ) egyedei közül, de akár eggyel sem.

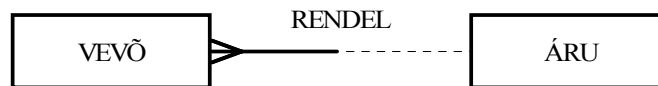


7.6. ábra.

A grafikon nyelven mindezt a 7.6. ábra fogalmazza meg.

2. $a_1 = 0$ és $a_2 = 1$ $[R(E_1(1,1):E_2(0,N))]$

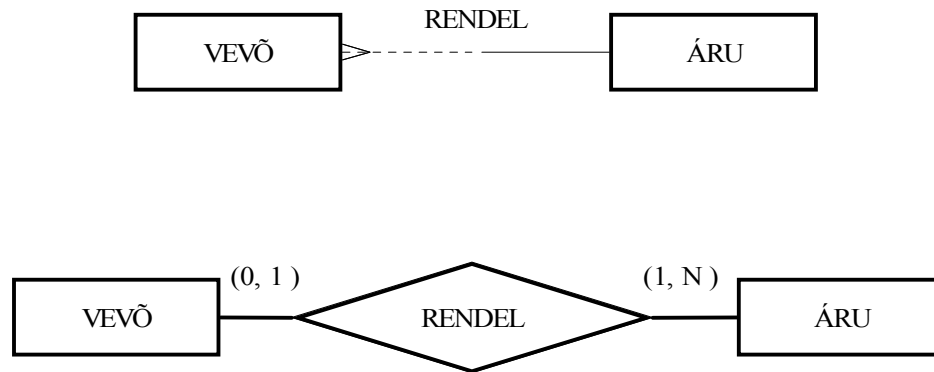
Itt az E_1 (VEVŐ) minden egyede kell, hogy létesítsen egy és csak egy kapcsolatot az E_2 (ÁRU) egy egyedével, míg ők (az E_2 egyedei) 0, 1 vagy akár több kapcsolatot is létesíthetnek az E_1 egyedeivel. Grafikus nyelven ezt a 7.7. ábra írja le.



7.7. ábra.

3. $a_1=1$ és $a_2=0$ $[R(E_1(0,1):E_2(1,N))]$

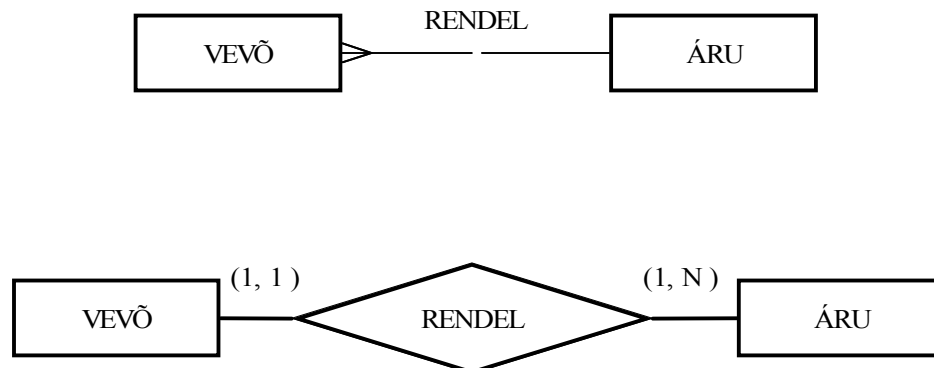
A kapcsolat szemantikája: az E_1 (VEVŐ) egyedeinek nem kell kapcsolatot létesíteniük az E_2 (ÁRU) egyedeivel, de ha megteszik, akkor azt csak eggyel tehetik meg. Az E_2 (ÁRU) egyedeinek legalább egy kapcsolattal kell rendelkezniük, de akár több kapcsolatot is létesíthetnek az E_1 (VEVŐ) egyedeivel. Amint látjuk, most az E_2 egyedeinek léte függ az E_1 egyedeivel való kapcsolat létezésétől. Ezt a 7.8. ábra mutatja be.



7.8. ábra.

4. $a_1=1$ és $a_2=1$ $[R(E_1(1,1):E_2(1,N))]$

Ebben a kapcsolatban az E_1 (VEVŐ) egyedei egy és csak egy kapcsolatot létesíthetnek (egyet muszáj, de attól többet nem szabad) az E_2 (ÁRU) egyedeivel, az E_2 egyedei pedig legalább egy egyeddel kell, hogy kapcsolatban legyenek az E_1 egyedtípusból. Itt mindkét egydtípus elemeinek léte függ a kapcsolat létezésétől. Grafikus megfogalmazási módon ez a 7.9. ábrán látható.



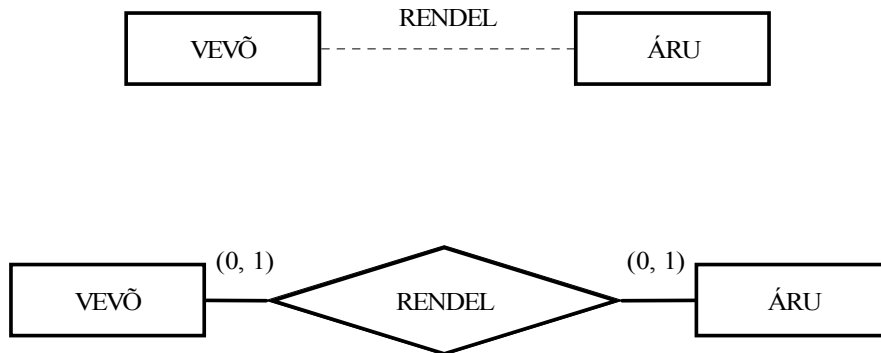
7.9. ábra.

7.5.1.3. Az 1:1 típusú kapcsolat-fok szemantikája

Ennél a kapcsolat-fok típusnál a maximális kardinalitásérték megegyeznek: $b_1 = b_2 = 1$, és az előzőekhez hasonlóan négy esetet fogunk elemezni a minimális kardinalitások alapján.

1. $a_1 = 0$ és $a_2 = 0$ $[R(E_1(0,1):E_2(0,1))]$

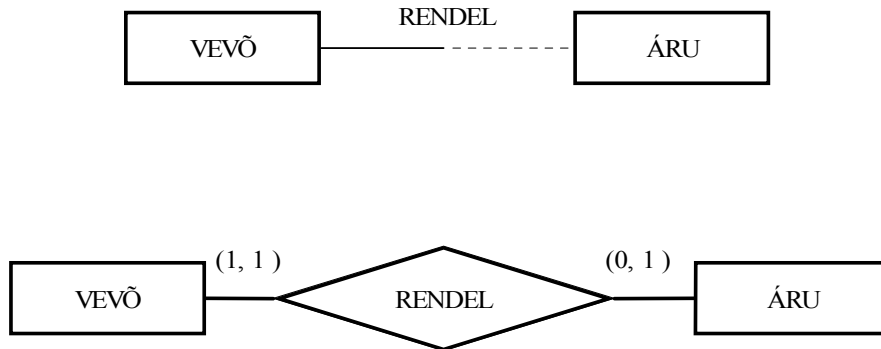
Ez a kapcsolat legfeljebb egy kapcsolatot engedélyez mindkét egyedtípus egyedeinek. Tehát az E_1 (VEVŐ) bármelyik egyede 0 vagy 1 kapcsolattal rendelkezhet az E_2 (ÁRU) bármelyik egyedével, és ez fordítottn is érvényes: az E_2 elemei kapcsolat nélkül is, de legfeljebb egy kapcsolat birtokában létezhetnek. A kapcsolat grafikusán is ábrázolható (7.10. ábra).



7.10. ábra

2. $a_1 = 0$ és $a_2 = 1$ $[R(E_1(1,1):E_2(0,1))]$

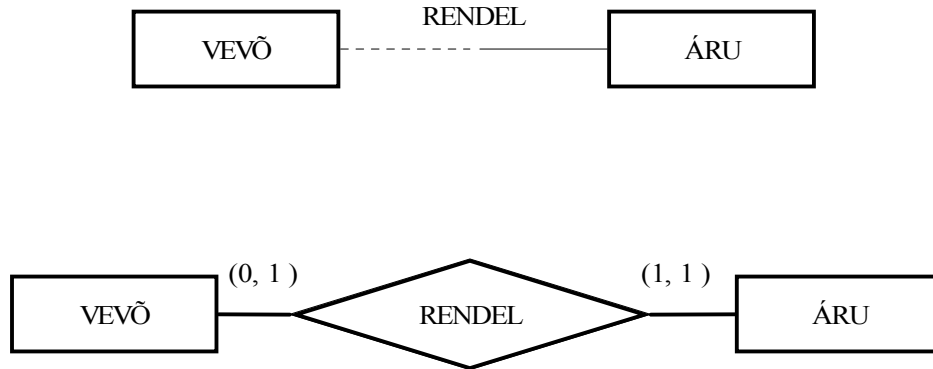
Az E_1 (VEVŐ) egyedtípus minden egyede egy és csak egy egyeddel lehet kapcsolatban az E_2 (ÁRU) egyedtípusból, az E_2 egyedei pedig 0 vagy 1 kapcsolatot létesíthetnek az E_1 egyedeivel. Ezt mutatja a 7.11. ábra.



7.11. ábra.

3. $a_1=1$ és $a_2=0$ $[R(E_1(0,1):E_2(1,1))]$

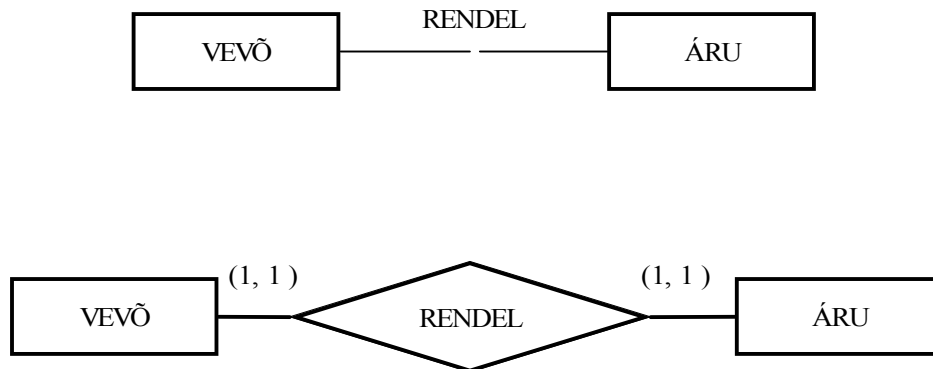
Ez a kapcsolat az előző tükörképének felel meg: Az E_1 (VEVŐ) egyedtípus egyedei 0 vagy 1 kapcsolatot létesíthetnek az E_2 (ÁRU) egyedeivel, az E_2 egyedei pedig egy és csak egy egyeddel lehetnek kapcsolatban az E_1 egyedtípusból. Az E_2 egyedtípus egyedeinek egzisztenciális függése a kapcsolatoktól szembetűnő (hiszen a minimális kardinalitás értéke 1). A 7.12. ábra erről szól.



7.12. ábra.

4. $a_1=1$ és $a_2=1$ $[R(E_1(1,1):E_2(1,1))]$

A kapcsolat szemantikája a szimmetria miatt egyszerű: az E_1 (VEVŐ) egyedtípus minden egyede legalább és legfeljebb (tehát pontosan) egy egyeddel kell, hogy kapcsolatban legyen az E_2 (ÁRU) egyedtípusból, és ez fordítva is érvényes. Az egzisztenciális függés a kapcsolattól mindkét egyedtípus elemeire vonatkozik (7.13. ábra).



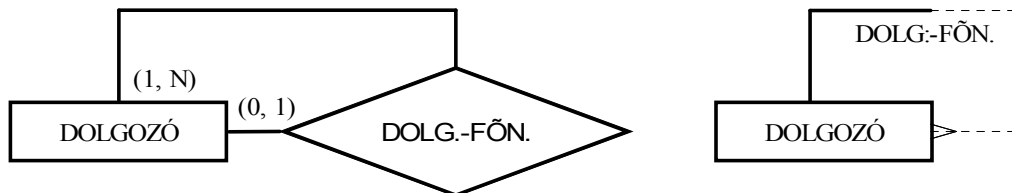
7.13. ábra.

7.5.2. A rekurzív vagy visszamutató egyedviszony

Az előzőekben különböző egyedtípusok közötti kapcsolattípusok fajtáit ismertük meg, illetve a különböző kapcsolattípusok egyedei között fellépő konkrét kapcsolatfajtákat. A megismert viszonyokat heterogén összefüggéseknek is szokás nevezni. Az ugyanazon egyedtípus egyedei között fellépő kapcsolatok fogalmi szintű tükrözését rekurzív kapcsolatoknak hívjuk. Ezek a homogén összefüggések fejezik ki a visszamutató (recursive, involuted) egyedviszonyt. A valamilyen közös ismerv alapján osztályozható rekurzív kapcsolatok rekurzív kapcsolattípusokba sorolhatók (a konkrét rekurzív kapcsolatok absztrakciójával eljutuk a rekurzív kapcsolattípusok fogalmához). A homogén összefüggések esetében az első és második egyedtípus megegyezik egymással, illetve az egyik a másiknak az alhalmaza (az egyed-előfordulás szemszögéből). Az előző (7.5.1.) pontban végzett kardinalitás-analízis, természetesen itt is érvényes.

A DOLGOZÓ egyedtípus DOLGOZÓ-FŐNÖKE rekurzív kapcsolattípusa szolgál a homogén összefüggések bemutatására (7.14. ábra) az alábbi kardinalitásértékeknek megfelelően):

Dolgozó - főnöke (Dolgozó - beosztott (0,1) : Dolgozó - főnök (1, N))

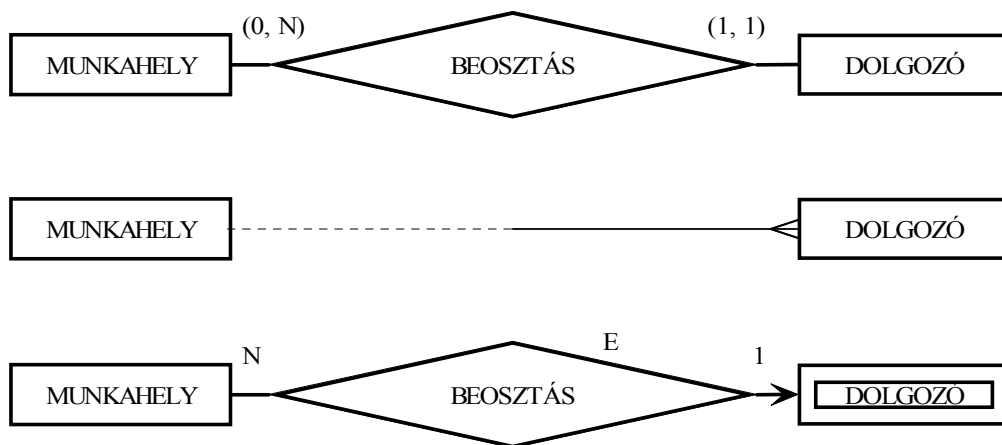


7.14. ábra.

7.5.3. Gyönge egyedtípus

Az egyedtípusokat és kapcsolattípusokat tartalmazó (az egyedtípus külső szerkezetét ábrázoló) diagramokat ER (a rövidítés az angol Entity és Relationship szavak kezdőbetűiből ered) diagramoknak is szokás nevezni. Az ilyen rajzokon többnyire csak a maximális kardinalitás-értékek szerepelnek, így az egzisztenciálisan függő egyedtípusok (azok, amelyeknek minimális kardinalitása 1) külön jelet kapnak, a “duplafalú téglalapot”, és gyönge egyedtípusoknak nevezzük őket. Ábrázolásban a kapcsolattípus “E” betűjelet visel (az egzisztenciális függés megjelölésére), a kapcsolattípust a gyönge egyedtípussal összekötő vonal

pedig a gyönge egyedtípus felé nyilazott. Az “E” betűjellel ellátott kapcsolattípusokat gyakran nevezik még gyönge kapcsolattípusoknak. Az egzisztencionálisan független egyedtípusokra és kapcsolattípusaikra használják még a “reguláris” jelzőt is a gyönge egyed- és kapcsolattípusoktól való megkülönböztetés céljából. Az gyönge egyedtípust tartalmazó szerkezet grafikus képe a 7.15. ábrán látható.

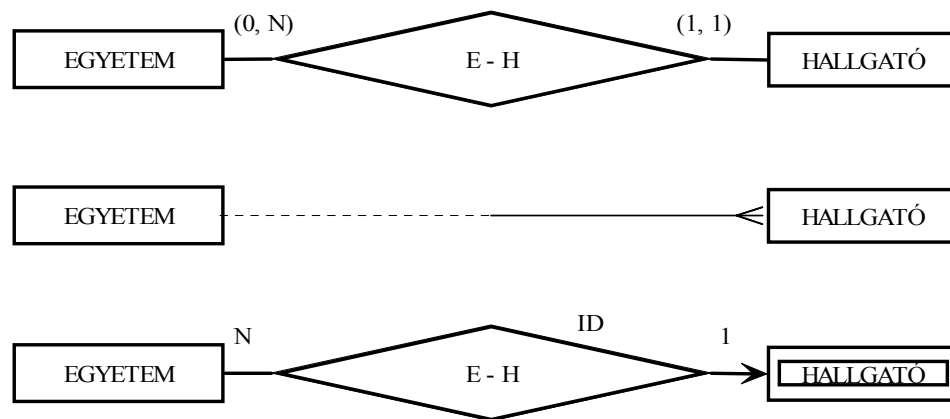


7.15. ábra.

7.5.4. Azonosító-függő egyedtípus

Az azonosító-függő egyedtípus a gyönge egyedtípusnak egy sajátos (specifikus) esete. Amennyiben a gyönge egyedtípus nem rendelkezik saját azonosítóval, hanem az egyik vele közvetlen kapcsolatban álló reguláris (egzisztencionálisan független) egyedtípus azonosítóját használja azonosítóként, akkor azt a gyönge egyedtípust azonosító-függő egyedtípusnak nevezzük. Az azonosító-függő egyedtípus egyedei tehát nem azonosíthatóak egyetlen saját tulajdonságtípusuk vagy tulajdonságtípuscsoportjuk alapján sem, hanem csak a hozzájuk azonosító-függő kapcsolattípussal kötődő reguláris egyedtípus azonosítójának “kölcsonvétele” (hozzácsatolásával). Az azonosító-függő kapcsolattípus (egyedtípus) minden esetben gyönge kapcsolattípust (gyönge egyedtípust) is jelent, de fordítva mindez nem érvényes (a gyönge egyedtípus nem jelenti azt, hogy azonosító-függő egyedtípusról van szó).

Az azonosító egyedtípus példájára a HALLGATÓ egyedtípust hozzuk fel amelynek a leckeönyv-szám az azonosítója az anyaegyetemen, de ez a leckeönyv-szám nem tudja őt azonosítani országos szinten, mert előfordulhat más egyetemen ugyanolyan leckeönyv-számmal rendelkező hallgató. Az országos szintű egyedi azonosítás céljából a 7.16. ábrán látható megoldás egy lehet az elfogadhatóak közül.



7.16. ábra.

7.5.5. Kapcsolattípusok közötti kapcsolatok

A kapcsolattípusok esetében a kapcsolat-foka és jellege (opcionálitása) mellett ügyelni illik arra a tényre is, hogy a kapcsolattípusok egymással is kapcsolatban lehetnek. Előfordulhat, hogy két vagy több kapcsolattípus egymást kölcsönösen kizárja. A kapcsolattípusok közötti kizáró kapcsolatok két félék lehetnek:

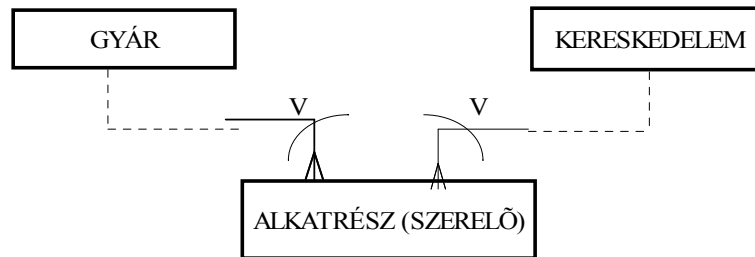
1. kizáró alárendelt és
2. kizáró fölérendelt kapcsolatok.

A kizáró alárendelt kapcsolat iskolapéldája lehet egy tetszőleges gyárnak a 7.17. ábrán látható alkatrészértékesítéssel kapcsolatos modellrésze. A gyár az alkatrészeket (alkatrész, félkész termék, késztermék) közvetlenül a fogyasztónak vagy a kereskedelemnek adja el (egyik kapcsolat kizárja a másikat).



7.17. ábra.

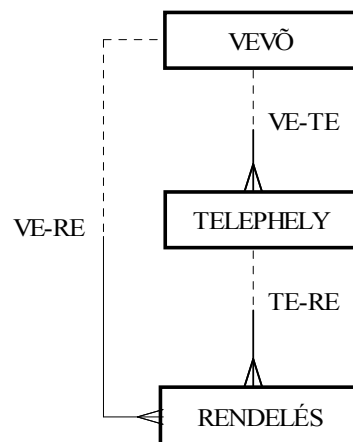
A másik (kapcsolattípusok közötti) kizáró kapcsolatfajta vegyük szemügyre egy szerelőüzem (nem termeli, csak összerakja a termékeket) modellrészletét. A szerelőüzem az alkatrészeket vagy közvetlenül az alkatrészek gyártóitól vagy a kereskedelemből szerzi be, ahogy ez a 7.18. ábrán is látható.



7.18. ábra.

7.5.6. Transzitiv szerkezetű modellrész

A tranzitív szerkezetű modellrész bemutatásához nem kell iskolapéldát szerkeszteni, mert a valós világ bővelkedik a “nyersanyagban”. Vegyük azt a helyzetet, hogy a gyár vevői között olyanok is vannak, amelyek kirendeltségekkel (telephelyekkel), azok pedig saját árurendelés-lehetőséggel is rendelkeznek (a nagy áruházláncok és áruházai). Ekkor a rendelés érkezik akár a központból, akár a telephelyről. A fent leírt helyzetet a 7.19. ábra tartalmazza.



7.19. ábra.

A tranzitív szerkezetű modellrész nyomban szembetűnik az elemzés folyamán, ez egyszerű, tiszta sor. A kérdés az, hogy maradhat-e vagy sem a tranzitív szerkezet a modellben (az egyedtípus külső szerkezetében)? Amennyiben az áthidaló szerkezet szemantikai jelentőséggel bír, úgy szükség van rá, egyébként viszont redundanciát okoz, vagyis eltávolítandó.

Az áthidaló szerkezet szükségességének megállapításához tételezzünk fel egy előfordulási halmazt a VEVŐ egyedtípusnak. A feltételezett vevő-előforduláshalmaz a VE-RE kapcsolattípuson keresztül egy rendelés-előforduláshalmazt eredményez. Jelöljük meg ezt a rendeléshalmazt R_1 -el. Ha most a feltételezett vevőelőfordulásokból a VE-TE kapcsolattípus segítségével egy telephely-előforduláshalmazt kapunk, amelyet a TE-RE kapcsolattípus segítségével egy újabb rendelés-előforduláshalmaz előállítására használunk, akkor megkapjuk a, mondjuk R_2 jelű rendelés-előforduláshalmazt. Amennyiben a két halmaz (az R_1 és az R_2) megegyezik egymással, az áthidaló (VE-RE) kapcsolat redundanciát okoz, tehát eltávolítandó.

7.6.AZ EGYEDTÍPUS BELSŐ SZERKEZETE

Az egyedtípus külső szerkezetének megfogalmazása először a valós rendszer elemzésének elején történik meg, amikor a rendszerelemzők megismerik azokat az (automatizálásra szánt) üzleti fogalmakat (és viszonyaikat) amelyekről naprakész adatokat szeretnének vezetni az információs rendszer megrendelői, felhasználói. Mivel az elemzés (a valós világ megismerése) a (valós) rendszertől, mint egészről indul ki, és halad a részletek felé, az egyedtípus külső szerkezetének ilyen kiépítése az adatmodell top-down megközelítésének felel meg.

A részletek megismerése folyamán a rendszerelemzők a foglamak (egyedek, egyedtípusok) és egymás közötti viszonyaik (kapcsolatok, kapcsolattípusok) mellett a valós világ fogalmait leíró jellemzőkkel (tulajdonság, tulajdonságtípus) is kapcsolatba kerülnek (megismerkednek). A megismert tulajdonságtípusok egymással is kapcsolatban lehetnek. Kizárólag a tulajdonságtípusok kapcsolatainak elemzésével, egy szabványosított eljárás (normalizáció) segítségével, kialakíthatók az egyed- és kapcsolattípusok (a logikai adatmodell, vagy az egyedtípus külső szerkezete). Mivel a tulajdonságtípusok és azok viszonyai az egyedtípus belső szerkezetét alkotják (a "részleteket"), a normalizáció alkalmazásával az adatmodell (az egyedtípus külső szerkezete) bottom-up megközelítésben alakítható ki.

Mivel az adott információs rendszer adatmodellje egy és oszthatatlan, a top-down megközelítésben kialakított megoldásnak egyeznie kell a bottom-up megközelítésben kialakítottal (hiszen ugyanarra az információs rendszerre vonatkoznak). Amennyiben

különbségek mutatkoznak az eltérő megközelítésben kialakított adatmodelle között, úgy azokat egymással és a valós világgal egyeztetni kell.

7.6.1. Tulajdonságtípusok közötti függések

A tulajdonságtípusok közötti függések csak adott egyedtípuson belül értelmezhetőek. A legismertebb tulajdonságok közötti összefüggés a funkcionális függés, amely lehet teljes vagy részleges. E kétfajta függés mellett még egy harmadikra, a tranzitív függésre is szükségünk lesz a harmadik normálforma kialakításához (amelyet általánosan elegendőnek tartanak az egyedtípus belső szerkezetének meghatározásánál). A harmadik normálformát kielégítő tulajdonsághalmazok egyedtípusoknak tekinthetők.

Az E_1 egyedtípus Y tulajdonságtípusa funkcionálisan függ (FD – functional dependence) az E_1 egyedtípus X tulajdonságtípusától, abban (és csakis abban) az esetben, ha az E_1 egyedtípus minden előfordulásában X minden értékét mindig csak egy Y érték kíséri. Az előbb leírt funkcionális függés az $f: X \rightarrow Y$ vagy rövidítve az $X \rightarrow Y$ jelöléssel is megadható. Maga a jelölés szavakkal leírva így szól: X funkcionálisan meghatározza Y -t, vagy Y funkcionálisan függ X -től. Általános esetben az X és az Y tulajdonságtípus-halmazokat is jelölhetnek.

A funkcionális függés fordított irányban függetlenséget hordoz, azaz, ha $X \rightarrow Y$, akkor érvényes, hogy $Y \not\rightarrow X$ (a függetlenséget a „ $\not\rightarrow$ ” jellel ábrázoljuk). Mutassunk rá arra is, hogy a funkcionális függés a két tulajdonságtípus közötti hierarchikus összefüggést hordozza magában.

In medias res. Példa: $\text{Leckekönyvszám} \rightarrow \text{Hallgató_neve}$. A két tulajdonságtípus értékészletének értékei között $N:1$ típusú kapcsolat van. Az azonos nevű hallgatók különböző leckekönyvszámot kapnak (N), de minden leckekönyvszám csak egy (1) hallgatóra vonatkozik.

A funkcionális függés lehet erős (ha minden bal oldalon lévő előforduláshoz, meghatározóhoz kell kötődnie jobb oldali előfordulásnak, meghatározottnak) vagy gyenge (vannak olyan X -értékek, amelyekhez nem tartoznak Y -értékek).

Példa: a $\text{Leckekönyvszám} \rightarrow \text{Hallgató_név}$ funkcionális függés erős, mert minden leckekönyvszámhoz tartoznia kell egy hallgatónévnek, a $\text{Leckekönyvszám} \rightarrow \text{Megítélt_kölsönfajta}$ viszont gyenge funkcionális függés, hiszen vannak olyan egyetemisták, akik nem igényelnek kölcsönt.

Last but not least: a funkcionális függés lehet teljes (full dependence) vagy részleges (partial dependence). Ha a függés bal oldalán (a meghatározó tulajdonság) egyetlen attribútum (tulajdonságtípus) áll, akkor elemi függésről beszélünk, ha viszont a bal oldal több tényezőből

(tulajdonságtípus) áll, akkor összetett függéssel van dolgunk. A funkcionális függések teljes vagy részleges volta csak az összetett függések esetében vizsgálható. Az összetett funkcionális függés akkor teljes, ha a jobb oldal (a meghatározott) a teljes (összetett) bal oldaltól (a meghatározótól) függ, de úgy, hogy ha az összetett bal oldal bármely összetevőjét elhagyjuk, a funkcionális függés megszűnik. Amennyiben az összetett meghatározó bármely tagját elhagyva a funkcionális függvény nem szűnik meg, részleges funkcionális függésről va szó.

Példa: nézzük az alábbi szerkezettel megadott SIKERES_VIZSGA egyedtípust és azon belül két kiválasztott funkcionális függést:

SIKERES_VIZSGA(Leckekönyvszám+Tantárgykód, osztályzat, ..., Tantárgy_neve)

Leckekönyvszám+Tantárgykód $\rightarrow$ osztályzat

Leckekönyvszám+Tantárgykód $\rightarrow$ Tárgy_neve.

Az első funkcionális függés nyilvánvalóan teljes, mert a bal oldal egyetlen tulajdonságtípusa sem határozza meg egymagában a jobb oldalt. A második funkcionális függés azonban részleges, mert a leckekönyvszám eltávolítása a meghatározóból (bal oldal) nem szünteti meg a funkcionális függést (a tárgykód önmagában is meghatározza a tárgy nevét).

Szemantikailag az $f: X \rightarrow Y$ funkcinális függés időben változó függvénynek tekinthető. Ez azt jelenti, hogy a meghatározó (bal oldal) és a meghatározott (jobb oldal) is változnak az időben, ezért más-más időpontban az X tulajdonságtípus ugyanazon előfordulás-értékéhez az Y tulajdonságtípus különböző előfordulás-értéke tartozhat. Ehhez a ténymegállapításhoz a következő példa (funkcionális függés) kapcsolódik: Kocsirendszám $\rightarrow$ Kötelező_szervizek_száma. A meghatározóhoz kapcsolt érték (a meghatározott értéke) változik az időben, hiszen a gépkocsi használati ideje folyamán az elvégzett kötelező szervizek száma folyamatosan növekszik.

A funkcionális függés intenzionális megszorításnak, korlátnak tekinthető, mert egy adott egyedtípus tulajdonságtípus-halmazára vonatkozik. A funkcionális függések felderítéséhez a valós rendszert (az ott használt fogalmakat) figyelmes elemzésnek vetjük alá az egyedtípusok jellemzőinek feltárásával, tulajdonságtípusainak megfogalmazásával és értéktartományaik meghatározásával.

A tulajdonságtípust (attribútumot) vagy a tulajdonságtípusoknak olyan csoportját, amelyik az adott egyedtípus minden előfordulás-értékére eltérő értékeket vesz fel azonosítónak, kulcsnak, esetleg kulcsjelöltnek nevezzük. Az azonosítónak minimális felépítésűnek kell lennie, másszóval az azonosító egyetlen valódi alhalmaza sem töltheti be az azonosító (kulcsjelölt) szerepét. Más megfogalmazásban ez azt jelenti, hogy minden funkcionális függésnek, amelyekben a meghatározó az azonosító (kulcs), teljesnek kell lennie.

Ha az egyik (E_1) egyedtypusban szerepel egy olyan másodlagos (leíró) tulajdonságtípus, amely egy másik (E_2) egyedtypusban azonosító (elsődleges kulcs), akkor a szóban forgó tulajdonságtípust az E_1 egyedtypusban külső (idegen) kulcsnak nevezzük, mert az E_1 -ben ugyan nem kulcs, de valahol máshol igen.

Az azonosító állhat egy vagy több tulajdonságtípusból. Az előbbi azonosítót eleminek (elementary) minősítjük, az utóbbit összetettnek (compound v. composite). Ha az elsődleges kulcs két vagy több (különböző egyedtypusokból eredő) külső kulcsból és az egyed legalább egy saját tulajdonságtípusából, akkor azt mondjuk, hogy hierarchikus kulcsunk van. (Egyesek a különböző egyedtypusokból eredő külső kulcsokból álló azonosítót is hierarchikus kulcsnak tekintik.)

A tranzitív függés lényege a következő. Jelöljük az adott egyedtypus tulajdonságtípusainak halmazát A -val, és jelölje az X és Y a tulajdonságtípusok olyan halmazát amelyekre érvényes, hogy $(X, Y \subseteq A)$ és $X \rightarrow Y$. Az Y akkor függ tranzitíven az X -től, ha létezik a tulajdonságtípusoknak egy olyan Z -vel jelölt halmaza, amelyre igaz, hogy $Z \subseteq A$, és érvényes a következő két funkcionális függés: $X \rightarrow Z$ és $Z \rightarrow Y$, de, ugyanakkor, természetesen (a funkcionális függések megszorításai következtében) a $Z \rightarrow X$ és $Y \rightarrow Z$ funkcionális függéseknek nem szabad fennállniuk.

A tranzitív függés kevésbé matematizált formája: az egyedtypus leíró (nem-kulcs) Y tulajdonságtípusa csak akkor függ tranzitíven az egyedtypus azonosítójától (X), ha őt (Y) a kulcson (X) kívül, egy másik Z tulajdonságtípus is funkcionálisan meghatározza, amely (Z) maga is funkcionálisan függ a kulcstól (X).

A dekompozíció az az eljárás, amelyik az egyedtypusokat egyenként, egymástól függetlenül vizsgálja, és a kedvezőtlen (nem megfelelő) belső szerkezetű egyedtypusokat osztással (darabolással) megfelelőbb belső szerkezetűre hozza (bontja).

Jelöljük az E_1 egyedtypus tulajdonságtípus-halmazát A -val, és legyen két olyan, $X \subseteq A$ és $Y \subseteq A$ al(rész)halmazunk, hogy $X \cup Y = A$. Az X és Y akkor alkotnak dekompozíciós párt az E_1 egyedtypusra vonatkozóan, ha az E_1 két, modjuk t_i és t_j ($t_i = \text{ext}(E_1)$ és $t_j = \text{ext}(E_1)$) egyed-előfordulásának metszethalmaz-egyenlőségéből az következik, hogy létezik olyan egyed-előfordulás (t), amelyre $t = t_i$ az X (rész)halmazon, és ugyancsak létezik olyan t egyed-előfordulás, amelyre $t = t_j$ az Y (rész)halmazon. A meghatározás (definíció) formális-halmazelméleti leírása a következő³:

$\forall t_i, t_j \in E_1$ esetén, melyre $t_i(X \cap Y) = t_j(X \cap Y) \exists t \in E_1$, melyre $t(X) = t_i(X)$ és $t(Y) = t_j(Y)$, akkor X és Y dekompozíciós párt alkotnak az E_1 egyedtypusra vonatkozóan.

Példa⁴: Az E_1 egyedtípus részhalmazai legyenek a következők: $X = \{A_1, A_2, A_3, A_4\}$ és $Y = \{A_3, A_4, A_5, A_6\}$, vagyis a metszet $X \cap Y = \{A_3, A_4\}$, és nézzük az E_1 egyedtípus egyedelőfordulásainak a következő részhalmazát (7.1. tábla).

7.1. tábla

	A_1	A_2	A_3	A_4	A_5	A_6
t_1	a	b	c	d	e	f
t_2	g	f	c	d	h	e
t_3	a	b	c	d	h	e
t_4	.	.	.	.	.	.

A 7.1. tábla egyedelőfordulásai alapján arra következtethetünk, hogy az X és Y részhalmazok az E_1 egyedtípus dekompozíciós párját alkotják, mert:

1. $t_1(X \cap Y) = t_2(X \cap Y) = \{c, d\}$ és
2. $t_3(X) = t_1(X)$ és $t_3(Y) = t_2(Y)$

7.6.2. Normálformák

A normálformák az egyedtípus belső szerkezetének az állapotát, minőségét fejezik ki (minősítik). Több normálforma is ismeretes (attól valamivel kevesebb használatos is), amelyeket általában az xNF jelöléssel illetnek. Az x számérték, az NF pedig a Normal Form angol szavak rövidítéséből ered. A normálformák jelölési sorában csak egyetlen tag különbözik az xNF jelöléstől, íme a sorban az első a legkedvezőtlenebb bormálalak): 0NF, 1NF, 2NF, 3NF, BCNF (Boyce-Codd normálforma), 4NF és 5NF.

A normalizálás az egyedtípus belső szerkezetének javítását célzó matematikai eljárás, algoritmus, amelynek lépéseit szükségszerűen csak szemantikai elemzés (a valós világgal való ismételt egyeztetés) kíséretében ajánlatos végezhajtani. Az optimális belső egyedtípus-szerkezetet biztosító normalizálás, mint eljárás vagy folyamat az egyedtípus külső szerkezetére is kihatással van. A logikai adatstruktúra (adatmodell) kiépítése (normalizálás) kétféle eljárással vihető végbe: dekompozícióval és szintézissel. A továbbiakban a normalizálás dekompozíciós eljárását fogjuk megismerni.

³ A dekompozíció egyszerűsített definíciójának alapjául szolgáló forrás: [25], 82. oldal.

⁴ Forrás: [25], 82. oldal.

Az egyedtípus ismétlődő adatot (csoportot vagy adatszoportot) tartalmaz, ha egy (vagy több) tulajdonságtípus az egyed-előfordulásokra (tehát a tulajdonságtípus-előfordulás-ok) két vagy több értékkel is rendelkezik. Ez más megfogalmazásban így hangzik: az ismétlődő adatot tartalmazó egyedtípus funkcionálisan független tulajdonságtípust tartalmaz.

A nem-normalizált, 0NF alakú egyedtípusok legalább egy funkcionálisan független, leíró tulajdonságtípust is magukba foglalnak.

Példa: HALLGATÓ (Leckekönyvszám, Hallgató_neve, ...Tantárgykód, Tantárgynév, Vizsgadátum, Osztályzat). A hallgató egyedtípus egyed-előfordulásai a 7.20 ábrán, táblázatban helyezkednek el. A Tantárgykód, a Tantárgynév, a Vizsgadátum és az Osztályzat leíró tulajdonságtípusok az egyes egyed-előfordulásokra több értéket is felvehetnek, vagyis ők nem

HALLGATÓ

Leckekönyvszám	Hallgató neve	...	Tantárgykód	Tantárgynév	Vizsgadátum	Osztályzat
E494	Beszédes Ákos		112	Matematika	2000.09.12.	9
			147	Fizika	2000.09.15.	7
			174	Elektrotechnika	2000.10.05.	6
E495	Néma Levente		174	Elektrotechnika	2000.07.14.	8
			112	Matematika	2000.09.15.	8
...	...		...	...	...	...

7.20. ábra.

függnek funkcionálisan a hallgató egyedtípus azonosítójától, a leckekönyvszámtól.

Az egyedtípus belső szerkezete bizonyosan legalább első normálformájú (1NF), ha minden leíró tulajdonságtípusa funkcionálisan függ az azonosítótól (minden egyed-előfordulásban minden tulajdonságtípus-előfordulás egy értéket tartalmaz.

Az egyedtípus belső szerkezetének első normálalakra hozása az azonosítótól funkcionálisan független tulajdonságtípusok egyetípusból való eltávolításából áll. Az eredeti egyedtípust dekompozícióval két új egyedtípusra bontjuk: az elsőben az eredeti egyedtípus funkcionálisan függő tulajdonságtípusai maradnak (és természetesen az azonosító is), a funkcionálisan függetleneket a második egyedtípusba helyezzük az eredeti egyedtípus azonosítójával kiegészítve. A bontás utáni két 1NF szerkezetű egyedtípus a 7.21. ábrán látható.

A második (új) egyedtípus a SIKERES_VIZSGA elnevezést kapta és összetett azonosítója van: Leckekönyvszám+Tantárgykód. Az első egyedtípus (az eredeti maradéka) azonosítója nem változott.

Az egyedtípus belső szerkezete legalább második normálalakú (2NF), ha első normálformában (1NF) van, és minden leíró (nem-kulcs) tulajdonságtípusa teljes funkcinális függéssel függ az azonosítótól (a teljes azonosítótól).

Az előző definíció két záradéka a következő:

1. ha az egyedtípusnak elemi (atomi) azonosítója van, akkor az 1NF szerkezet megléte automatikusan és bizonyosan a 2NF szerkezetet is jelenti és
2. a csupakulcs egyed (nincs leíró tulajdonságtípusa) mindig 2NF szerkezetű.

HALLGATÓ

Leckekönyvszám	Hallgató_neve	...
E494	Beszédes Ákos	
E495	Néma Levente	
...	...	

SIKERES_VIZSGA

Leckekönyvszám	Tantárgykód	Tantárgynév	Vizsgadátum	Osztályzat
E494	112	Matematika	2000.09.12.	9
E494	147	Fizika	2000.09.15.	7
E494	174	Elektrotechnika	2000.10.05.	6
E495	174	Elektrotechnika	2000.07.14.	8
E495	112	Matematika	2000.09.15.	8
...	...	...	...	...

7.21. ábra.

Az egyedtípus második normálalakra hozása a részleges függések kikeresésével kezdődik, hogy aztán eltávolíthassuk őket. A 2NF kialakítása három lépésből áll:

1. az egyedtípus azonosítójából kiválasztjuk azokat a tulajdonságtípusokat (vagy azok csoportját), amelyek maguk is funkcionálisan meghatároznak leíró tulajdonságtípusokat,
2. az előző pontban kiválasztott tulajdonságtípus(csoport)okat a hozzájuk tartozó (tőlük funkcionálisan függő) leíró tulajdonságtípusokkal külön egyedtípusokba gyűjtjük,
3. a teljes azonosítótól funkcionálisan függő leíró tulajdonságtípusok maradnak az eredeti egyedtípusban.

A HALLGATÓ egyedtípusnak elemi azonosítója van, ezért a 2NF ellenőrzését (kialakítását) a SIKERES_VIZSGA egyedtípuson mutatjuk be. Itt az azonosító része, a Tantárgykód funkcionálisan meghatározza a Tantárgynevet, tehát őket kell kiválasztani új egyedbe (TANTÁRGY). A 2NF formájú egyedtípusok a 7.22. ábrán láthatóak.

Az egyedtípus belső szerkezete bizonyosan legalább harmadik normálalakú (3NF), ha 2NF-ben van, és egyetlen leíró (nem-kulcs) tulajdonságtípus sem függ tranzitíven az egyedtípus azonosítójától.

Az ilyen, harmadik normálformájú belső szerkezettel rendelkező egyedtípusról elmondható, hogy minden leíró tulajdonságtípusa funkcionálisan függ az azonosítótól (1NF), teljes funkcionális függéssel függ az (összetett) azonosítótól (2NF), és a többi leíró tulajdonságtípustól funkcionálisan független (3NF).

HALLGATÓ

Leckekönyvszám	Hallgató neve	...
E494	Beszédes Ákos	
E495	Néma Levente	
...	...	

TANTÁRGY

Tantárgykód	Tantárgynév
112	Matematika
147	Fizika
174	Elektrotechnika
...	...

SIKERES VIZSGA

Leckekönyvszám	Tantárgykód	Vizsgadátum	Osztályzat
E494	112	2000.09.12.	9
E494	147	2000.09.15.	7
E494	174	2000.10.05.	6
E495	174	2000.07.14.	8
E495	112	2000.09.15.	8
...	...	...	...

7.22. ábra.

A harmadik normálalak elérése az egyedtípusban felfedezett tranzitív függés tulajdonságtípusainak egy új egyedtípusba történő kiválasztásával (eltávolításával) bonyolódik. Így az eredeti egyedtípus nem fogja tartalmazni a tranzitív függés meghatározott tulajdonságtípusait. Az új egyedtípusok a tranzitív függés meghatározó és meghatározott tulajdonságtípusait is felölelik.

RENDELÉS

Rendelésszám	Vevőkód	Vevőnév	Rendelésdátum
17494	112	Carnex	2000.09.12.
15944	147	Pionir	2000.09.15.
13675	174	Sever	2000.10.05.
19395	174	Sever	2000.07.14.
25495	112	Carnex	2000.09.15.
...	...		...

RENDELÉS 1

Rendelésszám	Vevőkód	Rendelésdátum
17494	112	2000.09.12.
15944	147	2000.09.15.
13675	174	2000.10.05.
19395	174	2000.07.14.
25495	112	2000.09.15.
...	...	...

VEVŐ

Rendelésszám	Vevőnév
112	Carnex
147	Pionir
174	Sever
...	

7.23. ábra.

A 3NF alakra hozás példáját kereskedelemben használt modellrésszel világítjuk meg. Rendelésnyilvántartási problémáról van szó, a második normálalakban lévő RENDELÉS egyedtípus belső szerkezetének 3NF-re javítását, pontosabban a javítás eredményét 8a bontás utáni állapotot) mutatja a 7.23. ábra.

Az előző normalizációs folyamatok (lépések, amelyek az egyedtípus belső szerkezetének javításait célozzák meg) dekompozíciós eljárással bonyolódtak, az eredeti egyedtípus (és a folyamat lépéseiben létrejövő újabb egyedtípusok) két vagy több egyedtípusra bontásával. A normalizáció az egyedtípus belső szerkezetét javítja, finomítja.

A dekompozíciós eljárás a tulajdonságtípusok kivetítésével (eltávolításával) jár: az 1NF-re hozáskor az azonosítótól független tulajdonságtípusokra, a 2NF esetében az összetett azonosítótól funkcionálisan nem teljesen (részlegesen) függő tulajdonságtípusokra, illetve a 3NF megvalósításánál az azonosítótól tranzitíven is függő tulajdonságtípusokra vonatkozóan. A kivetítés mindig új egyedtípusba történik, de a lépések elvégzésekor (befejezésekor) szükségszerű egyedtípus-ellenőrzések, esetleges egyedtípus-összevonások is történhetnek. Egy kikötést azonban tiszteletben kell tartani: a folyamat visszafordíthatóságát, vagyis, hogy az új egyedtípusokból mindig visszaállítható legyen az eredeti kiindulási) egyedtípus. Az ilyen dekompozíciós eljárás neve kapcsolatvesztésmentes v. veszteségmentes dekompozíció (non-loss v. loss-less decomposition).

A kivetítés vagy szétválasztás ellentétes művelete az összekapcsolás (join). Két egyedtípus összekapcsolásához szükségszerű az azonos tartalmú (esetleg azonos nevű is lehet, de ez nem feltétel) tulajdonságtípus jelenléte mindkét egyedtípusban. Az összekapcsolás veszteségmentessége ugyancsak kikötés, amelyet be kell tartani: az új, összekapcsolással létrejött egyedtípusnak hordoznia kell a szülő-egyedek minden információját (non-loss v. loss-less join).

7.6.3. Az egyedtípus szerkezeteinek megfeleltethetősége

Az előzőekből már tudjuk, hogy az egyedtípus optimális belső szerkezetének kialakításánál a külső szerkezet is változik. A 3NF alakú egyedtípusok alapján könnyen kiépíthető a logikai adatmodell (a transzverziók vagy az egyedtípus külső szerkezete) ha tartjuk magunkat az előbbekben erre vonatkozó ismeretekhez, amelyeket három megllapításba sűrítünk::

1. a külső kulcsok a fölérendelt egyedtípusokra mutatnak (a kapcsolattípus 1:N kardinalitású – az egyedtípus, amelyben a tulajdonságtípus külső kulcs az N oldalt képviseli, az egyedtípus, ahol ugyanaz a tulajdonságtípus azonosító, pedig az 1 oldalt),

2. az összetett kulcsok külső ulcsokból épülnek ki (állnak), és
3. a hierarchikus kulcs legalább egy külső kulcsot tartalmaz.

7.7.ADATMODELL, ADATBÁZIS

Ismereteink a valós világ konkrét jelenségeire (egyed vagy egyed-előfordulás), azok konkrét tulajdonságaira (tulajdonságtípus-előfordulás vagy tulajdonságérték) és a közöttük fennálló konkrét viszonyokra (kapcsolattípus-érték, konkrét kapcsolat) vonatkoznak. Az információs rendszer működésének folyamán az alapvető feladat a konkrét egyedtípusokat (a tulajdonságaikkal együtt) valamint kapcsolataikat leíró adatok összegyűjtése (biztosítása), elhelyezése, kezelése és igények szerinti formában és időben való megjelenítése.

Az információs rendszer fejlesztésének fontos mérföldköve a valós rendszer hibátlan tükröztetése a fogalmi szinten egyedtípusok, azok tulajdonságtípusai, és az egyedtípusok között fennálló generikus viszonyok megfogalmazására szolgáló kapcsolattípusok segítségével.

Az adatmodell az egyedtípusok, tulajdonságtípusok és kapcsolattípusok, valamint a rájuk vonatkozó megszorítások szervezett együttese. Az adatmodell szervezettségét az egyedtípusok intenzióinak (tulajdonságtípusainak) és transzverzióinak (kapcsolattípusainak) mélyreható elemzése biztosítja. Ezeket az elemzéseket az egyedtípusok belső és külső szerkezetének kiépítése folyamán alkalmazzuk, és ezekből erednek a strukturális megszorítások (amelyeket egyéb, más, tulajdonságértékekre, egyed-előfordulásokra, egyed-előfordulás halmazra, sőt, kapcsolatban lévő egyed-előfordulásokra vonatkozó leíró megszorítások egészíthetik ki.).

Az adatalmodell az egyedtípusok, tulajdonságtípusok és kapcsolattípusok, valamint a rájuk vonatkozó megszorítások aránylag önálló (aránylag független) szervezett együttese. A rendszerszemlélet alapján először az adatmodell kerül meghatározásra, kialakításra, és csak utána a viszonylag független részei, az adatalmodellek.

Az adatbázis az egyed-előfordulások, tulajdonságtípus-értékek és kapcsolattípus-értékek adatmodell alapján kialakított szervezett együttese. A meghatározásból is kitűnik, hogy az adatbázis kialakításához elengedhetetlen az előzőleg kidolgozott adatmodell, vagyis előbb az adatbázis absztrakt képét kell megalkotni, és csak utána a konkrét adatokat aszerint elrendezni.

Az információs rendszer egy-egy végfelhasználóját általában csak a rendelkezésre álló adatoknak csak egy sajátos válogatása, vetülete érdekli (pl. mely vevők rendeltek az X nevű áruból, modjuk Y hónapban).

A nézet (view) az adatmodellnek egy olyan absztrakt vetülete, amely egy, a felhasználó által megfogalmazott információs igény kielégítésére szolgál. Segítségével az adatbázisból csak azokat az adatokat kezeljük és jelenítjük meg, amelyek az említett igénnyel kapcsolatosak. A nézet megfogalmazása hármas jellegű: intenzionális, transzverziális és extenzionális. A felhasználót ugyanis csak bizonyos jelenségek (egyedtypusok) meghatározott konkrét tulajdonságai (intenzió) érdeklik, amely tulajdonságokat csak az egyedtypusok közötti kapcsolattypusok (transzverzió) alapján tudunk biztosítani, de azok közül is csak konkrét egyed-előfordulásra (extenzió) vonatkozóan.

A nézet akkor valós, ha a felhasználói kérelem megvalósítható a tervező által létrehozott adatmodell tényezőinek közvetlen felhasználásával. Más szóval ez azt jelenti, hogy a valós nézet az adatbázisból csak az adatmodellben szereplő valós tényezők konkrét értékeinek megjelenítését teszi lehetővé (pl. X dátumra vonatkozó megrendelések, a vevők – megrendelők – adataival; a vevő és a megrendelés valós egyedtypusok, és a közöttük levő kapcsolattypus is valós).

A nézet akkor virtuális, ha a felhasználói kérelem nem valósítható meg a tervező által létrehozott adatmodell tényezőinek közvetlen felhasználásával. A virtuális nézet, tehát, az adatmodellben szereplő valós tényezők mellett más, számítással, szűréssel, stb. létrehozott (nem létező) elemeket is tartalmaz (pl. az X nevű árut rendelő vevők listája, az Y hónapban általuk egyenként megrendelt össz árumennyiség feltüntetésével már virtuális nézetet jelent, mert az „X nevű árut rendelő vevő” egyedtypus nem létezik, sem az X áruból az Y hónapban megrendelt össz mennyiség).

A nézet tehát az össz ismereteknek csupán egy részhalmazát öleli fel, a felhasználói igényeknek megfelelően, és mint olyan, nem létező elemeket is tartalmazhat. Az adatmodell az adatmodellnek olyan valós részhalmaza, amely egy a felhasználói igény kielégítésére irányuló nézet előállítására szolgál.

7.8. ISMERETKEZELÉSI MÓDOK

Ha csupán a klasszikus számítógépes ismeretkezelésre gondolunk (figyelmen kívül hagyva a multimédia bámulatos, és egyre bővülő lehetőségeit: a többfajta grafikai ábrázolást, a hanganyagot, az álló- és mozgóképeket hordozó adatokat), akkor annak két fajtáját ismerjük és műveljük. Ezek pedig a:

1. tényszerű (faktografikus) vagy adatszerű és

2. szövegszerű ismeretkezelés.

A ténytyszerű ismeretkezeléssel (adatkezeléssel) kapcsolatban meg kell említeni, hogy már a görögök a feljegyzésre szánt tényeket (versenyek győztéseit, azok eredményeit, felismert és bizonyított természeti és társadalmi törvényszerűségeket, dalokat, mondákat, hiedelmeket, stb.) rekordoknak nevezték. Ez a fogalom, ha értelmében módosulva is, de fennmaradt mind a mai napig. Ma a világraszóló eredményeket hívjuk rekordoknak, de ugyanakkor a megfigyelt jelenséget leíró adatsornak is ez a neve a számítógépes adatfeldolgozás területén (és nemcsak magyar nyelven). A valamilyen szempontból hasonló jelenségeket leíró adatsorokat általában egymáshoz közel raktározták (regisztrátorok, dossziék) régen is, a dosszié angol megfelelője pedig talán a legelterjedtebb fogalom a számítástechnikában: a fájl (file). A hasonló jelenségeket leíró adatokat tartalmazza a fájl, a tételei pedig a rekordok (amelyek a konkrét jelenségeket írják le). A hasonló jelenségeket elíró adatsor (a tétel vagy rekord) adataira azt mondjuk, hogy azok a mezők. A hierarchikus felépítés tehát: fájl – rekord – mező – mező tartamla. Ezeknek a következő konceptuális szintű fogalmak (fogalmi szintű lényegeket) felelnek meg: egyedtípus – egyed-előfordulás – tulajdonságtípus – tulajdonságérték.

A ténytyszerű ismeret- vagy adatkezelést ma adatfeldolgozásnak hívják, és két alapvető módját ismerjük:

1. fájl- vagy állománykezelés és
2. adatbáziskezelés.

A klasszikus és túlhaladott állománykezelés extenzionális jellegű kezelést jelentett, az egyedtípusok (fájlok) közötti kapcsolatokkal senki sem törődött. Az adatoknak a háttértárolókön való előnyös fizikai elhelyezése volt a legfontosabb feladat a fájlkezelésnél, az adatok értelmével, jelentőségével kapcsolataival senki sem törődött.

Az adatbázis az egyed-előfordulások, tulajdonságértékek és kapcsolattípus-értékek fizikailag természetesen állományokban elhelyezett, de mindenképpen adatmodell szerint szervezett együttese. Az adatbáziskezelés nem zárja ki a klasszikus, állományszerű kezelést (az állományban-táblában való vertikális mozgást, vagyis az extenzionális kezelést), azonban rá elsősorban a transzverziális feldolgozás (a táblák-állományok közötti vízszintes mozgás) a jellemző.

A szövegszerű ismeretkezelést nem szabad összetéveszteni a szövegszerkesztéssel! A szövegszerkesztők szövegírásra, a begépelte szöveg megjelenítésére, és külalaki csinosítására alkalmasak, de adatkezelésre semmiképpen sem. A szövegállományoknak nincsenek rekordjaik, mezőik, szóval az alapvető egyed-tulajdonság-kapcsolat fogalomhármának (és

előfordulásainak) a megfelelője ismeretlen a szövegszerkesztés gyakorlatában, így ismeretkezelésről nem beszélhetünk.

A szövegszerű ismeretkezelés két, ma is alkalmazott formáját a

1. szövegkezelő rendszerek és
2. táblázatkezelők

biztosítják.

A szövegkezelő rendszerek közös megegyezésen alapuló, általános érdeklődésre vonatkozó adatokkal gazdagított (szerző, cím, megjelenés helye, megjelenés időpontja, a témát megjelölő kulcsszavak, stb.) bármilyen tartalmú leírásokat, megfogalmazásokat tartalmazhatnak. Emellett adatszótárral (tezaurusz) is kiegészítik őket, melynek keretében a deskriptorok a szótárban felvett kulcsszavak (előfordulási értékeinek) megjelenési helyét rögzítik, a tulajdonságtípus-tulajdonságérték pároshoz hasonlóan. A szövegkezelő rendszerekben valamilyen ismérv, kulcsszó alapján együtt, „fájlként” kezelhetők a dokumentumok, megvalósítva ezzel a másik adatdimenzió-párost: az egyedtípus-egyed-előfordulás párost. A harmadik páros, a kapcsolattípus-kapcsolatérték azonban hiányzik a szövegkezelő rendszerek megvalósításából, ezért nem tekinthetők adatbáziskezelő rendszereknek is egyúttal. Az ilyen rendszerek becsületes neve adatbank.

Az adatbank olyan igényesen és célszerűen, valamilyen szempont szerint “állományokba” összeválogatott és “fájlok” szintjén is elrendezett dokumentumok együttese, melynek keretein belül a kezelés a “fájlok” tematikájától függetlenül egységes, és bár az ismeretek az “állományok” szintjén átfedőek is lehetnek, tartalmilag “fájl” szinten nem kapcsolhatók össze.

A táblázatkezelők álmukban adatkezelésről álmodnak, sokan tévesen így is hívják őket, de persze attól messze állnak. A fogalomzavar onnan ered, hogy az egyedtípusnak, mint fogalmi szintű lényegnek a logikai szinten a leggyakrabban táblázattal ábrázolt reláció felel meg. A klasszikus táblázat nem ismeri a tulajdonságtípust, így a táblázat bármelyik oszlopának bármelyik elemébe (sorába), tetszőleges érték beírható mindenfajta ellenőrzés nélkül (ami természetesen lehetetlen egy adatbázis esetében). A táblázatban egyedtípusnak megfelelő fogalmak sem léteznek, tehát kapcsolattípusokról sem beszélhetünk.

A multimédiás korszak már rég bekopogtatott az életünkbe, hát magától értetődő a kérdés: vajon hogyan illeszkednek be a képek, az animációk, a grafikonok, a film- és hanganyagok, valamint az esszészerű leírások, stb. a klasszikus adatbázis korlátaiba? Az adatmodell és az adatbázis fogalmának megfogalmazásában csak egyedtípusokról és egyed-előfordulásokról, tulajdonságtípusokról és tulajdonságértékekről, kapcsolattípusokról és kapcsolattípus-értékekről (kapcsolatértékekről) van szó, az elraktározás és a megjelenítés formájáról említés nem esik. Ez

azt jelenti, hogy az adatmodell és az adatbázis szempontjából a rögzítés és a megjelenítés formája invariáns, teljesen lényegtelen. A definíciók nem érintik a fogalmi szintű lényegeket (egyedtypusok, -előfordulások, tulajdonságtypusok, -értékek, kapcsolattypusok és kapcsolattypus-értékek) ábrázolásmódját. A multimédiális adatok más, “közönséges” egyedtypusokhoz kapcsolódó egyedtypusokat jelentenek, pl. a hallgató arcképe a hallgató egyedtypushoz kapcsolódó HALLGATÓ_ARCKÉPE egyedtypusként modellezhető. A hallgatónak több arcképe is lehet elraktározva az adatbázisban, ezekre az azonosítók alapján tudunk egyértelműen hivatkozani, vagyis a HALLGATÓ_ARCKÉPE egyedtypusban a kapcsoló tulajdonságtypus (külső- vagy idegen kulcs) mellett azonosítónak is lennie kell.

A multimédia-elemek klasszikus adtbázissal való házasításának technikai megoldása már nem egyértelmű. Bármilyen fizikai szinten rögzített (becementezett) megoldás hátrányos, hiszen a bővítés és a változáskezelés ilyen esetekben lehetetlen, mert az adatkezelés fizikai és logikai (program) szinten is merev. Sem fizikai sem logikai adatfüggetlenséggel nem lehet dicsekedni.

A modern adatbáziskezelő rendszerek a (főnt felsorolt) fogalmi szintű lényegeket rögzített fizikai tárolási módszerekkel (még ha mindegyik más-más fizikai szerkezetben is, de általában mindegyik többféle tárolási módszert használ), de fogalmi (tartalmi) és logikai szerkezettől függetlenül valósítják meg. Ezért tekintjük őket általánosított adatkezelő rendszereknek. Itt a változáskezelés (tartalmi vagy foglami változtatás – új egyed-, tulajdonság- és kapcsolattypusok hozzáadása) minden hátrányos következmény nélkül, „élőben” is folyamatosan bonyolítható. Ezt a kényelmet a rendszer bonyolultságával kell megfizetni (egyések szerint a mai adatbáziskezelő rendszerek bonyolultabbak az operációs rendszereknél is). A multimédia-adatok kezelése egészen más jellegű, mint a klasszikus adatkezelés, tehát az egyébként is bonyolult adatbáziskezelő rendszert ki kellene bővíteni a multimédia-adatok kezelésével ahhoz, hogy a szó igazi értelmében általánosított (mindenfélre, szélesebb körben elterjedt adatok kezelésére alkalmas) valódi adatbáziskezelő legyen. Mivel ez korántsem egyszerű feladat, teljességgel érthető, hogy a mutimédia szinten általánosnak tekinthető adatkezelők még nem jelentek meg a piacon.

8. AZ INFORMÁCIÓS RENDSZER FELDOLGOZÁSVETÜLETE

Az információs rendszer adatvetülete után most a feldolgozásvetülettel ismerkedünk meg. Nagy a különbség a két vetület között. Az adatvetület önmagában is elemezhető és kiépíthető, a feldolgozásvetület (kialakítása, elemzése) viszont az adatvetület meglétét és ismeretét feltételezi. Az adatok nélkül nem határozhatók meg a feldolgozások. Adatok képezik a feldolgozás tárgyát, és ezzel együtt meg is határozzák (korlátozzák) azt. Az egyetemista átlagos arcképét nem lehet az átlageredményéhez hasonlóan megfogalmazni (meg hát mire is volna jó?), hiszen a képadatok összegezése vagy átlagozása (egyenlőre, tudomásom szerint) nem értelmezett.

A feldolgozások maguk is rendszert alkotnak az információs rendszeren belül, és mind a három absztrakciós szinten megtalálhatók a tényezői az elemi műveletektől kezdve a funkciókon keresztül a rendszerrészekig (alrendszerekig). A feldolgozás elemei kétszeresen is kötődnek az adatokhoz mindhárom absztrakciós szinten, mert egyrészt az információs tevékenységek (a feldolgozásvetület egyik tényezője) adatokra irányulnak, másrészt viszont az információs események (a feldolgozásvetület másik tényezője) ugyancsak adatok képében jelennek meg (a hallgató ideiglenes tartózkodási helyének-címének megváltozása – az új ideiglenes lakcím-adat megjelenése bemeneti információs eseményt ír le). A feldolgozásrendszert tehát együtt kell (vagy legalábbis kellene) elemezni az adatvetülettel valamennyi absztrakciós szinten. A hétköznapi fejlesztési gyakorlatban csak a középső absztrakciós szinten támaszkodik a feldolgozásrendszer az adatvetületre. Ez a szint az adatfolyamdiagramok (AFD, vagy DFD – Data Flow Diagram) szintje. A magasabb (rendszerrész, alrendszer vagy funkciócsoport) és alacsonyabb (elemi műveletek szintje) absztrakciós szinten általában megfelelnek a feldolgozásvetület adatrendszerre utaltságáról.

8.1. AZ ISMERETEK MEGJELENÉSI FORMÁI

Az adatok fizikai elhelyezését a háttértárolókon az operációs rendszer végzi. Kivételt ez alól csak azok az adatbáziskezelő rendszerek képezik, amelyek nem bíznak meg másban, és ezt a feladatot is önállóan, saját maguk valósítják meg. Általában tehát az operációs rendszer feladata a kért (keresett) adatok rendelkezésre bocsátása. Régebben voltak próbálkozások az egyes programnyelvek adatkezelő eljárásainak (rendszerének) megkerülésére, de akkor az alkalmazói programnak kellett gondoskodnia az adatok fizikai elhelyezésének módjáról és a hozzáférésről,

az operációs rendszer csak a háttértárolókon való fizikai adatrögzítést végezte. Az ilyen irányú igyekezetek azonban csakhamar zsákutcába jutottak.

A mai alkalmazások a szabványosított adatkezelési eljárásokat használják úgy az adatok beírásánál, mint visszakeresésénél. Az adatkezelési eljárások az operációs rendszerrel karöltve végzik ezt a funkciót, amely nem észlelhető (transzparens) a felhasználó számára.

Ha az információs rendszer földrajzilag távol eső, különböző részekből áll össze, azt mondjuk rá, hogy elosztott (distributed). Ilyen esetben az adatfeldolgozás csak valamilyen adatátviteli rendszer (data communication system) segítségével valósulhat meg. Az adatátviteli rendszert az adatkezelési eljárások veszik automatikusan igénybe akkor, ha a kért adatok nem a lekérdezést megfogalmazó számítógépen helyezkednek el.

A relációs adatbázisok a múlt század nyolcvanas éveinek elején jelentek meg. Ezek a relációs elven működő adatkezelők a fizikai adatkezelés mellett átvállalták a logikai-fizikai lekpzést is. Ennek aztán az lett a pozitív hozadéka, hogy az addig főleg a fizikai megvalósítással bíbelődő tervezők az adatbázisok logikai tervezése felé fordultak. Ma már az is világos mindenki számára, hogy a fogalmi (konceptuális) tervezés, a fogalmi adatmodell nem léphető át, mert a logikai szint a felhasználók és az eszközök megszorításaitól szenved, vagyis a valós világnak csupán torzított képét nyújtja. A relációs adatbázisok tervezése lehetővé teszi, de nem kényszeríti ki a normalizációnak az alkalmazását a tervezőtől. A relációs fejlesztés alappillére, a reláció, az adatok összetartozását jelölő logikai szintű lényeg. Az adatmodel, ami viszont fogalmi szintű kategória többfajta (hálós, hierarchikus, relációs) adatkezelő rendszerrel (logikai szint) is realizálható.

A séma a modellnek olyan leírása, amely különböző strukturális, szemantikai (tartalmi) és szintaktikai (formai) szabályoknak felel meg.

A séma tehát nem tévesztendő össze a modellel, mert annak csak a reprezentánsa. Ugyanaz a modell, a megbeszélte és elfogadott szabályok függvényében más-más sémához vezet(het), vagyis egyetlen modellnek több sémája is lehet.

Az ANSI/SPARC ajánlatok az adatbázisok elvi felépítésére (architektúrára) vonatkoznak, ezért a köznyelvben csak ANSI/SPARC adatbázis-architektúrának nevezik őket. Az ajánlat szerint az adatbázis leírására háromféle sémát kell kidolgozni. A számítógépen belüli adatszerzés leírását a belső séma tartalmazza. A valós világ adatszerzési változatait (a forrásdokumentumokon, a felhasználó és a tervező elképzeléseit) a külső séma rögzíti.

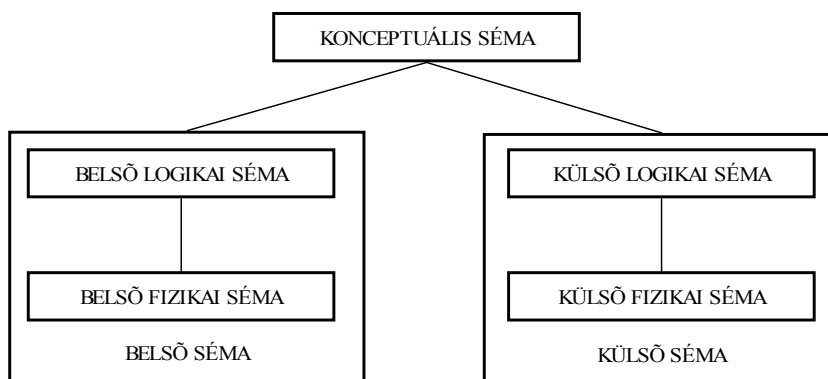
A belső és külső séma összehangolására szolgál a fogalmi séma. Vegyük a rendelés dokumentumot, amelyen a „megrendelt mennyiség” az adatbázisban a „rendmenny” tulajdonságtípusban ölt testet. Ezt a kapcsolatot is, az összes többivel együtt, tartalmaznia kell a

fogalmi sémának. A konceptuális sémában olyan adatok is helyet kaphatnak, amelyek az adatbázisban sohasem jelennek meg (pl. A számított adatok, amelyek a forrásdokumentumokon jelen vannak, az adatbázisba nem kerülnek bele). De van fordított helyzet is: a számítógépen, a bázisban létez(het)nek adatok, amelyek nem találhatók meg a dokumentumokon (mutatók, belső azonosítók, index-szerkezetek, stb.).

A konceptuális séma az adatmodellnek (illetve a fogalmi adatmodell alapkategóriáinak) olyan leírása, amely különböző strukturális, szemantikai (tartalmi) és szintaktikai (formai) szabályoknak felel meg.

A külső és belső séma is két absztrakciós szintet takar: a logikait, amely a tartalmat érinti, és a fizikait, amely az elrendezésre (formára) vonatkozik.

A külső séma a külső adatmodell megfogalmazására szolgáló, a konceptuális sémával harmonizáló, két absztrakciós szintű leírás, amely az absztrakciós szinteknek megfelelően tagolódik: külső logikai sémára és külső fizikai sémára.



8.1. ábra

A belső séma a konceptuális sémával harmonizáló, külső adatmodell rögzítését szolgáló, két absztrakciós szintű leírás, amely ugyancsak két részre (a szinteknek megfelelően) tagolódik: belső logikai séma és belső fizikai séma.

Az előző magyarázat után már nem szükséges külön részletezni a 8.1. ábrán¹ elhelyezkedő, három (pontosabban öt) fajta séma kapcsolódásait (struktúráját).

A belső séma két logikai szintjének megfelelő része a logikai és fizikai adatmodell leírásának felel meg, azaz a logikai és fizikai adatbázistervnek.

¹ Forrás: [11], 213. oldal.

8.2.BEMENET, RÉSZEREDMÉNY ÉS KIMENET

A számítógépen elhelyezett adatok feldolgozásának előfeltétele a géppel való kommunikálás. A felhasználók (emlékezzünk, hogy többfajta felhasználót különböztetünk meg) a következő két alapvető módon kommunikálhatnak a számítógéppel:

1. off-line – a felhasználó nincs közvetlen kapcsolatban a számítógéppel, amikor az a feldolgozást végzi – a feldolgozásra kerülő adatok előzőleg másodlagos adathordozókra kerülnek (szalag, lemez) a feldolgozó géptől függetlenül (off-line adatrögzítés) és
2. on-line – a felhasználó a feldolgozást végző számítógéppel közvetlen kapcsolatban van, és szükség szerint a feldolgozást is befolyásolhatja.

Az adatfeldolgozás tárgyát a bemeneti adatok (bemenet) képezik. A feldolgozás folyamán több közbülső lépés után jön létre a feldolgozás eredménye, a kimeneti adatok (kimenet).

A közbülső lépések a feldolgozásrészeket jelölik, melyeknek ugyancsak vannak eredményei: a részeredmények. Az átmenet (a részeredmény) a feldolgozáson belül megfogható olyan eredmény, amely egy aktivitásnak (feldolgozáson belüli operációnak, tevékenységnek) a kimenetét, ugyanakkor egy másiknak (másoknak) a bemenetét képezi.

Az adatok feldolgozásának két fajtáját műveljük ma is világszerte:

1. kötegelt feldolgozás (batch processing), melynek legkifejezőbb a következő leírása:
bemenet – feldolgozás1 – átmenet1 – feldolgozás2 – átmenet2 – ... kimenet

Ennél a feldolgozásfajtánál össze(kötegbe)gyűjtik a forrásdokumentumokat (pontosabban azok adatait), és a feldolgozás egyes tevékenységeit egymásután (egy menetben) végzik el az így egybegyűjtött adatokon. Az részeredmények (átmenet) átmeneti, rövid idejű rögzítésre kerülnek, majd hamarosan a következő tevékenységet végzik el rajtuk egészen addig, amíg el nem jutnak a kimenetig. Így történik a tömegmértű szolgáltatások és energiafogyasztások számlázása.

2. tranzakció orientált feldolgozás (transaction oriented processing)

Itt a feldolgozás teljes egészében megtörténik minden bemenő adatra külön-külön, nem várják és gyűjtik kötegbe az adatokat. Ilyenek a mai banki szolgáltatásokkal kapcsolatos feldolgozások: az állományváltozás könyvelése azonnali.

A számítógépes adatfeldolgozás alkalmával késleltetésekkel is találkozunk. A feldolgozást leginkább a következő két helyen késleltetik:

1. a bemeneten. A bemeneti késleltetés esetén a feldolgozás az adatok beolvasása, ellenőrzése (előfeldolgozása) és tranzakciós állományba rögzítése után megszakad. A

leolvasott villamos fogyasztásmérők adatai a beolvasás és formai ellenőrzés után a “változások” szokványos nevű állományba kerül (tranzakciós állomány). Ha ez a feldolgozás kötegetlen folytatódik tovább, akkor a további feldolgozásnak az elindító-jelen kívül nincs más bemeneti adata. A “változások” adattár belső bemenetnek számít.

2. a kimeneten. A feldolgozás bevégeztével nem mindig bölcs az azonnali nyomtatás. Több megfontolásból is késleltethető a feldolgozás a kimeneten: ha maga a feldolgozás is feldarabolt a nagyszámú feldolgozandó bemeneti adat miatt (nagy-tömegű adat feldolgozása esetén, ha nem áll rendelkezésre a checkpoint-restart lehetőség, a rendszerbukás miatt az egész feldolgozást meg kell ismételni, okosabb tehát a már feldolgozást feldarabolni). Másik ok lehet ha a feldolgozás és a nyomtatás különböző földrajzi helyen (esetleg több különböző földrajzi helyen) bonyolódik. A mágneses tárolóeszközök szállítása (ha már nincs lehetőség telekommunikációs adatátvitelre, ami ma már szinte hihetetlen) egyszerűbb és könnyebb, mint a jelentéseket tartalmazó papírtömegé. Ilyenkor a kimeneti adatokat átmeneti nyomtatási-kép állományba (print – image file) helyezik el, amely már a nyomtató számára elkészített formában van.

Az adatmodell nem tartalmazhat bemeneti, kimeneti és átmeneti adattükrözéseket. Az okok nagyon egyszerűek, ha tudjuk, hogy:

1. ezek az adatok általában átfedőek,
2. legtöbbször nem fogalmi szintű adatokról van szó, és
3. ezek az adatok nem normalizáltak (rendszerint több kapcsolt egyedtípusból erednek).

8.3.AZ ADATBÁZIS ÉS A FELDOLGOZÁSVETÜLET

Az adatbázis az adatoknak (ismereteknek) a fogalmi modell (konceptuális séma) szerint rendezett és számítógépen tárolt együttese. A könyvtárak kártyanyilvántartásait, bár kétségkívül azok is az ismeretek szervezett együttesét képviselik, nem tekintjük adatbázisoknak.

Az információs rendszerben használt adatokat két nagy csoportba oszthatjuk abból a szempontból, hogy milyen jelenségek milyen jellegű tulajdonságtípusának az értékeit fejezik ki. A jelleg fogalma alatt most a tulajdonságtípusok értékének időbeni tartósságára, illetőleg változékonyságára kell gondolni, amely szerint megkülönböztetünk:

1. törzsadatokat és
2. eseményadatokat.

A törzsadatok a valós világ időben viszonylag tartós (hosszúéletű) jelenségeihez tartozó azon tulajdonságtípus-értékek amelyek viszonylag hosszú ideig nem változnak. A törzsadatok fogalma a nyilvántartás fogalmához van közel, ugyanis általában a nyilvántartásra érdemes jelenségek adatait tekintik törzsadatoknak.

A valós világ nyilvántartásra érdemes jelenségei általában viszonylag hosszú életűek. Ez a megfogalmazás („viszonylagos”) viszont a jelenségek szükségszerű változékonyságát takarja: a keletkezésüket (születésüket), az átalakulásukat, változásukat és a megszűnésüket (halálukat). Ugyanez mondható el a jelenségek információs rendszerbeli türozódéseiről, az egyedekről, amelyekre a fönt leírt fázisokat felölelő megfogalmazást egyed-életciklusnak (entity life-cycle) nevezzük.

Az eseményadatok vagy változások a valós világ időben viszonylag tartós (hosszúéletű) jelenségeinek megváltozását (pontosabban a jelenségek tulajdonságainak változását) leíró adatok.

A törzs- és eseményadatok kétszeresen is kapcsolódnak egymáshoz, amit a következő módon fogalmazhatunk meg:

1. a kétfajta adat az eddigi gyakorlat beidegződése szerint átfedésben vannak, mert például a hallgató nevének változását a hallgató megváltoztatott neve (név – eseményadat, változás) és a változtatást végrehajtó személy neve (név – törzsadat) fémjelezi; ha még ideveszem, hogy a változtatásnál általában a hallgató régi neve (név – törzsadat) is szerepelni szokott (a hallgató egyedtípus megváltozott nevű hallgatójának azonosítóértékével együtt) ellenőrzési meggondolásokból, akkor a kapcsolódás tagadhatatlan,
2. léteznek olyan események is, amelyeknek az adatait az adatbázisban kell tárolnunk (vagyis nemcsak más jelenségek tulajdonságtípusértékeinek változását idézik elő, hanem ők maguk is rögzítésre kerülnek), tehát törzsadattá válnak; ha az eseményadatokból törzsadatok keletkeznek, akkor csonka egyed-életciklusú egyedtípus keletkezik: pl. befizetés a bankszámlára vagy diploma-honosítás (nem fogalmazható meg rájuk az életciklus módosítás szakasza).

A módosítás és a hibajavítás fogalma a mindennapi gyakorlatban gyakran összemosódik, pedig a két fogalom között egyértelmű a különbség. Fogalmi szinten módosításról akkor beszélünk, ha a valós világ változásait vezetjük át az információs rendszer adataira, hibajavításról pedig akkor, ha valamilyen (modjuk) módosítást hibásan hajtottunk végre (tévesen ütöttük be az adatot). A két fogalom téves összevonására (javítás – homonima) az azonos technikai végrehajtásból ered.

Az eseményadatok legtöbbször csak a forrásdokumentumokon (bizonylatokon) jelennek meg, míg a törzsadatokat a számítógépen őrzik.

Az adatfeldolgozás jellege változáson ment át az informatika, de főleg a számítástechnika fejlődésének köszönve. A bemenetekből manapság csak nagyritkán keletkezik azonnal, a feldolgozás végén (egy menetben) kimenet. A bemenetek ma általában beépülnek a törzsadatokba, és ott várják a további (újabb, a valódi) feldolgozást. Ezért a klasszikus feldolgozást kifejező képlet, a

bemenet – feldolgozás – kimenet

a következő kifejezőpárra bomlik

bemenet – feldolgozás1 – adatbázis

és

adatbázis – feldolgozás2 – kimenet.

A kifejezőpár első lánc a karbantartást (maintenance) jelöli, ami alatt a hozzáadást (új egyedek beírása), a naprakész állapotra hozást – aktualizálást (módosítást-update) és a törlést kell érteni. Ezek a tevékenységek nem adnak igazi kimeneteket.

A kifejezőpár második lánc a feldolgozást jelöli, amely kétféle lehet:

1. a szelektív visszakeresést, amely okosan megfogalmazva új ismereteket is nyújthat (az évet veszítő hallgatók listája nem csak a neveket árulja el, hanem azt is, hogy hányan vannak) és
2. új ismeretek előállítását, ami két alapvető módon történhet:
 - az előre megfogalmazott, rutinszerű (gyakran igényelt), mondhatni úgy is, szokványosan szükséges műveletsorok végrehajtásával (a hallgató átlageredménye, amely új ismeret, mert az adatbázisban nem található, de szükség van rá – ha nem is sokszor), és
 - a véletlenszerűen megfogalmazott (tallózásszerű) lekérdezésekkel.

Nem szabad elfelejtenünk, hogy a karbantartás folyamata is igényli a visszakeresést, ugyanúgy, mint a feldolgozásé, viszont a felhasználó szándéka egészen más az egyik, illetve másik esetben.

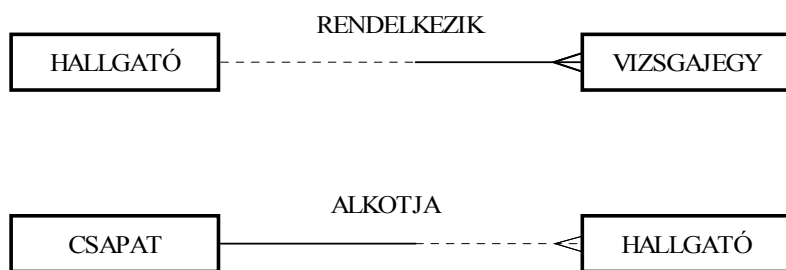
Az adatfeldolgozás (karbantartás és új ismeretek előállítása) megfogalmazása kétféle módon történhet meg (a mindennapi szóhasználatban azonban egyszerűen csak két fajta adatfeldolgozásról beszélnek):

1. procedurálisan (procedurális feldolgozás) – a programozó a programokban saját kezűleg írja a végrehajtási utasítások sorszámozását (procedúrákat), amelyek alapján a feldolgozás megtörténik, és

2. deklaratív (deklaratív feldolgozás) – az adatbáziskezelő rendszer tartalmazza az általános (parametrizált) adatkezelési (karbantartási és új ismeretek előállításához szükséges) eljárásokat, amelyekkel az összes karbantartási és ismeret-előállítási igény megvalósítható a megfelelő paraméter-értékek megadásával.

Az adatbázisban az adatok kapcsolatban állnak egymással. Megtörténhet, hogy egy adat megváltoztatása fogalmi szinten, kapcsolat jelenléte miatt, egy másik adat megváltoztatását is kiváltja (ellenkező esetben az adatbázis inkonzisztens állapotba kerülne). Ha az ilyen fogalmi szinten kapcsolt változások sorának végrehajtását az adatbáziskezelő rendszer átvállalja, akkor azt automatikus átvezetésnek (propagation) hívjuk, ha nem, akkor a programozónak kell gondoskodnia a változások végigvezetéséről. Példának ismét a hallgatót vesszük, akinek az átlageredményét is nyilvántartjuk. Minden új vizsgaosztályzat beírása, vagy vizsga megsemmisítése szükségszerűen az átlageredmény módosítását váltja ki, különben az adatbázis tartalmi ellentmondásával (inkonzisztenciájával) kellene számolnunk.

A mai adatkezelők a kapcsolatok jellegének megadása alapján (persze külön kérésre) átvállalják az automatikus átvezetést a törlésre és az értékváltoztatásra. A 8.2. ábrán az élet egy kissé elferdített képének része található iskolapéldaként a propagáció elvének bemutatására. A kapcsolatok opcionális, illetve kötelező jellege (0 illetve 1 minimális kardinalitásérték) szaggatott illetve folytonos vonallal jelölt. A HALLGATÓ leckeönyvszáma (azonosítója) a kapcsolat a VIZSGAJEGY felé (RENDELKEZIK), a CSAPAT csapatkódja pedig az ALKOTJA



8.2. ábra

kapcsolatot képviseli a HALLGATÓ irányában.

Nézzük milyen automatikus propagálási lehetőséget kínálnak a mai adatbáziskezelők. Ha a főlérendelt egyed típusban megváltozik az azonosító értéke (Az azonosító értékének megváltoztatása szigorúan tilos! Az informatikai halálos bűnök egyikének tartják az azonosító módosítását!), akkor a kulcs-megszorítás tiszteletben tartása mellett a kapcsolódó alárendelt egyed típusokban a külső kulcsok értékét, egészen a hivatkozási mélységig meg kell változtatni (amit az adatkezelő át tud vállalni). Ha az alárendelt egyed típusban változik meg a külső kulcs

(vagy az elsődleges kulcs) értéke, annak nincs kihatása a fölérendelt egyed típusra, viszont a kulcs-megszorításnak és a hivatkozásiépség-megszorításnak (az elsődleges kulcs esetében a kulcs-megszorításnak) eleget kell tenni (itt tehát nincs átvállalható propagáció). Ha a fölérendelt egyed típusban törölünk egy egyedet, az az alárendelt egyed típusban, az alárendelt egyed típus kapcsolattípusának jellegétől függően kétféle következménnyel járhat. Ha az alárendelt egyed típus kapcsolata a fölérendelt felé kötelező, akkor szükségszerűen törlésre kerülnek a fölérendelt egyed típusban törlésre ítélt egyedhez kapcsolódó alárendelt egyed típus-előfordulások. Amennyiben az alárendelt egyed típus kapcsolata a fölérendelt felé opcionális, akkor csak a külső kulcsok értékének törlése szükséges a fölérendelt egyed típusban törlésre ítélt egyedhez kapcsolódó alárendelt egyed típus-előfordulásokban.

A 8.2. ábrán bemutatva az előző magyarázat így hangzik. A HALLGATÓ-VIZSGAJEGY kapcsolatnál adott hallgató leckekönyvszámának megváltozása az adott hallgatóhoz tartozó vizsgajegyek soraiban a külső kulcs (leckekönyvszám) értékének megváltoztatását váltja ki. Ugyanez vonatkozik a CSAPAT-HALLGATÓ kapcsolatra is. A fölérendelt egyed típusban végrehajtott törlés propagálása már különbözik a két kapcsolat esetében. Adott hallgató (fölérendelt egyed) törlése a HALLGATÓ-VIZSGAJEGY kapcsolat esetében a törlendő hallgatóhoz kapcsolódó vizsgajegy-sorok törlését is jelenti egyúttal (az alárendelt kapcsolata a fölérendelt felé kötelező). Egy meghatározott csapat törlése a CSAPAT-HALLGATÓ kapcsolat-szerkezetben viszont csak a törlésre ítélt csapathoz kapcsolódó hallgató-sorok CSAPAT felé mutató kapcsoló-értékeinek (külső kulcsértékek) törlését váltja ki (az alárendelt kapcsolata a fölérendelt felé opcionális).

8.4.A NAVIGÁLÁS KONCEPCIÓJA, LEKÉRDEZÉSEK ÉS JELENTÉSEK

A navigálás koncepciója nagyon régi keletű: “Navigare necesse est, vivere non est.”² Az adatbáziskezelésben ez a fogalom a hálós adatbázisok uralma idején, a hetvenes évek folyamán jelent meg. A navigálás folyamata a kezdetben fizikai szinten történt, ami a fizikai címek program által történő kezelését (rögzítését) jelentette. Minden fogalmi vagy logikai szintű változás, a változásban érintett programok átdolgozásával járt. Ma a logikai szintű navigációt használjuk, ami a fizikaihoz hasonlóan a transzverzális mozgások megvalósítását teszi lehetővé az adatmodell kapcsolattípusainak köszönve..

² Navigálni (hajózni) muszáj, élni nem.

HALLGATÓ

<u>Leckekönyvszám</u>	<u>Hallgató neve</u>	...
E494	Beszédes Ákos	
E495	Néma Levente	
...	...	

TANTÁRGY

<u>Tantárgykód</u>	<u>Tantárgynév</u>
112	Matematika
147	Fizika
174	Elektrotechnika
...	...

SIKERES_VIZSGA

<u>Leckekönyvszám</u>	<u>Tantárgykód</u>	<u>Vizsgadátum</u>	<u>Osztályzat</u>
E494	112	2000.09.12.	9
E494	147	2000.09.15.	7
E494	174	2000.10.05.	6
E495	174	2000.07.14.	8
E495	112	2000.09.15.	8
...	...	...	...

8.3. ábra.

A hallgatók eredményeinek (sikeres vizsgáinak) listázásához a 8.3. ábrán látható adatmodell-részlet szolgál alapul számunkra. A feldolgozás menete a következő: A HALLGATÓ tábla segítségével megkeressük a soron következő hallgatót, és az ott (a táblasorban) található azonosító (leckekönyvszám) értékével a SIKERES_VIZSGA táblában (ahol a leckekönyvszám külső kulcs) rákeresünk az első osztályzatra (ha egyáltalán létezik olyan beírás). Arra a kérdésre, hogy vajon a meglett osztályzat melyik tantárgyra vonatkozik, a következő művelet sor adj meg választ. Az első megtalált osztályzat sorából a tantárgykód (a SIKERES_VIZSGA táblában ugyancsak külső kulcs) értékével felkeressük a TANTÁRGY táblát (ahol a tantárgykód azonosító, vagy elsődleges kulcs) és megkeressük a tantárgy nevét. A tantárgy nevének megtalálása után visszalépünk a SIKERES_VIZSGA táblába, mégpedig arra a sorra, amelyről elhagytuk a táblát, és a további sikeres vizsgák után kutatunk (ha vannak). Ugyanez a helyzet amikor a hallgató utolsó sikeres vizsgájának megtalált neve után a SIKERES_VIZSGA táblába való visszalépést a HALLGATÓ táblába való visszalépés kell, hogy kövesse, mégpedig ugyanarra a rekordra, amelyről elhagytuk a HALLGATÓ táblát. Minden egyedtípus elhagyása előtt (az utolsót kivéve) meg kell jegyezni a helyet ahonnan kiléptünk, hogy majd folytatni tudjuk a feldolgozást (ez a feldolgozás sikeres befejezésének szükséges feltétele).

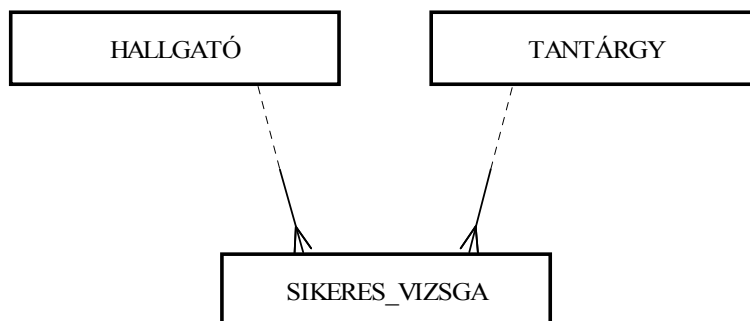
A navigálás természetesen megoldható procedurálisan is, de már jó néhány éve szinte kizárólag csak a navigálás deklaratív megfogalmazását használják az adatbáziskezelésben. Ezt a fajta megfogalmazást teszi lehetővé az SQL nyelv (Structured Query Language – strukturált lekérdezőnyelv), amit már a neve is sejtet. A lekérdezéseket tehát SQL-ben, a jelentéseket pedig

az SQL-re épülő, vagy vele rokon jelentés-készítő (report-writer) segítségével fogalmazzák meg. Mindkét segédeszköz a kérelmek deklaratív leírását teszi lehetővé, megszabadítva ezzel a programozókat az utasítások garmadájának írásától.

A lekérdezést a SELECT mondat tartalmazza, ahol tulajdonképpen paraméterezhető az előre elkészített általános SELECT-mondat. Ugyanezt a mondatot egy, a képernyőn megjelenő táblázat kitöltésével is lehet paraméterezni az adatkezelők többségénél, sőt, váltogatni is lehet egy SELECT megalkotásánál a táblázatos és a SELECT-mondat szerkesztővel történő kialakítása között. A SELECT mondat rögzíti, hogy mely egyedtípusokban előforduló milyen tulajdonságtípusok értékeit kívánjuk megjeleníteni melyik kimeneti egységen és milyen feltételek mellett, esetleg milyen egyszerű függvényeket alkalmazva a tulajdonságtípusokra, milyen sorrendben és alakban az adatbáziskezelő-rendszer segítségével.

A jelentés-készítő a jelentéshez szükséges adatokat ugyanúg lekérdezés segítségével valósítja meg, mint az SQL, csak egy a megjelenítési formájában paraméterekkel alakítható (módosítható) eredménytáblát biztosít kényelmes csoportosításokkal és csoportösszegezésekkel.

Tegyük fel, hogy a 8.3. ábrán látható modell-részlet felhasználásával egy meghatározott hallgatónak a sikeres vizsgáira vagyunk kíváncsiak a beírt érdemjeggyel együtt. A 8.3. ábrán lévő példa logikai adatmodellje a 8.4. ábrán látható.



8.4. ábra

Általában véve a SELECT mondat hármas jellegű: extenzionális, intenzionális és transzverzális. Extenzionális, mert az egyed-előfordulások közül csak a WHERE feltételnek megfelelőket veszi be az eredménytáblába (szelektál), itenzionális, mert csak a feldolgozás szempontjából lényeges (a SELECT szó után felsorolt) tulajdonságtípus-értékek jelennek meg az eredménytáblában (projekció-kivetítés), és transverzális, mert a több (a FRM szó után felsorolt) egyedtípusból eredő tulajdonságtípus-értékek biztosítása csak a transzverzális jellegnek, a kapcsolattípusok létezésének, és az erre épülő lekérdezéseknek köszönhető. A projekció mint fogalmi művelet így épül be a SELECT mondatba.

A következő példa egy adott leckeönyvszámú (495) hallgató sikeres vizsgáit listázza, a tantárgykód és az osztályzat megjelentetésével (a leckeönyvszám is megjelenik az eredménytáblában):

```
SELECT leckeönyvszám, tantárgykód, osztályzat
FROM sikeres_vizsga
WHERE leckeönyvszám=495;
```

Az előző egytáblás (egy egyedtípusra vonatkozó) lekérdezés tehát szelektál a SIKERES_VIZSGA egyedtípus egyedelőfordulásai közül (extenzióra vonatkozó művelet) a WHERE utáni feltétel alapján, de a kiválasztott sorokból (egyed-előfordulás tulajdonságértékei közül) csak a projekció részben rögzített (a SELECT szó után következő) tulajdonságtípusok (intenzióra vonatkozó művelet) értékeit jeleníti meg.

A többtáblás lekérdezések transzverzális jellegűek, hiszen a táblák közötti vízszintes mozgásra épülnek. A többtáblás lekérdezések transzverzális jellege kényszeríti ki a navigációt (tehát navigálni valóban muszáj...) az egyedtípusok (a táblák) között. A transzverzió megvalósítására három fajta lehetőség mutatkozik:

1. többszörös select,
2. kombinált select és
3. összekapcsolás (join operáció).

A többszörös select több, összefüggő, egytáblás select megfogalmazásából áll. Az fent leírt lekérdezés után például lekérdezhető a hallgató neve a HALLGATÓ egyedtípusból (a leckeönyvszám értékének alapján), majd az első lekérdezés eredménye alapján ismertté vált tantárgykódok alapján kikereshetőek a tantárgynevek is (a TANTÁRGY egyedtípusból), amelyeket az adott leckeönyvszámú hallgató sikeresen levizsgázott.

A 8.4. ábrán lévő logikai adatmodellre megfogalmazott kombinált select mindhárom egyedtípust érinti (tartalmazza), összekapcsolás alkalmazása nélkül:

```
SELECT leckeönyvszám, hallgató_neve, tantárgynév, osztályzat
FROM hallgató, sikeres_vizsga, tantárgy
WHERE hallgató.leckeönyvszám =495 AND
      hallgató.leckeönyvszám=sikeres_vizsga.leckeönyvszám AND
      sikeres_vizsga.tantárgykód=tantárgy.tantárgykód;
```

Az SQL-megvalósítások (az egyes SQL nyelvek) szintaktikája eltérő lehet, de az alapelv ugyanaz.

A transzverzió harmadik fajta megvalósítása két vagy több egyedtypusnak a kapcsolattípus vagy kapcsolattípusok alapján történő új egyedtypusba való „összekovácsolásából” áll. Például az előző ábrán látható logikai adatmodell esetében az összekapcsolás (join) művelete segítségével egy SIKERES_VIZSGA_TANTÁRGYNEVVEL elnevezésű egyedtypust lehet létrehozni, amely tulajdonképpen a SIKERES_VIZSGA egyedtypust az egyedelőfordulások minden sorában a sikeresen vizsgázott tantárgynevekkel bővíti. A join művelet nagyon lassú, mert köteget feldolgozást takar (az egyik egyedtypus minden egyed-előfordulása egy új tulajdonságtípussal bővül, amelybe az érték a kapcsoló – kulcs-külső kulcs – vagy közös tulajdonság értékazonossága alapján kerülnek), függetlenül a feldolgozás igényeitől. Más szóval a join művelet a teljes SIKERES_VIZSGA_TANTÁRGYNEVVEL táblát létrehozza, annak elenére, hogy csak egyetlen előfordulására vagyunk kíváncsiak (ezért mondja Halassy Béla, hogy a tranzakció-orientált feldolgozáson belüli join alkalmazása közel halálos bűn).

A feldolgozás során ügyelni kell a navigálás irányára is, hiszen ha a fölérendelt felől haladunk az alárendelt felé, akkor több találattal kell számolnunk az 1:N kapcsolati fok miatt, míg fordítva mindig csak egy előfordulást találunk.

A tranzakciós (tranzakció-orientált) feldolgozás minden tételre egyenként végrehajtja az összes feldolgozási tevékenységet (végighaladva az egész adatbázison, ha a feldolgozás úgy kívánja). A sikeres vizsgát tett egyetemi hallgatók listájának elkészítésénél a feldolgozás veszi a soron következő hallgatót a HALLGATÓ egydtypusból (táblából), majd megjegyezve a leckekönyvszámát és a helyet ahol elhagyja a HALLGATÓ táblát, a SIKERES_VIZSGA táblához fordul, ahol több előfordulást találhat (egy hallgató több vizsgaakadályt vett sikeresen). Lehorgonyoz az első sikeres vizsgánál, megjegyzi a tantárgykódot és a helyet ahol elhagyja ezt a táblát a TANTÁRGY tábla kedvéért, ahol már csak egy előfordulást találhat, és a sikeresen vizsgázott tárgy nevét tudhatja meg. Ezután a SIKERES_VIZSGA táblába tér vissza arra a helyre ahonnan azt elhagyta, hogy a hallgató következő sikeres vizsgájára térjen (ha van ilyen), aztán ismét a TANTÁRGY tábla következik, majd vissza a SIKERES_VIZSGA táblába egészen addig, amíg nem talál több sikeresen letett vizsgát az aktuális hallgatóra. Ekkor visszatér a HALLGATÓ táblába arra a helyre ahonnan azt elhagyta, és rátér a következő hallgatóra (ha van olyan), és a fent leírt ciklus megismétlődik. A feldolgozásban a fölérendelt felől az alárendelt irányába történő mozgás (navigálás) mindenképpen ciklusszervezést igényel akár kombinált selectel, akár összekapcsolással oldjuk meg, és csupán a ciklus megszervezésében van

különbség: éppen a ciklus megszervezésének a módja dönti el, hogy a feldolgozás tranzakció-orientált-e vagy kötelgelt.

A SIKERES_VIZSGA_TANTÁRGYNEVVEL tábla létrehozása egy megszakítást jelent a feldolgozásban, vagyis nem hajtjuk végre az összes feldolgozási tevékenységet minden tételre egyenként, hanem a join művelettel, kötelgelt feldolgozással átmeneti táblát hozunk létre.

8.5.FELDOLGOZÁSTERVEZÉS

A feldolgozás folyamata tevékenységek és események lácolatát öleli fel. A tevékenységeket események váltják ki, a tevékenységek befejezése pedig többfajta eseményt is jelenthet. Egyes tevékenységek többfajta eseményt is elindíthatnak, mint ahogy a tevékenységek vége (befejező szakasza) egyszerre több eseményt is kiválthat. A fenti érvelés arra mutat rá, hogy általános esetben a tevékenységek és az események hálót alkotnak. A tevékenységek tárgya az adat (adatok), a tevékenységeket kiváltó illetve követő események pedig maguk is adatokkal jellemezhetők. A feldolgozást a bemenet (a bemeneti esemény) indítja el, és, ahogy ezt a 8.3. pontban láthattuk, a feldolgozás nem mindig végződik valódi kimenettel (jelentéssel). A tervező feladata, hogy az ismert bemenet, az adatmodell, és a megfogalmazott kimenet (jelentés) alapján olyan feldolgozás-hálót alkosson meg, amelyben a tevékenységek és események logikai és időbeli sorrendje a bemenetből az adatmodell alapján létrehozza a kimenetet. A feldolgozás ideje alatt a bemeneti adatok egy része beépül az adatbázisba, másik része a feldolgozás logikai és időbeli sorrendjére lesz kihatással, és lehetnek olyan adatok is, amelyek megjelennek a jelentéseken is. Ezt az adatvándorlást a bemenetről a kimenet felé a legtöbb esetben sajnos csak a középső absztrakciós szinten mérlegelik és ábrázolják. A bajnak a forrása az előző mondatban szereplő csak szócska. A középső szint feletti és alatti szintekről ugyanis nem volna szabad megfeledkezni. A következőkben az adatfeldolgozás három absztrakciós szintjével ismerkedünk meg.

8.5.1. A feldolgozás modellezése

A mai információs rendszerek tervezése (és fejlesztése is) a top – down megközelítési elv szerint történik. Ez a szemléletet a strukturált programozás és tervezés hagyta ránk, és

általánosan elfogadott (úgy tűnik mérföldkövé alakult) elvvé vált a feldolgozástervezésben is. A strukturált tervezésben ehhez a hozzáálláshoz elválaszthatatlanul hozzátartozik a fekete doboz (black box) fogalma, amely az egész és a rész viszonyát mintázza a következő módon. A tervezés elején csak a rendszer, az egész az ismert, a részletek teljes hiányában, így az elején az egészet fekete dobozként ábrázoljuk. A következő menetben az egésznek a részletei kerülnek górcső alá, így a fekete doboz szűrőként kezd, de a részletek között újabb, második szintű fekete dobozok keletkeznek, amelyek majd a következő menetben kerülnek megvilágításra, miközben lassan (a következő menetek folyamán) kirajzolódik a vezérlés fonala (pontosabban hierarchiája). Az ilyen hozzáállásban az adatfeldolgozás és adatkezelés csak a legutolsó menetben, a hierarchia legalsó szintjén kerül sorra.

A mai felfogásunk szerint az adatokat elsődlegességük miatt, és az időbeni állandóságuk miatt, és a vezérlést az adatfeldolgozási műveletek logikai és időrendi sorrendje szabja meg. Az adatmodell az automatizálni kívánt valós rendszer minden egyed-, tulajdonság- és kapcsolattípusát tartalmazza. A megrendelő információigénye, ha teljes, tartalmazza az összes bemenete(ke)t és kimenete(ke)t, ami alapján aztán az adatmodell segítségével megfogalmazható a feldolgozás az egyes egyedtípusok érintésével (a kapcsolattípusoknak köszönve). Ezzel, mivel rendelkezésre áll az adatmodell és az igény pontos megfogalmazása, tehát ismerjük a kiinduló egyedtípust és a többi egyedtípust, melyeket a kapcsolattípusok segítségével be kell járni az információs igény kielégítéséhez, egyúttal adott a vezérlés is, vagyis a (felhasználó konkrét igényét kiszolgáló) funkció makrosztruktúrája. Ez már a konkrét számítógép-program vázát jelenti, amit hasznos dolog lejegyezni (program suit, program skeleton).

A feldolgozás modellezésének lehetőségéhez az a felismerés vezetett, amely az információs tevékenységek és események tipizálhatóságára mutatott rá. Az információs események tipizálhatóságára az az új nézőpont utalt, hogy az események fölfoghatók úgy is, mint az új egyed-előfordulások megjelenései. A bemeneti adatok megjelenése bemeneti eseményeket jelez, amely bemeneti tevékenységeket vált ki, a kimeneti adatok előállítása pedig a kimeneti tevékenységek befejezését jelző kimeneti eseményekkel zárul. A rendelés (egyedtípus) megjelenése felfogható a "rendelés-esemény" eseménytípus előfordulásaként, amely valamilyen elemi műveletet, tevékenységet vagy feldolgozást eredményez.

Az információs tevékenységek teszik lehetővé a rendszer működését. Mivel a műveletek tárgya (bemenete) és eredménye (kimenete) is esemény, illetve valamilyen egyed-előfordulás megjelenése, a műveletek a következő módon is osztályozhatók: az ugyanazon (bemeneti, kimeneti vagy átmeneti) eseménytípusú műveletek ugyanazon (közös) művelettípusba tartoznak,

vagyis vannak bemeneti, kimeneti és átmeneti művelettípusaink. Emellett még valamennyi művelettípust elvégzhetjük kötegelten és tranzakció-orientáltan is.

Az információs tevékenységek műveletek sorából (hálójából) állnak. Az összefüggő tevékenységek funkciókat alkotnak. Egy-egy funkció a felhasználó által megfogalmazott konkrét információigényt elégít ki a kimenetek (az adatok) olyan formában történő tárlásával, amely könnyen és gyorsan értelmezhető a felhasználó számára. Az azonos bemeneti, kimeneti és/vagy átmeneti egyedtípusokkal dolgozó hasonló funkciók funkciócsoportokba (funkciótípusokba, rendszerrészekbe) tömörülnek. Az egyszerűbb rendszerek esetében a funkciók alkotják a rendszerrészeket.

A funkciótípusok, a funkciók, a művelettípusok, a műveletek és az őket összekötő átmeneti események (amelyek egyébként a funkciótípusokat is összekötik) alkotják az információs rendszer funkcionális modelljét (a műveletek vagy feldolgozások konceptuális modelljét, a konceptuális feldolgozásmodellt).

A funkcionális modell (a konceptuális feldolgozás-modell) többféle módon is megvalósítható. A funkcionális modellt megvalósító műveletstruktúra a logikai feldolgozásmodell (a műveletek vagy feldolgozások logikai modellje).

A feldolgozások elvégzéséhez szükséges konkrét műveletek végrehajtása, elvégzése a rendszert működtető hardver- és szoftvereszközöktől függ. A műveletek illetve feldolgozások végrehajtását lehetővé tevő fizikailag is végrehajtható műveletek struktúráját fizikai feldolgozás-modellnek (a műveletek vagy feldolgozások fizikai modelljének) nevezzük.

A konceptuális feldolgozásmodell (a funkcionális modell) sokkal változékonyabb a fogalmi adatmodelltől, mert a funkció és a funkciótípus fogalma a felhasználó/megrendelő által megfogalmazott információs igénytől (bemenetcsoportoktól, jelentéscsoportoktól) függ.

Mivel a funkciók bemenetének és kimenetének alkotói, sőt, a funkciók közötti kapcsolat megteremtői is az adatok, őket tekintjük elsődlegesnek (mint ahogyan ezt már számtalanszor hangsúlyoztuk). Viszont a fogalmi feldolgozásmodell (a funkcionális modell) vissza tud hatni bizonyos esetekben az adatmodellre. Ez a felhasználó igényeinek kielégítését szolgáló jelentések gyors létrehozását lehetővé tevő adatmodell-bővítésekben nyilvánul meg.

8.5.2. Feldolgozástervezés középső szinten

A hetvenes években a feldolgozás leírása mellé (vagy helyett) szemléltető rendszer- és program-folyamatábrákat mellékeltek. A program-folyamatábrák (flow chart) túlon túl

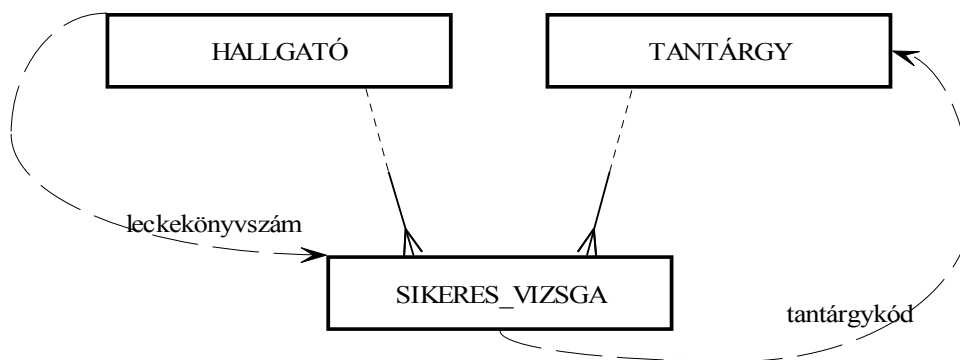
részletesek, és ma már, a negyedik generációs nyelvek elterjedt használatának korában senkinek sincs ideje és türelme bíbelődni velük. A rendszer-folyamatábrákat felváltották más, modernebb, kifejezőbb diagramok, amelyeket a továbbiakban ismertetünk.

8.5.2.1. Adatkezelési gráf

Amint erről már szó volt, az információs rendszer tényezőit összhangba kell hozni a vetületeken (adat-, feldolgozás- és környezetvetület) belül, de más vetületek tényezőivel is. Az adatvetületen belüli homogén harmonizáció (tényezők egyeztetése, összehangolása) adatmodellezés útján érvényesül. Az adatmodellezés önállóan, a másik két vetület tényezőit figyelmen kívül hagyva is elvégezhető, az adatok közötti kapcsolatok időállóságának köszönhetően. Egy felsőoktatási intézményben mindig lesznek oktatók, hallgatók, épületek, előadótermek, tantárgyak, osztályzatok, stb., amelyek között a fennálló kapcsolatok nem a feldolgozás függvényei. A feldolgozásmódel viszont már nem alakítható ki a már kidolgozott adatmodell ismerete és figyelembe vétele nélkül.

A fegyelmezetten és pontosan megfogalmazott információigény lehetővé teszi a megfelelő adatkezelési és adatelőállítási műveletek rögzítését a kívánt cél, a megfelelő jelentés (kimenet) összeállítása érdekében. Ha ismerős az adatelőállítási műveletekhez szükséges számítási műveletek halmaza, valamint az adatok forráshelye (az adatmodell alapján), a programváz gazdagítható az adatelőállítási és adatkezelési műveletek logikai helyével vagy idejével a programban. Ezeknek a gazdagított programvázaknak a létrejöttét segítik az adatkezelési gráfok.

Az adatkezelési gráf az adatmodellnek olyan navigálási útvonallal kiegészített része, amely a felhasználó egy információs igényének a megválaszolására szolgál. A 8.4. ábrán bemutatott adatmodell adatkezelési gráffá alakítva a 8.5. ábrán látható.



8.5. ábra

A leírásból kiderül, hogy egy konkrét adatmodell alapján több adatkezelési gráf is előállítható, és bár az ábráról hiányzanak, de az adatkezelési gráfon az adatelőállítási műveletek is feltüntethetők. Az adatkezelési gráf az elérési út modellezésének legfontosabb eszköze.

8.5.2.2. Adatfolyam diagram

A feldolgozástervezés középső szintjének második hasznos eszköze az adatfolyam diagram (magyar rövidítése AFD, DTP – Dijagram Toka Podataka, DFD – Data Flow Diagram). Az adatáramlási diagram az információs rendszer fontosabb folyamatait, adattárait (tábláit), adatfolyamait és környezeti elemeit tartalmazza (tartalmazhatja még, bár ezt ritkán használják, az anyagfolyamáramlást és az anyagtárolást is). Az adatmodell diagram elvben tartalmazhat vezérlési és adatkezelési műveletek leírását is, viszont az adatelőállítási műveletek nem szerepelnek rajta. Az AFD a feldolgozástervezés középső absztrakciós szintjén megfogalmazott

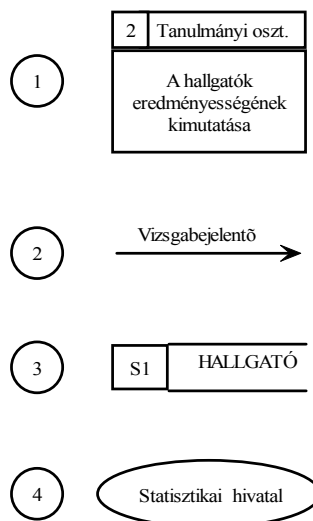
bemenet – feldolgozás1 – adatbázis – feldolgozás2 – kimenet

lánc vizuális megjelenítését nyújtja.

Az adatáramlási diagram elemei (szimbólumai) a 8.6. ábrán láthatók:

1. Folyamat (eljárás),
2. Adatfolyam,
3. Adattár(oló),
4. Környezeti (külső) elem.

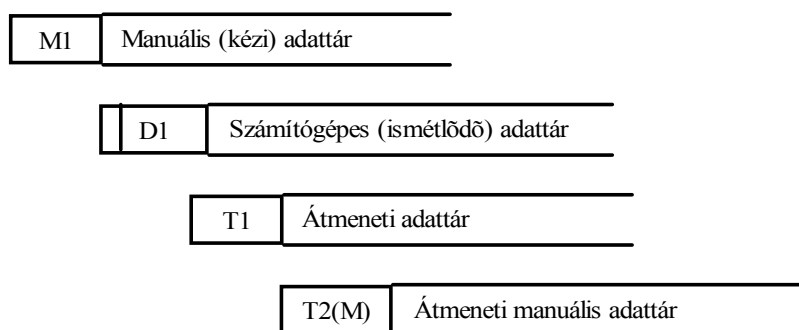
Az eljárás vagy folyamat (tevékenység) jele a három részre osztott téglalap. A folyamatnak van bemenete, amelyet kimenetű alakít. A felosztott téglalap bal felső részében van az eljárás azonosítója. Ez egy puszta szám, amelyet a diagramon általában balról jobbra és felülről lefelé haladva önkényesen rendelnek a folyamatokhoz, vagyis nem a tevékenységek rajzolásának vagy végrehajtásának sorrendiségét jelölik. A téglalap jobb felső részében az eljárást végrehajtó szervezeti egység neve (esetleg a munkát végző személy munkabeosztása) szerepel. A téglalapnak ez a része általában csak az AFD első vázlatán tartalmaz beírást, és az ilyen AFD-ket fizikai AFD-knek nevezik (í fizikai adatáramlási ábra anyagfolyamot is tartalmazhat), a logikai AFD-ken ez a rovat üres marad. A tevékenység, a folyamat elnevezése a téglalap középső, legtágasabb részébe írandó, és



szükségszerűen valamilyen műveletnek a rövid megnevezését foglalja magában. A semmitmondó, általános megnevezések kerülendőek, a folyamat tartalmának szabatos, lakonikus leírása pedig az adatáramlási diagram mellékleteként (és nem a diagramon) fogalmazható meg.

Az adatfolyam az AFD-n nyíllal ábrázolható, és adatokat szállító csatornákat jelképez, amiért egyes szakkönyvek adat-csővezetéknek (data pipeline) is hívják őket. Az áramlás iránya a nyílazás irányának felel meg. Az adatfolyamok, ha csak lehet egyedi, beszédes nevet kell adni, és azt a nyílra (nyíl közelébe) írni. Az adatfolyamok részletes leírását külön dokumentum tartalmazza. Egyes magától értetődő helyzetekben az adatfolyam neve elhagyható: pl. adattárkarbantartás esetében. A már működő rendszerek leírásánál gyakran alkalmazzák az adatfolyamok nevéként a használatban lévő dokumentumok neveit.

Az adattár(oló) az adatok időleges (átmeneti, rövid idejű) vagy tartós (hosszú idejű) tárolását teszi lehetővé. Az adattárak azonosítói betűből és szám(ok)ból állnak: a "T"-vel kezdődő azonosítók átmeneti adattárat jelölnek, a fizikai AFD-ken az "M"-mel kezdődő azonosítók a kézi (manuális) adattárakat azonosítják, a "D"-vel kezdődőek pedig a számítógépeseket; a logikai AFD-ken "L"-lel kezdődő adattár-azonosítók is előfordulhatnak. A leggyakrabban előforduló adattár-jelöléseket és magyarázataikat a 8.7. ábra mutatja.



S1. 8.7.

A környezeti (külső) elemek azért kerülnek ábrázolásra az adatáramlási diagramon, mert valamilyen hatással vannak az ábrázolt rendszerre: vagy adatot szolgáltatnak, vagy adatot igényelnek a vizsgált rendszertől. A környezeti elem egyidőben adatigénylőként és adatszolgáltatóként is felléphet a rendszerrel szemben. A környezeti elem jele az ellipszis, és passzívnak tekintik abban az esetben, ha adatot csak igényel az ábrázolt rendszertől. A külső elem ismételt ábrázolása ugyanazon a diagramon áttekinthetőbbé teheti azt, ilyenkor a szimbólumba rajzolt ferde vonal jelöli a többszörözést.

A fent ismertetett AFD-elemek között az adatfolyamok jelentik a kapcsolatot, de nem minden kapcsolat megengedett. A 8.1. táblázat az elemek összekapcsolhatóságát tartalmazza. Azok az elemek összekapcsolhatóak, amelyekre a hozzájuk tartozó sor és oszlop

metszéspontjában található cella az „**IGEN**” beírást tartalmazza, a „nem” tartalmú cella pedig a hozzá tartozó elemek összekapcsolásának a tiltását jelenti.

8.1. táblázat

	Környezeti elem	Folyamat	Adattár
Környezeti elem	nem	IGEN	nem
Folyamat	IGEN	IGEN	IGEN
Adattár	nem	IGEN	nem

Az adatáramlási diagram (AFD) készítésénél is érvényes a top-down megközelítés, vagyis először az összefoglaló, átfogó (áttekintő) jellegű diagram készül el, a részletekbe bocsátkozó, az áttekintő diagramon feltüntetett funkciókat boncolgató (részletező), úgynevezett alacsonyabb szintű diagramok csak azután kerülnek kidolgozásra. A többszintes AFD azért is szükséges, mert az ajánlások szerint az egy lapon feltüntetett folyamatok száma nem haladhatja meg a hetet (mások szerint a tizet), mert az már túlsúfolt AFD diagramokhoz vezetne. Az áttekintő (1. szintű) diagram fölé rajzolható még egy ún. kapcsolatdiagram, amelyet kontextus diagramnak is szokás nevezni. Ez a 0. szintű diagram az egész információs rendszert egyetlen folyamatként ábrázolja, feltüntetve a hozzá kapcsolódó környezeti elemeket.

Akár az áttekintő, akár a kontextus diagram alapjául szolgálhat a későbbi bontásnak (dekompozíciónak). Az alacsonyabb szintű diagramokon a magasabb szintű diagramok folyamatai kerülnek feldolgozásra, részletezésre. Az 1. szintű AFD a rendszerrészeket (funkciócsoportokat, a kisebb rendszerek esetében a funkciókat) ábrázolja, és rendszerint mellőzi az adattárolók feltüntetését (kivételt csak akkor tesznek ha egy fontos adattárat legalább két rendszerrész használ). Az első szinten az eljárások (folyamatok) egyszámjegyű számokat kapnak foyamatazonosítóként, a második szinten pedig annak az első szintű eljárásnak az azonosítóját veszik alapul, amelyre a részletezés vonatkozik: az első szintű folyamat azonsítójának decimális rendszerű alátörésével állítják elő az azonosítót. Ha a második szintű AFD a 3 jelzésű rendszerrészt taglalja, akkor az eljárások azonosítói 3.1, 3.2, 3.3, stb. lesznek. Ugyanez a logika érvényesül az eljárás-azonosítók meghatározására a további alacsonyabb szinteken is. A szintek számát illetően az a kialakult vélemény, hogy a két-három szintes AFD leírás a legtöbb rendszer számára elegendő.

Az alacsonyabb szintű diagramok diagramok rajzolásakor, a rendszer jobb ismeretének birtokában kiderülhet, hogy a felsőbb szinten valamit helytelenül ábrázoltak. Ekkor a felsőbb szintű AFD módosítását kell először elvégezni, és csak utána folytatni az alacsonyabb szintű AFD kidolgozását, a különböző szintű diagramok közötti ellentmondás elkerülése céljából. Az

alacsonyabb szintű adatfolyam diagramok bemeneti és kimeneti adatfolyamai meg kell, hogy egyezzenek annak a magasabb szintű eljárásnak a bemeneti és kimeneti adatfolyamaival, amelyiket a szóban forgó adatfolyam diagram részletez (ezt a megkötést egyensúly-szabálynak is nevezik). Az adatfolyam nem ágazhat el! Ha egy többpéldányos dokumentum (bizonylat) különböző példányai más-más szervezeti egységbe vagy különböző munkaköröket végző személyekhez kell, hogy kerüljenek, a címzetteket egymástól elkülönítve, számukra külön adatfolyam-nyilat biztosítva kell ábrázolni. Hasonló a helyzet, ha a többpéldányos dokumentum példányai más-más forrásból erednek, csak itt a feladókat és a tőlük eredő adatfolyamot kell elkülönülten ábrázolni. Az AFD-k nem tartalmazzak döntési szimbólumokat, mert a döntéshozatalt a folyamat (eljárás) részének tekintjük, vagy külön folyamatot fogalmazzunk meg rá (a döntésre v. döntéshozatalra). Az AFD-k egyik fő erénye az áttekinthetőség, a fontosabb folyamatok felesleges részletektől mentes ábrázolása.

Az AFD hierarchiaszintjeinek számára vonatkozó szabály nincs. Tapasztalatok és vélemények vannak (az ajánlott szintek száma 2-3), de a szükséges szintek számát az elemi szintű folyamatok határozzák meg. Az eljárás akkor elemi szintű, ha a leírása nem igényel több helyet egy A4-es oldal felénél, és az adatfolyamainak a valóságban is van megfelelőjük (a további bontás pedig mesterkélten elnevezésű, valóságfedezet nélküli adatfolyamokat eredményezne). A diagramokon az elemi szintű eljárásokat a folyamat szimbólumának jobb alsó sarkába helyezett csillaggal jelöljük. Az AFD hierarchiaszintjeinek számát egy meglévő információs rendszer áttérzésénél az áttervezést kiváltó probléma jellege határozza meg: az AFD olyan szintű részletekig készül, amelyen ábrázolhatók a helytelen működés okai (okozói). Az elemi (szintű) folyamatok bontása nem ajánlott, de a leírásaik (mi és hogyan játszódik az elemi folyamatokban) csatolása az adatfolyam diagramokhoz, mellékletként kötelező (a többi eljárás rövid leírása is ajánlott).

Az adatáramlási diagram rajzolása vagy a környezetet feüntető kontextus diagrammal vagy a redszerrészeket ábrázoló áttekintő (első szintű) diagrammal kezdődik. Az AFD kialakításának kezdetén a tervezőknek megkülönböztetett figyelemet kell fordítaniuk a³:

1. rendszer hatáira,
2. környezeti (külső) elemekre (amelyek bemeneteket biztosítanak a rendszer számára és/vagy a rendszer által létrehozott jelentések egyikére-másikára igényt tartanak),
3. legfontosabb bemeneti és kimeneti adatfolyamokra (ezek a rendszer és az őt körülvevő környezet kölcsönhatását tükrözik, képviselik, fogalmazzák meg) és
4. vizsgált rendszer funkciócsoportjaira (rendszerreszeire, funkcióira).

³ Forrás:[2], 89. oldal.

Az AFD alacsonyabb szintjeinek kidolgozása a következő, újabb tevékenységek elé állítja a tervezőket a feladat sikeres és hatékony megoldása érdekében⁴:

1. a rendszerben használatos összes bizonylat (dokumentum, nyomtatvány) összegyűjtése a létrehozó/küldő szervezeti egység(ek)/ a létrehozó-küldő személy munkabeosztása(i) és a címzettek megjelölésével,
2. a rendszer szempontjából legfontosabb bizonylatok kiválasztása és azok bizonylatjegyzékbe vétele,
3. dokumentumáramlási diagram kidolgozása (a szervezeti egységek vagy a dokumentum kezeléséért felelős személy(ek) munkabeosztása(i) alkotják a csomópontokat, az áramló bizonylatok pedig a nyilakat),
4. a bizonylatáramlási diagramon a vizsgálandó rendszer határainak a meghúzása (berajzolása) a megrendelő bevonásával,
5. a határon belüli szervezeti egységek/munkabeosztások eljárásokkal való helyettesítése, az esetleg később szükségessé váló további adatáramlás feltűntetésével,
6. az így kapott AFD ellenőrzése a teljesség, a pontosság és az egyértelműség szempontjából,
7. ellenőrzés és a felhasználóval való elfogadtatás a szükséges javítások elvégzésével és
8. az így kapott diagram szükség szerinti további lebontása (a 6., 7. és 8. pontok ismételt végrehajtásával) egészen az elemi szintű eljárásokig.

A tervező(k) és a felhasználók szoros együttműködését nem lehet elégszer kihangsúlyozni: a dokumentumáramlási diagram és az AFD kialakításának folyamán az együttműködés folytonos egyeztetést tesz lehetővé, ami nagyon fontos a hibák korai javításának szempontjából.

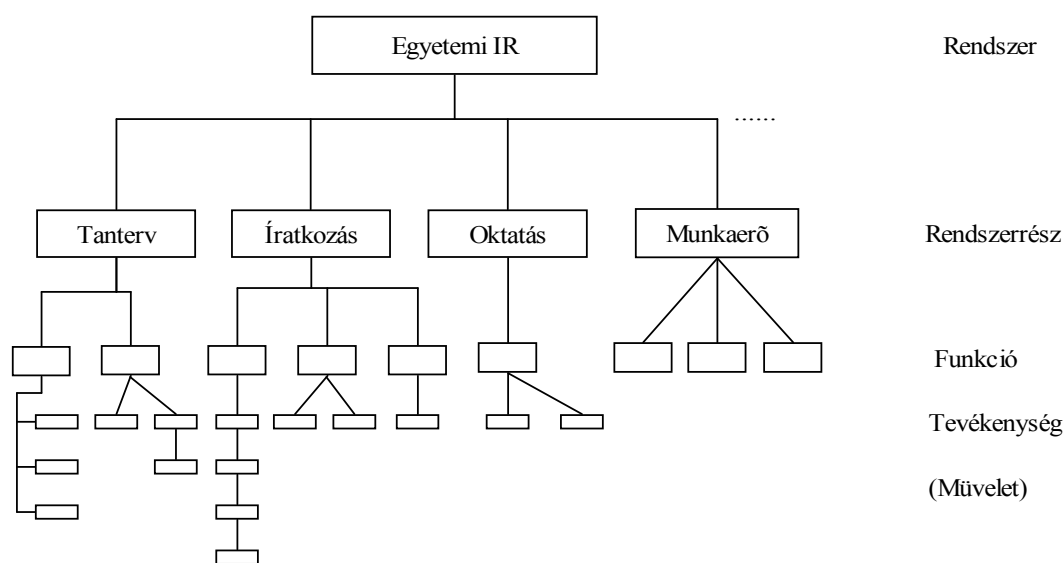
8.5.3. Makrotervezés

Az előzőekben a feldolgozástervezés modellezésének középső szintjével ismerkedtünk meg. A két általunk legfontosabbnak tartott, középső szintű modellezés-technikát ismertettük: az adatkezelési gráfot és az adatfolyam-diagramot. A valós rendszer rendszer-szemléletű megközelítése azonban a fentről-lefelé elvet hangsúlyozza. Az első lépést fentről a makrotervezés irányítja, és a vizsgált rendszer rendszerrészekre bontásában állapítja meg. A rendszerrészek olyan fő funkciócsoportok (kisebb rendszerek esetében funkciók), amelyek hasonló, összetartozó funkciókat jelölnek. A funkciók tevékenységekből állnak, a tevékenységek

⁴ Forrás:[2], 89. oldal.

pedig műveletekből. Így az egész funkcionális modell a következő, különböző bonyolultságú „aktivitásokból” áll: rendszer, rendszerrész, funkció, tevékenység (és művelet). A tevékenységek („aktivitások”) vertikális absztrakcióját a 8.8. ábra⁵ szemlélteti.

Az információs rendszer feldolgozásrendszerének (a funkcionális modell architektúrájának) kialakításában még ma is kivételesen fontos szerepet játszhat(na, ha nem feledkeztek volna meg róla oly sokan) a BSP (Business System Planning-vállalati rendszertervezés) technikának „Az információs rendszer architektúrájának kialakítása” nevű módszere.



8.8. ábra

A BSP-t az IBM fejlesztette ki az információs rendszerek fejlesztésének tervezésére és nyomonkövetésére. A BSP alkalmazását az információs rendszerek fejlesztése területén gyakran nevezik stratégiai tervezésnek is. A BSP technika fent megnevezett módszere témaköröket érinti:

1. az információs rendszer alaparchitektúrájának (általános felépítésének, struktúrájának) kialakítása,
2. a főbb rendszerrészek és az azok közötti kapcsolatok meghatározása és
3. a lépcsőzetes megvalósítás megtervezése és ütemezése.

A BSP technika (egyesek módszernek, sőt módszertannak is nevezik) leírása sok, információs rendszerek tervezésével foglalkozó könyvben megtalálható.

A BSP alapelvei ma is aktuálisak:

⁵ Forrás: [11], 62. oldal.

1. „Az információs rendszernek a megrendelő cég ügyviteli céljait és fejlődésének irányelveit kell támogatnia az ügyviteli stratégiának az információs rendszer stratégiájába való beépítésével,
2. Az információs rendszernek a vezetés, a tervezés és az ellenőrzés valamennyi szintjén megfogalmazott információs igényt ki kell elégítenie,
3. Az információs rendszer feladata, hogy következetes (tartalmi ellentmondásoktól mentes) adatokat biztosítson,
4. Az információs rendszernek biztosítani kell a hosszútávú fejlődés (fejlesztés) lehetőségét a vezetési és szervezési változásoktól függetlenül,
5. Az összes információs igényt kielégítő információs rendszer fázisonkénti megvalósításának lehetősége alapelvárás.”⁶

Maga a BSP technika vagy módszer a következő lépésekből épül fel:

1. A tanulmány (stúdium, módszertan) alkalmazásának jóváhagyása a megrendelő által,
2. Előkészület a tanulmány alkalmazására,
3. Az első egyeztetési találkozó-sorozat a módszertan alkalmazásával kapcsolatosan,
4. Az ügyviteli (tevékenység) modell meghatározása,
5. A főbb adatszoportok (adatosztályok) kialakítása,
6. A meglévő (jelenlegi) információs rendszer elemzése,
7. Az előző lépésekkel kapcsolatos vélemények begyűjtése interjú segítségével,
8. Az interjú értékelése, az interjúkban megfogalmazott problémák elemzése, következtetések levonása,
9. Az információs rendszer architektúrájának kialakítása,
10. Az információs rendszer megvalósításának fázisokra osztása és a fázisok ütemezése (fontossági sorrendjének kialakítása),
11. Az információs rendszer megvalósítási tervének kialakítása és
12. Jelentéskészítés és a stratégiai terv (a BSP project) bemutatása (prezentálása).

Az információs rendszer alrendszerének meghatározása szempontjából a BSP technika negyedik, ötödik és kilencedik lépése a mérvadó: a tevékenységmodell, az adatosztályok és az információs rendszer architektúrájának a meghatározása.

A tevékenységmodell (ügyviteli folyamatmodell) a főbb tevékenységek (folyamatok) és döntések logikailag összefüggő, szervezett halmaza, amelyeket a szervezet legfontosabb céljainak elérése érdekében hajtanak végre az adott vállalatnál.

A BSP technika a folyamatokat (tevékenységeket) három csoportra osztja:

⁶ Forrás: [7.], 26. oldal, a szerző fordításában.

1. a tervezés és vezetés folyamatai (tevékenységei),
2. a termelés és/vagy szolgáltatás tevékenységei (folyamatai) és
3. segédfolyamatok.

A tevékenységmodell kialakításánál a legfontosabb termék(ek) illetve szolgáltatás(ok) meghatározása a cél. Ez a szervezet ügyviteli célkitűzéseinek (célja, célhierarchiája) elemzése alapján dönthető el (határozható meg). A tevékenységmodell folyamatai a termék illetve szolgáltatás következő négy életciklus-fázisa alapján fogalmazhatók meg:

1. tervezés,
2. technológia és termelés,
3. termék (szolgáltatás) ellenőrzés és
4. cikluszárás (értékelés).

A három csoportba osztható ügyviteli folyamatok összegyűjtése után a tevékenységek elemzése és négytől tizenkét tevékenységcsoportba való összevonása következik.

Az adatosztály fogalma nagyon közeli az egyedtípus fogalmához (a gyakorlatban az egyedtípust adatosztályként kezeljük). Az adatosztályok halmaza az információs rendszer adatvetületét jelöli, amiről az előző fejezetben volt szó. Az adatosztályok kialakítása a következő két elgondolás alapján fogalmazhatók meg:

1. az adatok, amelyek ugyanazt az „ügyviteli” egyedtípust írják le, ugyanabba az adatosztályba tartoznak és
2. az ügyviteli folyamatok elemzésével olyan logikai kapcsolatok fogalmazódhatnak meg, amelyeket adatokkal szükséges leírni, így ezek külön adatosztályt jelentenek.

Első közelítésben csak az ügyvitel legáltalánosabb egyedtípusait (adatosztályait) kell figyelembe venni, úgymint:

1. ügyfelek,
2. eszközök,
3. kötelezettségek,
4. termelési/szolgáltatási folyamatok és
5. dolgozók.

Ezután a legáltalánosabb egyedtípusok kevésbé általános adatosztályokra bomlanak, mint például az ügyfelek vevőkre és szállítókra, az eszközök nyersanyagra, félkész termékekre, késztermékekre, gépekre, berendezésekre és szolgáltatásokra, a kötelezettségek tervekre, törvényes kötelezettségekre és szerződésekre, stb.

Az adatosztályok létrehozása után hozzájuk kell rendelni az őket valamilyen alapvető módon befolyásoló tevékenységeket, az azokat meghatározó események segítségével. Ezt az

összekapcsolást vagy hozzárendelést az egyed-esemény mátrix segítségével ábrázoljuk. (matrica odnosa poslovni procesi/klase podataka). Az események vagy eljárások általában a mátrix soraiban helyezkednek el a termelés/szolgáltatás életciklusában való megjelenési sorrendjük alapján. Az oszlopokban az adatosztályok jelennek meg a következő elhelyezési algoritmus szerint: a mátrix első oszlopába az az adatosztály kerül, amelyet a mátrix első sorában elhelyezkedő tevékenység (esemény) hoz létre, a második oszlopba az az adatosztály, amelyet a második sorban megjelenő tevékenység hoz létre, stb. A mátrix sorainak és oszlopainak kiosztása természetesen fordított is lehet. Az ügyviteli erőforrások szempontjából a tevékenységek csoportosíthatóak, az így kapott tevékenység-csoportokat alrendszereknek tekintjük, amelyekhez tehát a megfelelő adatosztályok kapcsolódnak. Az egyed-esemény kapcsolati mátrix elemei éppen az előbb említett tevékenység-csoportok szerint csoportosíthatók. A csoportosítás logikája a következő: a csoport (blok, a grafikus ábrázoláson téglalap) bal felső sarkát az ügyviteli erőforrások szempontjából csoportosított valamelyik tevékenység-csoport első tevékenysége, és az általa létrehozott adatosztály képezi, a megfigyelt tevékenység-csoport utolsó tevékenysége, és az általa létrehozott adatosztály pedig az információs rendszer tevékenység-csoport által meghatározott alrendszerének a jobb alsó elemét. Az egyed-esemény mátrix elemeinek ilyen módon kialakított csoportjai általában a mátrix átlója mentén helyezkednek el.

A BSP módszer (technika, módszertan) helye és szerepe az információs rendszerek fejlesztésének életciklusában az életciklus fázisai (lépései) alapján becsülhető fel:

1. az információs rendszer fejlesztésének megtervezése,
2. rendszerelemzés és -tervezés,
3. az információs rendszer megvalósítása,
4. az információs rendszer működtetése és karbantartása és
5. az információs rendszer minősítése és ellenőrzése.

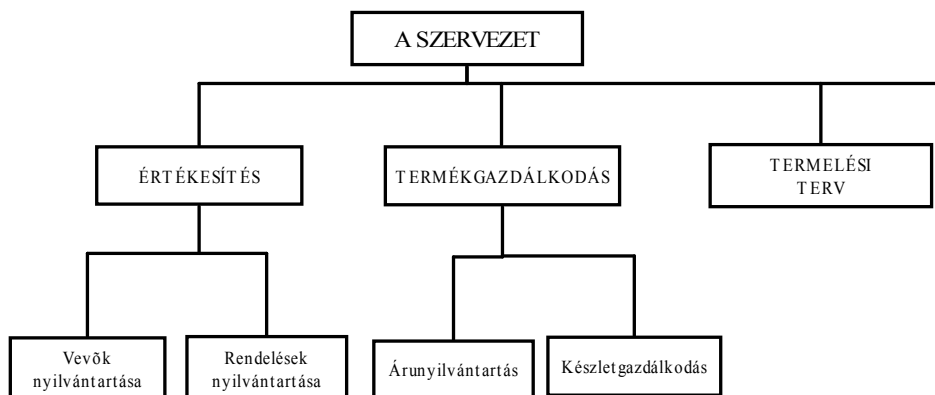
Sok informatikai gyakorlatban jártas szakember véli úgy, hogy a BSP technika a project management témakörébe tartozik, és nem az információs rendszerek fejlesztésébe. Éppen ezért léteznek olyan rendszerfejlesztési életciklusok, amelyekben a fent megjelölt első lépés (az információs rendszer fejlesztésének megtervezése) – vele együtt a BSP is – nem szerepel.

Az előző bekezdésben szereplő megfontolások és gyakorlati tények ellenére is hangsúlyozni kell a BSP technika „Az információs rendszer architektúrájának kialakítása” nevű lépésének gyakorlati hasznosságát a feldolgozások makrotervezésében. Az egyed-esemény mátrix létrehozása itt az információs rendszer alrendszereinek kialakításában segít, amelyet az irodalomból [11.] kölcsönzött példával a 8.2. táblázat szemlélteti.

8.2. táblázat

Egyedtípusok (Adatosztályok) Funkciók (Csoportosított tevékenységek)	V e v ő	S z e r z ő d é s	R e n d é s	Á r j e g l y é z é k	Á r u	K é s z l e t
Vevőnyilvántartás	C	C	-	-	-	-
Rendelésnyilvántartás	C	U	C	U	U	U
Árunyilvántartás	-	-	U	C	C	U
Készletgazdálkodás	-	-	-	U	C	C
Termelési terv	-	-	U	U	-	-
Durvaprogram(termelésütemezés)	-	-	U	U	-	-

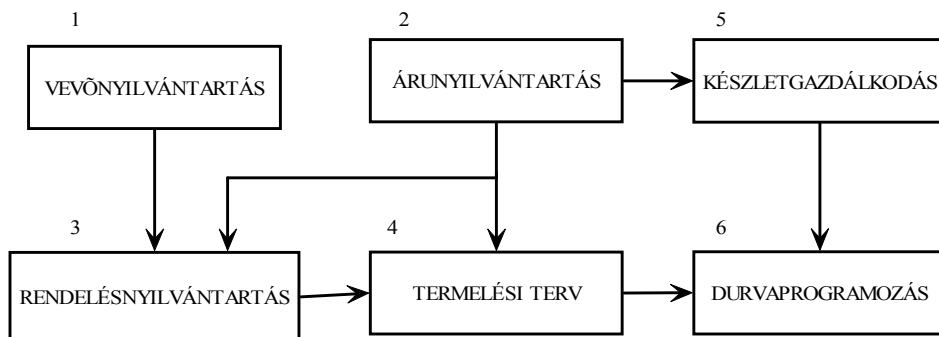
A mátrix elemei a következő három értéket vehetik fel: C – létrehozza (create), U – használja (use) és – nem használja. A mátrix megmutatja, hogy mely funkciókhoz mely adatosztályok kapcsolódnak. Ezek alapján a feldolgozás statikus (funkcionális) felépítése (architektúrája) is meghatározható (8.9. ábra⁷).



8.9. ábra.

⁷ Forrás [11], 244. oldal.

A feldolgozás statikus képe nem ad eligazítást a rendszer működésével, a feldolgozások időbeli sorrendjével kapcsolatosan, márpedig az információs rendszert működtetni kell, hiszen azért készült. A feldolgozások logikai (időbeli) sorrendjét a 8.10. ábrán feltüntetett háló ábrázolja⁸.



8.10. ábra.

8.5.4. Mikrotervezés

A makrotervezés az információs rendszer alrendszerének (fő rendszerrészeinek, 8.2. tábla) meghatározására, a feldolgozás-hierarchia (8.9. ábra) kialakítására és a feldolgozás-háló (8.10. ábra) megszerkesztésére szolgál. Mivel ott csak a fő funkciók (rendszerrészek) valamint azok feldolgozásbeli kapcsolata kerül „terítékre”, a makrotervezés szintjét gyakran nevezik a feldolgozások konceptuális szintjének is, mivel arról szól, hogy mit milyen egymásutániságban kell elvégezni.

Az információs rendszer fő rendszerrészei funkciócsoportokat takarnak. A rendszerrészekben (alrendszerben) belül a funkciók meghatározása, végrehajtásuk logikai (időben) sorrendjének meghatározása (a funkciócsoport működésének a megszervezése) tartozik a feldolgozástervezés logikai szintjéhez (itt ugyanis a funkciócsoportok működésének mikéntjéről van szó). Ezt a kérdéskört elemzik a középső absztrakciós szintre kidolgozott technikák a

bemenet – feldolgozás1 – adatbázis – feldolgozás2 – kimenet

lánc taglalásával. Az adatfolyamatábra (AFD- adatfolyam diagram) tartalmazza a fő tevékenységeket (funkciókat) is, és a fő adathalmazokat (az egyedtípusokat) is, ugyanakkor azonban ábrázolja a vezérlési és adatkezelési műveleteket is.

⁸ Forrás: [11], 245. oldal.

Az adatfeldolgozásban háromfajta műveletet alkalmazunk: adatkezelési, új adat előállítására szolgáló és vezérlési műveleteket. A feldolgozástervezés gyakorlatában a legtöbb figyelmet az adatkezelési műveletek kapnak. Erre utal az adatkezelési gráfok és az adatáramlási diagramok (AFD – adatfolyam diagram) széleskörű alkalmazása a feldolgozástervezés középső absztrakciós szintjén. Az teljesség és egyértelműség érdekében célszerű az adatfeldolgozás elemzését és tervezését mikroszinten is elvégezni. A mikroszintű elemzés és tervezés fontos eszközévé válhat egy régi (régén elfelejtett) technika amit csak döntési táblázat néven emlegettünk hajdanán, és a vezérlési műveletek bekerekítését, ellenőrzését tehetné teljesebbé. Az új adat előállítására szolgáló műveletek a feldolgozás vezérléséhez hasonlóan mostohagyereknek számítanak, pedig az adatelőállítási műveletek elemzésének és tervezésének is van egy hasznos technikája (igaz régi módszerről van szó), amelyet (sajnos) ma nem süvegelnek meg a modern tervezők/elemzők, ez pedig az előzmény-háló (precedencia-háló).

8.5.4.1. Döntési táblázatok

Az adatfeldolgozás egyes funkcióinak megvalósításában előfordulhatnak olyan bonyolult helyzetek vagy feltételcsokrok, amelyek olyan sok elemet ölelnek fel, melyeknek folyamatos (egyidőbeni) fejbeni nyomonkövetése már meghaladja az emberi képességeket. Másrészt a feltételrendszerek (különösen ha túl bonyolultak) programban való rögzítése nem ildomos a feltételek változékonysága miatt. A legjobb megoldásnak az ilyen helyzetekben a döntési táblázatok alkalmazása látszik. A döntési táblázatok az időben gyorsan változó döntési helyzetek leírására különösen alkalmasak, mert áttekinthetőek, egyszerűen módosíthatók, programtervezésre, de program-elemzésre is jók, sőt, még a döntéselőkészítési folyamatban is jó szolgálatot tehetnek (a számítógépes alkalmazástól teljesen függetlenül). Döntési táblázat többféle módon is készülhet (különbféle elemeket tartalmazhat). A 8.3. tábla olyan ötvöztött döntési táblázatot ábrázol, amelyik a [22]-es és [13]-as irodalmi forrásból táplálkozik.

Maga a döntési táblázat a 8.3. táblán kihangsúlyozott (vastagított) kerettel rendelkezik, körülötte csak az eligazítást segítő magyarázatok találhatók. A döntési táblázat két fő részből áll: a felső döntési részből és az alsó akció vagy tevékenységi részből. A fő részek további két részre bomlanak.

A döntési főrész feltételekre és értékmátrixra tagolódik. A feltételekben konkrét értékek is szerepelhetnek, az értékmátrix pedig a feltételek igaz és hazug voltának összes kombinációját tartalmazza, vagyis a feltételek beteljesülését (megvalósulását) minősíti. Az értékmátrix

elemei (e_{ij}) általában csak I (igaz) esetleg Y (yes), H (hamis) vagy N (no) és – (ami a közömbösséget jelöli, vagyis azt, hogy a feltétel beteljesülésének eredménye-minősége nem befolyásolja az eredményt) értéket vehetnek fel. A döntési táblázat harmadik negyede, vagyis a tevékenységi vagy akció főrészt tevékenységek negyede, ahogy a neve is utal rá, az elvégzendő vagy elvégezhető tevékenységeket tartalmazza. Az akció főrészt második része (a negyedik negyed) a funkciómátrix, amelyet néha kiválasztott funkcióknak is neveznek. A funkciómátrix elemei (a_{ij}) csak X jelet vagy – jelet tartalmazhatnak. A funkciómátrix elemének X értéke azt jelenti, hogy a hozzá tartozó sorban feltüntetett tevékenységet el kell végezni, a – érték pedig azt, hogy a hozzá tartozó sorban elhelyezkedő tevékenységet nem kell végrehajtani.

8.3. táblázat.

		Feltételek	Értékmátrix
IF	D	F_1	$e_{11} e_{12} \dots e_{1k}$
	Ö	F_2	$e_{21} e_{22} \dots e_{2k}$
	N	F_3	$e_{31} e_{32} \dots e_{3k}$
	T	.	.
	É	.	.
	S	F_n	$e_{n1} e_{n2} \dots e_{nk}$
THEN	A	T_1	$a_{11} a_{12} \dots a_{1k}$
	K	T_2	$a_{21} a_{22} \dots a_{2k}$
	C	T_3	$a_{31} a_{32} \dots a_{3k}$
	I	.	.
	Ó	.	.
	K	T_m	$a_{m1} a_{m2} \dots a_{mk}$
		Tevékenységek	Funkciómátrix

A döntési táblázat bemutatására olyan iskolapéldát választunk, amelyet minden hallható ismer (vagy legalábbis ismernie kellene). A feladat elég egyszerű ahhoz, hogy belőle a döntési táblázat létrehozására összpontosíthasson a hallgató, és ne magára a feladatra. A döntési táblázatot az általános másodfokú egyenlet megoldására alkalmazzuk. Tételezzük fel, hogy a másodfokú egyenlet alakja: $ax^2+bx+c=0$. Ennek az egyenletnek a gyökeit kiszámíthatjuk a 8.4. táblán elhelyezkedő döntési táblázat alapján.

8.4. tábla

$b^2 - 4ac > 0$	Y	N	N
$b^2 - 4ac = 0$	N	Y	N
$b^2 - 4ac < 0$	N	N	Y
$rex = -\frac{b}{2a} \quad imx = \frac{\sqrt{ b^2 - 4ac }}{2a}$	X	X	X
$imx1 = 0 \quad imx2 = 0$	X	X	-
$rex1 = rex \quad rex2 = rex$	X	X	X
$rex1 = rex1 + imx \quad rex2 = rex2 - imx$	X	-	-
$imx1 = imx \quad imx2 = -imx$	-	-	X

8.5.4.2. Előzmény-háló

A feldolgozástervezés folyamán a vezérlési utasítások fontos szerepet töltenek be a funkciók helyes működésének kialakításában. A vezérlési utasítások megfogalmazásánál ügyelni kell arra, hogy egy lehetőség, egyetlen ág se maradjon ki a bonyolult feltételek (döntési helyzetek) esetében sem: ebben segít a döntési táblázat.

Az információs rendszer kimeneteinek, vagy új adatok előállításának tervezésekor szükségeszerű rögzíteni a kimenetek illetve új adatok előállításának módját a rendelkezésre álló adatok alapján. Az új adatok vagy kimenetek előállításának elemzése és tervezése könnyebben elvégezhető egy teljesen formális eljárás vagy módszer segítségével, amelyet előzmény-hálónak (precedencia-hálónak) esetleg előzmény-gráfnak (precedencia-gráfnak) hívnak.

A modern információs rendszerek esetében az egyik legkellemetlenebb állapot áll be akkor, ha az előzmény-háló megszakad vagy módosul. Manapság ugyanis nem az a régen jól bevált „közvetlen” feldolgozás történik, amelyben a bemenetektől az egymenetes feldolgozás végén, mintegy „egyszuszra” kimenetek keletkeznek (generálódnak), hanem az adatokat adatbázisban tároljuk (törzsadatok), esetleges későbbi feldolgozásuk céljából. A

bemenet – feldolgozás – kimenet

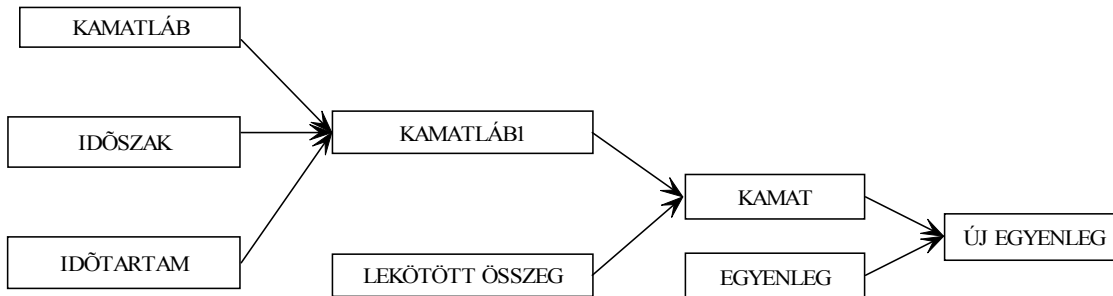
stílusú feldolgozás ma már túlhaladottnak számít (nem is igen alkalmazzák már sehol), helyette a következő módon végezzük a feldolgozást:

bemenet – feldolgozás1 – adatbázis

és

adatbázis – feldolgozás2 – kimenet.

A feldolgozás funkciói és műveletei így teljesen szétszóródnak az információs rendszeren belül, ami az előzmény-háló fontosságára mutat rá. Az előzmény-háló nem más, mint a konkrét funkció foglami modellje vagy belső struktúrája a végrehajtandó funkcióhoz szükséges adatok szempontjából. A 8.11. ábrán látható a lekötött betétek kamatszámítását szemléltető precedencia-háló egy irodalomból vett példa⁹ szelídített változata.



8.11. ábra.

8.5.5. Feldolgozás-konzisztencia

A feldolgozástervezés a legtöbb esetben csupán a középső absztrakciós szint adatfolyamdiagramjaival történik. Az adatáramlási diagram az adatokat és tevékenységeket tükrözi a végrehajtásuk sorrendjének megfelelően. A kontextus diagram, amely a nulladik szintű AFD-nek felel meg, az egész információs rendszert egyetlen tevékenységként (folyamatként) ábrázolja. Az első szintű diagram a fő rendszerrészeket (alrendszereket vagy funkciókat), a második szintű diagramok már az egyed típusok tulajdonságtípusait érintő funkciók tevékenységeit taglalják.

A fő funkciócsoportok (rendszerrészek, alrendszerek) meghatározása a dokumentumok áramlása alapján pontatlan, szubjektív is lehet. Ennek kiküszöbölésére, vagyis a rendszerrészek objektív meghatározására alkalmazzák a makrotervezés módszereit. A makrotervezés leggyakrabban használatos módszere az IBM által kifejlesztett BSP (Business System Planning) metodológia „Az információs rendszer architektúrájának kialakítása“ néven ismert módszere. Ezzel biztosított a makrotervezés és a középső absztrakciós szintű feldolgozástervezés közötti összhang.

⁹ Forrás: [22], 421. oldal.

Az AFD alsóbb szintjeinek esetenként a feldolgozások bonyolult döntési helyzeteit is ábrázolniuk kell a funkciók szerves részeként. Ilyen esetekben tesznek jó szolgálatot a döntési táblázatok, amelyek segítik az összetett feltételekben való eligazodást, a feldolgozás alapos és pontos elemzését és tervezését. Az AFD-k az információs rendszer kimeneteit is tartalmazzák, a kimenetek előállítási technológiáját azonban nem. A felhasználó igényeit lefedő jelentések adatait azonban a rendelkezésre álló adatokból elő kell tudni állítani: ebben segítenek az előzmény-hálók. Tartalmazzák az új adatokhoz illetve kimenetekhez szükséges adatokat, és az előállítás ütemezését (dinamikáját), ezzel biztosítva a konzisztenciát a feldolgozások középső absztrakciós szintje és a mikrotervezés között.

Az adatfolyam diagram ugyan híven tükrözi a bemeneteket, az adattárolókat és a kimeneteket, de semmit sem mond a kimenetek létrehozhatóságáról. Az AFD-ken nincs adat arról, hogy a feltüntetett kimenetek előállíthatók-e a rendelkezésre álló (bemeneti) adatokból, sem arról, hogy az összes rendelkezésre álló (és feltüntetett bemeneti) adatra szükség van-e a jelentések adatainak előállítására. Az AFD arról sem szól, hogy a bemeneteket hogyan használják a feldolgozások (és melyek azok az adatok, amelyekre egyetlen feldolgozás sem hivatkozik, ami természetes is, hiszen ezeket a leírásokat az előzmény-háló tartalmazza, amiből viszont hiányzik a feldolgozásban szereplő adatok eredete.

A kimenetek előállításának szükséges és elégséges feltételét (a kimenetek létrehozásához szükséges és elégséges adatokat) a HIPO (Hierarchy plus Input Process Output) diagramok tartalmazzák. Az IBM által megalkotott HIPO módszer a bonyolult szoftvertermékek dokumentálásának segédleteként látta meg a napvilágot. A módszer top-down szemléletet támogat, és a fő rendszerrészekről (funkciócsoportok, alrendszerek), a funkciókon keresztül a tevékenységekig, sőt, műveletekig taglalja a feldolgozásvetületet a bemenetektől egészen a kimenetekig. A részletezettségi szint szempontjából a HIPO módszer három különböző komponensből áll [13]:

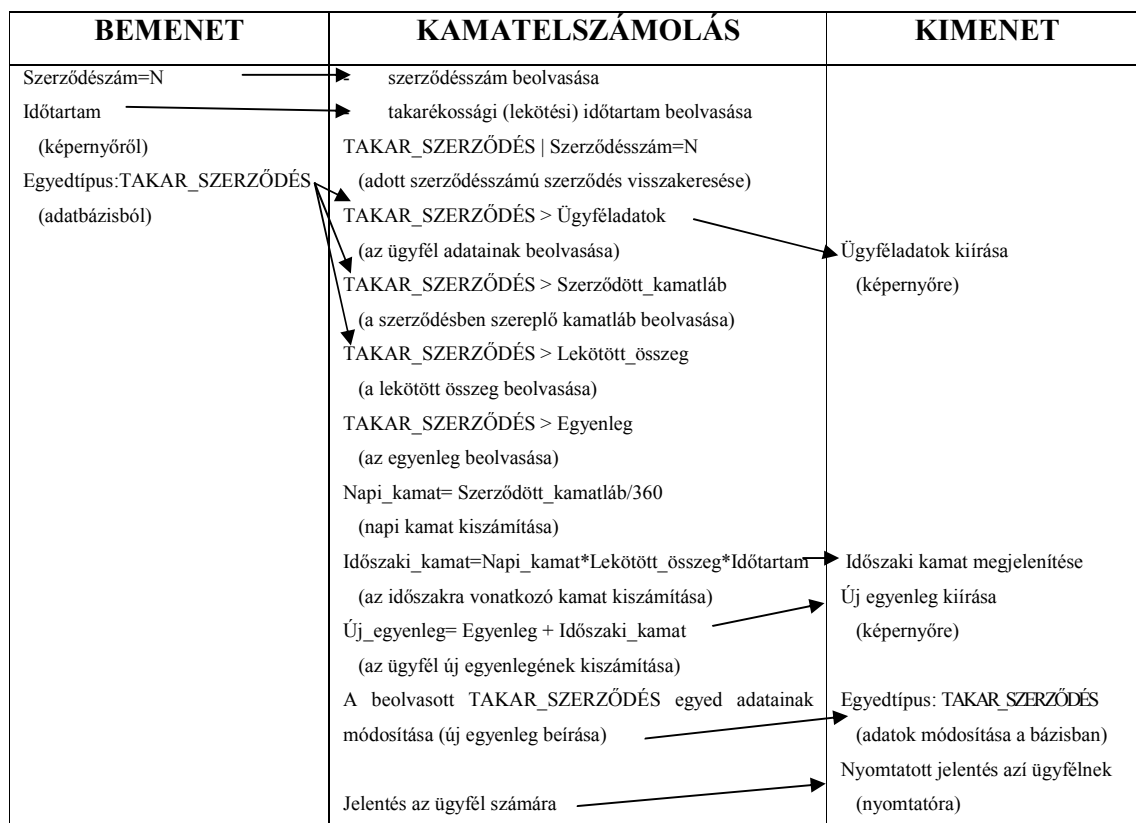
1. Vizuális „tartalomjegyzék”,
2. Áttekintő diagramok és
3. Részletes diagramok.

A HIPO módszer vizuális „tartalomjegyzéke” az információs rendszer első szintű (durva) bontását adja. Mivel a fő rendszerrészek kialakítására már van választott módszerünk (a BSP metodológia „Az információs rendszer architektúrájának kialakítása” elnevezésű módszere), a HIPO módszer első komponensével nem foglalkozunk tovább. Ez a vizuális „tartalomjegyzék” eredményként a BSP említett módszerével egybevágó (vagy ahhoz nagyon hasonló, de

mindenképp annak megfelelő) eredményt ad. A BSP alkalmazása a 8.2. táblán, illetve a 8.9. ábrán látható.

Az áttekintő diagramok szintje a feldolgozástervezés középső absztrakciós szintjének felelnek meg (adatáramlási diagramok-AFD, adatkezelési gráfok szintje), és csakúgy mint az AFD-k, a funkciók megvalósításának logikáját és ütemezését nem tartalmazzák (a kimenetek előállításának szükséges és elégséges feltételéről sem szólnak).

A részletes HIPO diagramok a tevékenységek műveleteit is ábrázolják, vagyis a feldolgozásvetület legfinomabb részleteit is megtaláljuk rajtuk. Míg az áttekintő diagramok bemenetei és kimenetei rendszerint dokumentumok vagy egyedtípusok, addig a részletes diagramok a legtöbb esetben az egyedtípusok tulajdonságtípusait kezelik bemenetként és kimenetként egyaránt. A részletes HIPO diagram bemutatása a 8.12. ábrán látható a kamatelszámolási probléma kapcsán.



8.12. ábra

9. AZ INFORMÁCIÓS RENDSZER KÖRNYEZETVETÜLETE

Az információs rendszer környezetvetülete a felhasználókat, az erőforrásokat és a szabványokat és a (szabványosított) eljárásokat foglalja magába. Az erőforrások a valós világ tükrözésére szolgáló információs rendszer előállítására (megvalósítására) és működtetésére szolgálnak, ezek pedig az idő, a pénz és a technikai eszközök (a hardver és a szoftver). A technikai eszközökről általánosságban már esett szó az ötödik fejezetben, a konkrét megoldási lehetőségekről pedig nincs sok értelme eszmefuttatást folytatni, hiszen a kombinációs lehetőségek száma a hardver és szoftver terén még akkor is nagyon nagy, ha a két terület gyors fejlődését teljesen figyelmen kívül hagyjuk. Az idő és a pénz általánosan elfogadott vélemény szerint a project management vonzáskörébe tartoznak, így ebből a jegyzetből ők is kimaradnak.

A környezetvetületnek így csak két tényezője maradt az ismerkedésre, ezek a felhasználók és a szabványok és eljárások (az utóbbi kettőt rendre együtt emlegetik és egy tényezőnek tekintik. Róluk szól ez a néhány oldalas kilencedik fejezet.

9.1. A FELHASZNÁLÓI KÖRNYEZET

A környezetvetület kizárólag valós tényezőket ölel fel, azt az objektíven létező közeget, amelyben az információs rendszer fejlesztése, megvalósítása és működtetése zajlik. Ez a megállapítás érvényes a felhasználókra is, és a felhasználói környezetre is. A környezetvetületet nem kell kitalálni, kialakítani, létrehozni, mert az már létezik.

A felhasználói környezetbe a végfelhasználók (end user), az alkalmazási felhasználók (application user), a fejlesztési csoport tagjai, a vezető, és bizonyos értelemben a szervezeti egységek tartoznak.

9.1.1. Felhasználói pszichológia

Az információs rendszer a végfelhasználóknak készül. Ebben csak annyi a probléma, hogy a végfelhasználók sokszor maguk sem tudják pontosan megmondani, hogy mit is szeretnének, mit várnak el az információs rendszertől. Másrészt a végfelhasználók inhomogén csoportot alkotnak, hiszen különböző szociális, politikai és kulturális rétegekhez tartoznak. Az információs

igények begyűjtésénél erre a jellegre (a végfelhasználók csoportjának inhomogén jellegére) ügyelni kell. A fejlesztő csoport kapcsolatot kell, hogy teremtsen az inhomogén végfelhasználói csoport minden rétegével (természetesen nem minden nap, és nem az összes végfelhasználóval egyenként). A végfelhasználóknak készülő jelentések a megcélzott csoport tagjainak (tehát a végfelhasználóknak) is érthetőeknek kell lenniük, nemcsak az elemzőknek és a tervezőknek.

Az alkalmazási felhasználók testreszabott, konkrét feladatokat (funkciókat) végeznek az információs rendszeren belül. Számukat tekintve kisebb csoportot képviselnek mint a végfelhasználók, és homogénebb közösséget képeznek az előzőknél. Szakmai szempontból „közös nevezőn” vannak, közös zsargonból használják a fogalmaikat, és a fogalmakat takaró kifejezéseket. A fejlesztés számukra akkor hasznos, ha a szakterületüknek megfelelő és szokványos, a szakterületet a szemontjukból jól tükröző menürendszer és beviteli képernyős űrlaprendszer biztosít. A különböző alkalmazási felhasználók számára testreszabott használati utasításokat kell biztosítani, amelyekben mindenki a saját feladatainak számítógépes (információs rendszer által biztosított) elvégzésére kap eligazítást. Jobb helyeken a használati utasítások mellett az alkalmazási felhasználók számára kiképzést (a használatot oktató tanfolyamot) is biztosítanak.

A fejlesztési csoport tagjai heterogénebb (kevésbé homogén) csoportot képeznek az alkalmazási felhasználók csoportjától. Ők is csak hús-vér emberek, akik közül egyesek feltételek a tudásukra, mások annyira el vannak telve a tudásukkal, hogy a földre sem lépnének, ha nem volna muszáj, ismét mások arrogánsak, vagy aszociálisak az együttműködés szempontjából. A jó rendszer kifejlesztésének feltétele az, hogy a fejlesztő csoport tagjai (e főleg az elemzők és a tervezők) jó „színészek” legyenek: el tudják képzelni, bele tudják magukat élni a többi felhasználói csoport tagjainak szerepébe (meg hát a sajátjukba is). A jó végtermék (információs rendszer) érdekében a megrendelőknek és a felhasználói csoportoknak közösen kell fáradozniuk, és mindenkinek „szívügyként” kell kezelnie az „ügyet”. A fejlesztési tevékenységek összehangolása a fogyasztói csoportok között a különböző bemutató-, munka- és egyeztetési értekezleteken történik.

A vezető szerepét az információs rendszer fejlesztésében nem szabad lebecsülni. Ő az, aki példát mutat a fejlesztési csoporttal való együttműködés terén és ösztönzi a munkatársait az eredményes közös munkára (ha a vezetőt hidegen hagyja az információs rendszer fejlesztésének ténye, vagy sorsa, akkor legtöbbször az egész projekt – a fejlesztés – sikertelenségre van ítélve). A vezető ellenőrizheti és számonkérheti a hatáidők betartását úgy a fejlesztői csoporttól, mint a saját beosztottjaitól. A fejlesztői csoportnak tehát érdeke a vezetővel való jó együttműködés kialakítása. Az előzőekből jól kiérezhető, hogy a fejlesztői csoport vezetőjének, de a többi

fejlesztőnek is, amellet, hogy jó színész, és jó verbális vonással rendelkező, könnyen kapcsolatot teremtő személy, még pszichológiai ismeretekkel is rendelkeznie kellene.

9.1.2. Munka-pszichológia

A munka-pszichológia ugyan, az időhöz és pénzhez hasonlóan, a project management hatáskörébe tartozik, mégis fontosnak tartjuk, ha csak vázlatosan is, a fejlesztés leggyakoribb problémáinak felvetését. Az információs rendszer fejlesztési csoportját többféle szakember alkotja. A fejlesztési csoport tagjainak részvétele a fejlesztési folyamatban nem egyenletes: pl. az elemzők és a tervezők a fejlesztés elején és a rendszertesztelési fázisban aktív(bb)ak, a programozók pedig a programtervezési, programkódolási, programtesztelési és rendszertesztelési fázisokban. Az információs rendszer fejlesztési folyamata (az információs rendszer életciklusa három időszakra – folyamatra – bontható: fejlesztési időszak, működtetési időszak – gyakorlati alkalmazás – és változáskezelés – change management) után a gyakorlati használat időszaka következik, amely egybeesik a változáskezelés időszakával. Az időbeni egybeesést azonban nem kellene összetéveszteni és összekeverni, hiszen a működtetés és a változáskezelés két különböző lényegre támaszkodik. A működtetést végző szakemberek gyakran „belenyúlnak” a programokba, hogy azokat a saját „szájízükhöz” igazítsák, pedig ez nem az ő hatáskörükbe tartozik. És ez ugyan miért baj, merülhet fel a kérdés. Azért, mert minden változtatásnak (néha csak nehezen belátható) következményei lehetnek. Ezért a változtatásokat azoknak kell elemezni és megtervezni, akik teljes átlátással rendelkeznek a rendszerről. Az információs rendszer egyik kitörölhetetlen jellege a változás, ami nem divat, nem kitalált aktivitás, hanem objektív szükséglet. Ez a tény, illetve jelenség ugyancsak az információs rendszer munka-pszichológiájába tartozik.

A fejlesztési csoport (team) vezetője inhomogén csoportot igazgat, hiszen a tagok nagyon eltérő képzettségi és érdeklődési körökből valók. Az elemzők és a tervezők inkább a felhasználók irányában nyitottak, és a rendszernek, mint egésznek a működésében érdekeltek. Munkájuk gyümölcsét csak hosszabb idő elmúltával látják meg, és tudják értékelni. A programozók viszont azonnal programokat szeretnének írni (ők a türelmetlenebbek), hogy mielőbb láthassák és megmutathassák munkájuk eredményét. Ezt sem szabad figyelmen kívül hagyni, vagy kihagyni az információs rendszerek munkapszichológiájából. A rendszertesztelés folyamán a hibák megjelenésekor az elemzők/tervezők és a programtervezők/kódolók egymásra mutogatnak. A tervezők azt állítják, hogy a programozók nem tudnak programokat írni, amire a programozók

azt válaszolják, hogy a rendszertervezők nem tudják megfogalmazni a rendszertervben, hogy mit is akarnak (az állítás és válasz sorrendje persze fordított is lehet). Ha a fejlesztést egy a fejlesztésre szakosított cég végzi, a nehézségek a fejlesztési csoport tagjainak egyenletlen terheléséből fakadhatnak, amit részben a mátrix-alapú munkaszervezéssel (lásd bővebben a 2.2 pontban) igyekeznek enyhíteni.

A következő munkaszervezési probléma, ami szintén a munka-pszichológia területére kívánczik, az egyes feladat- vagy szerepkörök keverése, különösképpen az informatikai vezető, az adatadminisztrátor (adatgazda), az adatbázis-adminisztrátor és a rendszergazda vonatkozásában.

Az informatikai vezető (CIO – Chief Information Manager) elsősorban szervező, menedzser, aki az informatika általános, vállalati szintű problémáival foglalkozik: az informatikai stratégia kialakítása és megvalósítása, az információs rendszer működéséhez szükséges erőforrások biztosítása, az informatikai munkálatok durvaprogramozása (ütemezése), stb. Az adatgazda vagy adatadminisztrátor a cég főinformatikusa, aki az ismeretek megismerését, elrendezését és kezelését vállalati szinten irányítja. Személyes felelősséggel tartozik az egységes adat- és feldolgozásmóddel kialakításáért, a modellekben szereplő rész(let)ek összehangolásáért, a szabványok betartásáért (minőségellenőrzés) és a változáskezelésért (change management). A két szerepkör szorosan kapcsolódik egymáshoz, ezért gyakran egy személyt bíznak meg mindkettő betöltésére. Az adatbázis-adminisztrátor (database administrator) az információs rendszer működtetési időszakában az adatok előkészítésének, átalakításának (adatkonverzió), ellenőrzésének, rögzítésének, hozzáférési jogosultságainak és hibajavítások területein tevékenykedik (ezekért felelős), a fejlesztés folyamán pedig részt vesz az egységes adatmodell fejlesztésében, gondozásában és megőrzésében, valamint az adatbáziskezelő rendszer kiválasztásában, annak programozási lehetőségeinek feltárásában, valamint a szabványosított eljárások kidolgozásában. A rendszergazda vagy rendszer-adminisztrátor (System Administrator) általában az információs rendszer fő rendszerrészeinek a felelőse, de kisebb cégek esetén az egész információs rendszer működését felügyeli, ide értve a számítógép-hálózatok működését és a felhasználók és azok jogosultságainak kezelését is. Az utóbbi két szerepkör ugyancsak szoros kapcsolatban áll egymással, ezért gyakran ezeknek az elvégzését is egy személyre ruházzák rá.

Két olyan fogalom is szerepel az előző bekezdésben, amelyet nem, vagy csak ritkán veszünk komolyan, ezek pedig a minőségellenőrzés (Quality Control) és a változáskezelés (Change Management). Mindkettőért a cég főinformatikusa a felelős. A minőségellenőrzés alatt az előírt vagy elfogadott szabványok betartásának számonkérését (minden törvény – szabvány annyit ér, amennyire betartják azt) értjük. A változások elkerülhetetlenek az információs

rendszerek életében, hiszen ők a változó valós világot tükrözik vissza a lehető legjobb közelítéssel. A változáskezelés a fejlesztés része, ezért nem a működtetőknek kell gondoskodiuk róla, hanem csak jelezni a változás szükségességét. A fejlesztők feladata, hogy eldöntsék vajon szükség van-e a rendszer vagy annak egy részének áttervezésére. A változások dokumentációba való átvezetéséről nem kell döntenie a fejlesztőknek: az kötelező.

9.2.A SZABVÁNYRENDSZER

Az információs rendszer fejlesztésére és működtetésére nincs nemzetközileg elfogadott szabványrendszer. A vállalatoknak így önerőből kell kialakítaniuk a saját, belső szabványukat. A cégeknél a belső használatra kidolgozott és elfogadott szabványrendszer a közös (legáltalánosabb értelmezésben értendő) nyelv használatát kell, hogy előírja az alapkoncepció, a struktúra, a terminológia, a megfogalmazási mód, a kritériumrendszer, a tervezési algoritmus, a dokumentációkezelés, a technikai támogatás és a metaszabványok területén.

A belső (információs rendszerre vonatkozó) szabvány nem lehet tetszőleges. Az információs rendszer ugyanis nem üvegbúra alatt hermetikusan őrzött rendszer, hanem számtalan szállal kötődik a külvilág, a környezet (törvény által előírt, kötelező, vagy nem kötelező de hasznos – pl. ISO ajánlások) szabványokkal. A technikai eszközök használata a fejlesztésben és működtetésben eleve kikényszerít bizonyos viselkedési formát, illetve tevékenységet. A különböző beállítottságú felhasználók más-más módon gondolkodnak és szemlélik a világot: sajátos, egymástól eltérő terminológiát, különböző megfogalmazási módokat, stb. használnak. A külvilág eltérő szabványait, előírásait, szokásait, megegyezéseit a belső szabványoknak kell kibékíteniük (közös nevezőre hozniuk, szinkronizálniuk), ami azt jelenti, hogy a belső szabványok megfogalmazása fáradságos és hálátlan feladat, mert mindig akadnak elégedetlenek a szabvány alkalmazásának úgyszólván minden területén.

A szabvány maga is rendszer, mert egymással összefüggő tényezőkből épül fel, de ezt a tényt és megállapítást már a 4.2.5. pontban tisztáztuk. A szabvány absztrakció is egyben, mert az előírásokat ki kell gondolni, az ott használatos ismereteknek, fogalmaknak nevet kell adni, a szabvány felépítését (struktúráját) ki kell alakítani. Kétfajta szabványt különböztetünk meg (használnunk is illene mindkettőt): a felhasználói és fejlesztési (meta) szabványt.

A felhasználói szabványok az információs rendszer tényezőire vonatkoznak, úgy mint az adatokra, tevékenységekre, eseményekre, technikai eszközökre, stb. A metaszabványok (róluk a 6.3.8. pontban esett már szó) a project management munkáját segítik, mert a rendszerfejlesztés folyamatát és az információs rendszer tervdokumentációjának kidolgozását (szerkezet, forma és

tartalom szempontjából) szabályozzák. A metaszabványok a rendszerfejlesztés keretén belül külön kell(ene), hogy foglalkozzanak az elemzési eljárásokkal, a különböző tervezési módszerekkel, a programozási technikákkal, a teszteléssel és az átadási előírásokkal. Nagyon tömören ez úgy fogalmazható meg, hogy a metaszabványok az információs rendszer fejlesztésével kapcsolatos összes adatra, eseményre, tevékenységre, erőforrásra és felhasználóra vonatkozó előírások.

Az előzőekből az is kiderül, hogy a metaszabványok keretén belül dokumentációs szabvány is helyet kell, hogy kapjon. Az információs rendszer terve dokumentáció formájában ölt testet, és az ide vonatkozó előírások elsősorban a dokumentáció felépítésére, tartalmára és ábrázolási módjára vonatkoznak. Mivel a dokumentáció a felhasználó adataival kapcsolatos adatokat tartalmazza gyakran nevezik metaadatbázisnak is. A dokumentációkészítés eszköze másodlagos szempont: készíthető akár „gyalogosan”, kézzel is, de annak nem sok értelme van, hiszen több szoftvertermék is minőségesen el tudja végezni helyettünk a „kézimunkát”. A dokumentáció minőségét a megalkotóinak a képességei és az elfogadott szabványok adják meg nem pedig a létrehozására kiválasztott eszköz (szoftver) minősége.

9.3.AZ INFORMÁCIÓS RENDSZER DOKUMENTÁCIÓJA

Amennyiben a szabványok az információs rendszer fejlesztésére vonatkoznak, úgy azokat metaszabványoknak nevezzük. Vannak tehát metaszabványok és szabványok, és ezek a fejlesztésben végighúzódnak két vezérfonalat dokumentálják: a menedzselését és a megalkotását. Így kétfajta dokumentációt kell vezetni a fejlesztés folyamán¹:

1. projektdokumentáció és
2. az információs rendszer dokumentációja (rendszerdokumentáció).

A projektdokumentáció a rendszerfejlesztés folyamatát örökíti meg az összes, a fejlesztés folyamatával kapcsolatos szervezési és adminisztratív tartalmak leírásával, gazdagítva ezzel a tapasztalatokat a projektmenedzsment témaköréből.

A rendszerdokumentáció az információs rendszer tervét szabványosított formában tartalmazza (ha a fejlesztők a fejlesztés és a dokumentációkészítés folyamán tartották magukat valamilyen elfogadott szabványokhoz). A rendszerdokumentáció három dokumentumból, három részből áll:

1. tervdokumentáció,

2. felhasználói dokumentáció és
3. üzemeltetési dokumentáció.

A tervdokumentáció az információs rendszer leírása, ábrázolása diagramok, táblázatok és tárgyilagos, lakonikus mondatok segítségével. A tervdokumentáció készül el elsőként, mert a másik kettő erre épül.

A tervdokumentáció **első fejezete** a BEVEZETŐ vagy ÖSSZEFOGLALÓ. Benne a célt vagy célhierarchiát és az összefoglalót, vagy átfogó képet kell leírni. Ha célhierarchiáról van szó (a legtöbb esetben ez várható), akkor a részcélok prioritására (fontossági sorrendjére) is utalni kell. A pontosan és jól meghatározott cél precízen utal a tartalomra. Itt található tehát a rendszer tartalmának (tényezőinek) leírása, a rendszer határainak megjelölése. Az információs rendszer tartalma a valós rendszerben alkalmazott fogalmak mögött (háttérben) rejtőzködik. A fogalmak a leendő egyed típusok forrásai, amelyek aztán, jóval később, a feldolgozásokat is behatárolják (ezekből válogathat majd a felhasználó). A rendszerhatárok meghúzása után következhet a tagolás, amit legcélszerűbb az ismereti elv (adatelv) szerint elvégezni. Ezzel a statikus struktúrához jutunk, de itt, a bevezetőben kívánatos még a rendszer viselkedéséről is adni egy rövid ismertetést, valamint a továbbfejlesztés irányelveiről.

A tervdokumentáció **következő három fejezete** az információs rendszer három vetületének leírására fenntartott hely, amit szokványos a vetületek nevei alapján elnevezni:

1. adatvetület(i fejezet),
2. feldolgozásvetület(i fejezet) és
3. környezetvetület(i fejezet).

Mivel az első két vetület az elemzők és tervezők intellektuális erőfeszítéseinek a terméke ezeket három absztrakciós szinten kell megfogalmazni: fogalmi (konceptuális), logikai és fizikai szinten. Idő és fáradság takarítható meg, hála a relációs adatbáziskezelő-rendszereknek, mert alkalmazásuk esetén átvállalják a logikai adatmodell fizikai adatmodellé való átalakítását, valamint a fizikai szint megvalósítását (mellékesen szólva módosítási lehetőséget sem biztosítanak a tervezők és programozók számára). A két absztrakt vetület dokumentálásánál figyelembe kell venni az ANSI/SPARC ajánlásokat (8.1. ábra) és három fajta sémát kell készíteni (dokumentálni). Sohasem szabad megfeledkezni arról, hogy az események (amelyeket adatokkal írunk le) és a tevékenységek a valós rendszerben bonyolódnak, és a számítógépre való leképezésüket az az információs rendszer végzi, amelyet éppen dokumentálunk. A számítógépen belül elhelyezkedő adatok szervezését a belső séma ábrázolja, a gépen kívüli, valós világ adatainak összefüggéseit pedig a külső séma ábrázolja. A külső és belső séma közötti kapcsolatot

¹ Forrás: [11.], 270. oldal.

a fogalmi séma biztosítja. A három absztrakciós szint leírása mellett a dokumentációban helyet kell adni az egymás mellett elhelyezkedő szintek közötti leképzések leírására is, valamint a leképzések miértjére (okaira). Ezzel együtt lesz csak teljes a kép a valós világ leképzésének menetéről, és a valós világ képének információs rendszer által történt elferdítéséről.

A harmadik vetületnek (környezeti vetület) csak egy, fizikai szintje van, mert az információs rendszernek a valós rendszerben is létező tényezőit foglalja magába. A környezeti vetületet leíró fejezet a felhasználókról, a szükséges erőforrásokról és a szabványokról szól. A hardver- és/vagy szoftverkönyezet változása esetén csak a környezeti vetület megváltozott erőforrására vonatkozó részeket kell átírni.

A felhasználói dokumentáció az alkalmazási és végfelhasználók eligazodását segíti feladataik megoldásában, illetve a jelentések megértésében. A bevezető itt is olyan általános (technikai) tájékoztatót tartalmaz, amelyet mindenkinek tudnia kell, a továbbiakban pedig az alkalmazási felhasználók általános eligazítását tartalmazó rész következik, majd annyi külön egység (fejezet) ahányféle alkalmazási felhasználója lesz a rendszernek. Ezek a fejezetek az össz lehetőséget, tevékenységet, eseményt, adatot, képernyőleírást, jelentést, stb. tartalmazzák, amelyek a rendszer használata közben adódnak, történhetnek, megjelenhetnek, stb. de csak a fejezetben megjelölt felhasználó-félre vonatkozóan.

Az üzemeltetési dokumentáció a működtető személyzetet segíti. A felhasználói dokumentációtól bonyolultabb, mert az üzemeltetésnél a rendszerfunkciókon kívül egyéb feladatok is elvégzésre várnak. A rendszerfunkciók leírása mellett tehát helyet kell adni külön fejezetekben a sajátos működtetési funkciókat végző személyek (adatbázis-adminisztrátor, operátor rendszergazda, stb.) eligazításának is.

Elfogadott szabványok és jó dokumentáció hiányában az információs rendszert nem lehet teljes mértékben (teljes értékűen) használni. Működtetni, karbantartani és változtatni pedig egyenesen lehetetlen. A rendszerdokumentáció (a tervdokumentáció, felhasználói dokumentáció és üzemeltetési dokumentáció hármását értjük alatta) az információs rendszerrel együtt kerül átadásra (kötelezően), és a megrendelő tulajdonát képezi.

10.FELHASZNÁLT IRODALOM

1. Baga E.: Delphi másképp..., GDF, Budapest, 1999.
2. Bana I.: Az SSADM rendszerszervezési módszertan, LSI, Budapest, 1994.
3. Bennatan E.M.: SoftverProject Management: A Practitioner's Approach, Second Edition, McGRAW-HILL Book Company Europe, 1995.
4. Čarapić M., Pendić Z., Furht B., Veselinović D.: Osnovi projektovanja informacionih sistema zasnovanih na primeni računara, Tehnička knjiga, Beograd, 1974.
5. Date C.J.: "An Introduction to Database Systems", Vol. 1, Third Edition, Addison-Wesley Publ. Co., Reading, MA, 1981.
6. Đorđević B., Vodoprivredni sistemi
7. Đurković J., Tumbas P.: Analiza i projektovanje informacionih sistema – CASE STUDY, EF Subotica – ALEF Novi Sad, 1997.
8. Görög M.: Általános projektmenedzsment, Aula Kiadó, Budapest, 1996.
9. Halassy B.: Az adatbázis-tervezés alapjai és titkai, IDG, Budapest, 1994.
10. Halassy B.: Az információs rendszerek alapfogalmai, SZÁMALK, Budapest, 1982.
11. Halassy B.: Ember – Információ – Rendszer, IDG, Budapest, 1996.
12. Kuzmanov S., Mogin P., Petković I., Popović M.: "Automatizacija projektovanja šeme baze podataka", JAHORINA '88.
13. Lazarević B., Jovanović V., Vučković M.: "Projektovanje informacionih sistema" I deo, Naučna knjiga, Beograd, 1986.
14. Lazarević B.: "Prošireni model objekti-veze" interni materijal, Laboratorija za informacione sisteme FON-a, Beograd, april 1990.
15. Lazarević B.: Baze podataka, Radni materijal za interne kurseve, Beograd, 1989.
16. Lerner A.J.: Principi kibernetike, Tehnička knjiga, Beograd, 1970.
17. Marjanović Z.: ORACLE – relacioni sistem za upravljanje bazom podataka, Breza, Beograd, 1990.
18. Mogin P., Luković I., Govedarica M.: Principi projektovanja baza podataka, FTN-STYLOS, Novi Sad, 2000.
19. Mogin P., Luković I.: Principi baza podataka, FTN-STYLOS, Novi Sad, 1996.
20. Noszkay E.: Informatikai és rendszerszervezési alapismeretek, GDF, Budapest, 1994.
21. Purba S., Sawh D., Shah B.: How to Manage a Succesfull Software Project, John Wiley & Sons, 1995.
22. Raffai M.: Információrendszer-fejlesztés, Novodat Kiadó, Győr, 1999.
23. Strukturirana systemska analiza, IskraDelta, Ljubljana, 1984.
24. Šuc L.: Projektovanje baza podataka, Izobrazevalni center DELTA, Ljubljana, 1985.
25. Szelezsán J.: Adatbázisok, LSI, Budapest, 1996.
26. Ullman J.D.: "Principles of Database Systems", Computer Science Press, 1980.
27. Ullman J.D., Widom J.: A First Course in Database Systems, Prentice Hall Inc., 1997. (Adatbázisrendszerek – Alapvetés, PANEM, Budapest, 1998.).
28. Veljović A.: Modeliranje podataka – Erwin (materijal za kurs), CIT, Beograd, 1998.
29. Veljović A.: Modeliranje procesa – Bpwin (materijal za kurs), CIT, Beograd, 1998.
30. Yourdon E., Constantine L.L.: "Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design", Prentice Hall, Inc. Engl. Cliffs, N.J., 1979.
31. Kovács Attila, Sediviné Balassa Ildikó: Információs rendszerek tervezése, szervezése, GDF, Budapest, 2000.
32. Garcia-Molina H., Ullman J.D., Widom J.: Database System Implementation, Prentice Hall Inc., 2000. (Adatbázisrendszerek megvalósítása, PANEM, Budapest, 2001.).