

1 Bevezetés az információrendszerek alapfogalmaiba

Az informatikában nagyon sok fogalmat használnak nem elég pontosan meghatározott tartalommal, vagy csak helyileg egységesült tartalommal, amelyet mások nem fogadnak el. Ezért ebben a jegyzetben illetve a hozzá csatlakozókban használt legfontosabb fogalmakra adunk meghatározásokat.

Ezzel az információrendszerek és fejlesztésük legfontosabb fogalmait alapozzuk meg, ami lehetővé teszi, hogy megtárgyaljuk a módszerek bizonyos technikáit.

1.1 Módszer vagy módszertan

Módszer

Speciális formájú eljárás vagy eljárások halmaza, amely bármely mentális erőfeszítést, gondolkodást igénylő tevékenységi körben előfordulhat, különös tekintettel nagy bonyolultságú feladatok megoldásánál.

Szisztematikus (rendszeres) eljárás, technika vagy információ gyűjtési eljárás, amelyet valamely tudományág alkalmaz vagy megfelel egy tudományágnak;

Készségeknek, tapasztalatoknak és technikáknak szervezett halmaza.

Ebben az értelemben egy technológiai előírást, amely például informatikai rendszerek elkészítésére vonatkozik, módszernek nevezhetünk. Azonban az ilyen módszerekre a 'módszertan' elnevezés terjedt el, nemcsak magyarul.

Módszertan (Információrendszereknél)

Az ismereteknek, tudásnak olyan szervezett halmaza, amelyet a módszerekről gyűjtöttek össze. (A módszerek tanulmányozásának tudománya: egy értekezés vagy tanulmány a módszerről.)

Egy módszer elveinek a gyűjteménye, amely minden konkrét esetben egy az adott helyzethez illesztett módszer formájában testesül meg, amely alkalmas az adott feladat megoldására. ([Checkland81])

Egy módszertan eljárások, technikák, eszközök és dokumentációs segédeszközök gyűjteménye, amely fázisokból, szakaszokból áll össze. Azonban egy módszertan több mint ezek egyszerű összege. Általában valamilyen filozófiai álláspontra helyezkedik, amely meghatározza az alapvonalait, ellenkező esetben nem lenne több mint egy recept vagy egy szakácskönyv.

Tehát módszertan alatt egy módszer valamilyen meta-leírását fogjuk érteni, amely konkretizálható minden egyedi esetben és tartalmaz útmutatásokat a technikai megoldásokra.

Technika

Technikai (műszaki) eljárás vagy tevékenységek szervezett halmaza.

Az információrendszerek stratégia és informatikai tervezése során a terv leírásának módja nagyon fontos. Ezt a módot jelölés technikának, diagram technikának nevezzük.

Jelölés technika (Diagram technika)

Jelek, szimbólumok, karakterek (betűjelek), rövidítések rendszere, amelyek technikai (műszaki) tények, mennyiségek rögzítésére szolgál.

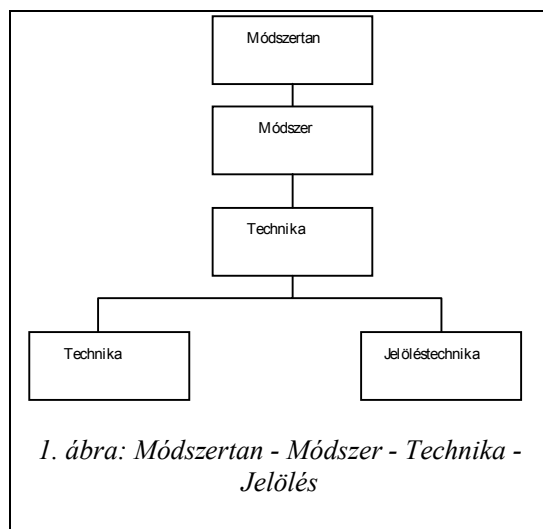
1.2 Mit nevezünk rendszernek

Definíció 1-1 Rendszer

Egymással kapcsolatban álló részekből áll, amelyek egy közös cél érdekében működnek együtt.

A rendszerelemzők általában egy szervezeten belül dolgoznak. Itt a 'rendszer' azt jelenti, hogy szervezett módon dolgoznak valami megvalósításáért. Kívülről nézve, a szervezeteknek van határa, amelyen túl létezik az a környezet, amelyben a rendszer (szervezet) működik. Egy szervezetet nyílt rendszernek kell tekintenünk mivel más rendszerekkel áll kapcsolatban, nevezetesen azzal a környezettel, amely a határain kívül helyezkedik el. Belülről nézve, a rendszer további alrendszereket tartalmaz, amelyek az együttműködésen keresztül szolgálják a közös célt. Ezt az együttműködést szabályok és a külvilágból érkező impulzusok, (stimuláló) események irányítják.

Definíció 1-2 Szinergia (Synergy)



A részrendszerek hatásainak, kibocsátásainak az összegződése: ha közösen egy egységnek tekintjük őket, az egész nagyobb mintha csak egyedi különálló elemekként fogjuk fel őket.

Erre vonatkozik a következő idézet: "Az egész mindig nagyobb mint a részeinek összege", Arisztotelész.

A rendszerek osztályozása:

- Természetes - Ember által alkotott
- Absztrakt (pl. nyelvek, számrendszerek)
- Procedurális (pl. jogrendszer, szervezeti struktúrák)
- Fogalmi (pl. relativitás elmélet)
- Konkrét

- fizikai (pl. számítógép rendszer)
- társadalmi (pl. szervezetek)
- Zárt rendszerek
- Nyílt rendszerek (pl. információrendszer, egy szervezet, cég, stb.)
- Determinisztikus (pl. számítógép program)
- Nem-determinisztikus (pl. egy szervezetre a külvilágban bekövetkező események ill. ezek hatása)
- Valószínűségi (pl. minőség ellenőrző rendszer)
- Véletlenszerű (pl. tőzsde)
- Emberi rendszerek (pl. a szervezet stratégiai tervezői, fejlesztő csoport)
- Gépi rendszerek (pl. számítógép)

- Ember/gép rendszerek (pl. tranzakciós rendszer, vegyi üzem)
- Tervszerű rendszerek (pl. számítógépes információrendszer)
- Adaptív rendszerek (pl. szervezetek)
- Nem adaptív rendszerek
- Egyszerű rendszerek
- Bonyolult rendszerek

Definíció 1-3 Részrendszer

Részrendszer alatt önmagában megálló rendszer értünk, egyedül a kívülálló megfigyelő által elfogadott nézőpont különbözik, pl. egy szervezet információrendszere a szervezeten belül, egyszerre részrendszer és önmagában megálló rendszer.

Informatikában, emberek, gépek, módszerek halmaza, amelyek bizonyos tevékenységek végrehajtására vannak megszervezve.

A rendszerszemléletű megközelítés

Érdekes erről az általános megközelítésről szót ejteni. Több módszertanról állítják azt készítőik, hogy a rendszerszemléletű megközelítést követik. Az általános rendszer elméletről több alapvető munka is megjelent pl. ([Bertalanffy68],[Churchman68]). Ezek alapján érdemes egy kicsit az idetartozó fogalmakat absztraktabb módon megtárgyalni, mielőtt azokat esetleg az információrendszerek közegében használnánk és túlnyúlnak a rendszerek alap definícióin.

A rendszerszemléletű megközelítés egy egységes (holisztikus) valaminek (entitásnak, objektumnak, stb.) tekinti a rendszert nem meg feledkezve az alkotó részekről. Ez a szemlélet észleli az alkotórészek aktivitását, tevékenységét, de ugyanakkor figyelmet fordít a rendszer egésze által mutatott aktivitásra.

Néhányan ezt a megközelítést 'fekete doboz' megközelítési módnak tekintik, a rendszer bemeneteit és kimeneteit meghatározva írják le a rendszert. Természetesen, a rendszer határainak a meghatározásához, azaz a bemenetek és kimenetek pontos megadásához, meg kell mondani azokat a (rendszer)állapotokat is, amelyekhez hozzátartoznak. Ezen a módon nemcsak egy szervezetet tekinthetünk rendszernek, hanem azt az információt is, amely a szervezeten belül található, azaz az adatokat és a hozzájuk kapcsolódó folyamatokat.

Miért érdemes a rendszerszemléletű megközelítést alkalmazni? Egyrészt a modern szervezetekben egyre gyorsuló ütemben nézünk szembe a bonyolultság és a változatosság növekedésével. A bonyolultság növekedését a következőknek tulajdoníthatjuk:

- a technológiai fejlődés;
- az értékesítési piacok kiterjedése, növekedése;
- a kutatás és fejlesztés eredményei és ennek hatása;
- az életszínvonal emelkedése;
- a termékek szakadatlan változása;
- a nemzetközi politikai és gazdasági hatásokkal az összefonódás, és a kölcsönös függőség;
- a gazdaság magán és nem-magán szférájának növekvő kölcsönös függősége.

Másrészt, új elméletek és technikák állnak rendelkezésre, amelyekkel a rendszerszemléletű megközelítést sikerrel lehet alkalmazni a vezetésben, irányításban és a szervezet információrendszereire.

Elméleti megközelítések:

- matematikai rendszerelmélet,
- kibernetikai rendszerelmélet, információ-visszacsatolás elmélete
- irányítás/vezérlés elmélet,
- információ elmélet,
- játék- és döntésemélet,
- informatika és számítógéptudomány,
- szimuláció, stb.

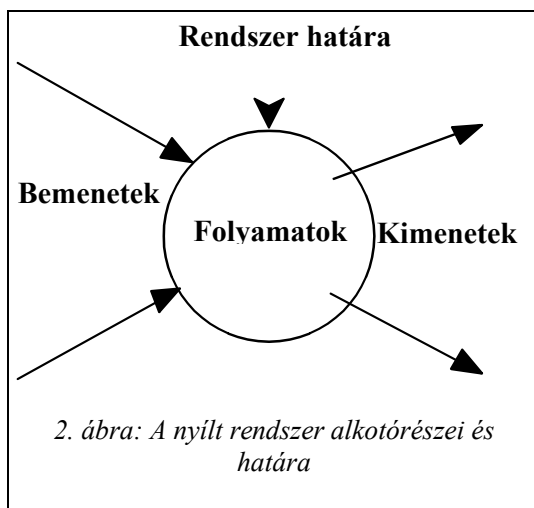
1.2.1.1 A rendszermodellek mint absztrakt rendszerek

Absztrakt rendszerek csak fogalmakból állnak - szemben a *fizikai* rendszerekkel, amelyek valamit csinálnak, valahogy viselkednek -, amelyek a fizikai valóságban nem léteznek, csak a fogalmak. ideák világában. Az absztrakt rendszerek nem csinálnak semmit - ez a legfőbb megkülönböztető jel a fizikai rendszerekhez képest - azonban van *céljuk*, valamilyen fizikai rendszer ábrázolása, leírása.

Egy információrendszer készítésénél általában az első lépés egy absztrakt rendszer létrehozása, amely egy szervezet bizonyos részét modellezi. A következő lépésekben ezt az induló modellt bővítik, és ily módon modellek sorozatát hozzák létre, amelyek egyre pontosabban írják le a kívánt fizikai rendszert. (Más mérnöki tudományokban is hasonló elvek szerint járnak el, pl. hídépítésnél, vagy épületek tervezésénél; az absztrakt rendszer tartalma ábrázolja a kívánt fizikai rendszert.)

1.2.1.2 A nyílt rendszerek

Nyílt rendszer a legáltalánosabb típusa a rendszereknek, ezt mutatjuk be az ábrán (Ild. 2. ábra). Vagyis egy nyílt rendszer folyamatokból áll, amelyek a környezetükből valamilyen *bemeneteket* (**input**) kapnak, és *kimeneteket* (**output**) állítanak elő. Természetesen közbenső eredmények is képződhetnek a folyamatok egyes szakaszaiban. A *rendszer határa* választja el a rendszert magát a környezetétől, és tulajdonképpen a folyamatok, a bemenetek és a kimenetek határozzák meg. A bemeneteket és kimeneteket a nyílt rendszer statikus míg a folyamatokat a rendszer dinamikus elemének tekintjük. Pl. egy cég iktatási és postázási rendszere nyílt rendszer, egy információrendszer is az.



1.2.1.3 A zárt rendszerek

Elméletileg, egy zárt rendszer olyan rendszer, amelyik nem lép kapcsolatba a környezetével, erre példa egy olyan kémiai 'reaktor', egy lezárt edény, amelyben valamilyen vegyi reakció folyik. Azonban, tökéletesen zárt rendszer legfeljebb laboratóriumi körülmények között létezhet csak, ha egyáltalán annak tekinthető ebben az esetben is. Sokkal hasznosabb ha zárt rendszernek azt tekintjük, amelyik a minimalizálja a környezetével zajló cserének a bizonytalansági fokát, azaz csak az előre pontosan meghatározott bemeneteket (jól-definiált inputok) fogadja el és csak előre pontosan meghatározott kimeneteket bocsát ki a feldolgozás után. (Akinék ismerősnek tűnik ez a meghatározás

az nem véletlen, ez a *strukturáltság* alapelve.) Ezzel szemben, a nyílt rendszereknek meg kell birkóznuk a bizonytalan bemenetekkel, ezért nagyon alkalmazkodó képeseknek (*adaptivitás*) kell lenniük, pl. ilyenek az emberek, és a társadalomban létező szervezetek.

A zárt és nyílt rendszereket egy skála két szélső pontjának tekinthetjük, annak megfelelően, hogy a bemeneteiknél milyen fokú bizonytalansággal kell számolnunk. A tipikus rendszerek, amikkel a gyakorlatban találkozunk általában egyik szélsőséghez sem tartoznak, hanem köztük helyezkednek el valahol.

1.2.1.4 A részrendszerek: rendszerek különböző szinteken

Egy rendszert leírhatunk néhány mondattal, egy diagrammal vagy akár egy több ezer oldalas dokumentummal. Egy olyan leírást, amely rengeteg részletet tartalmaz, *alacsony absztrakciós szintűnek* tekintünk, míg egy *magas absztrakciós szintű* leírás kevés részletet tartalmaz. Egy rendszert általában szisztematikus módon, részekre bontva, dekomponálva vagy finomítva (ekvivalens kifejezések) írunk le a folyamatainak magas absztrakciós szintjéből indulva és lejutva egy alacsonyabb absztrakciós szintre, amikor is a rendszer, úgy mond, *részrendszerekből* áll.

Ezt a lépés sorozatot rendszerint nagy és bonyolult rendszereknél hajtjuk végre, ezek a rendszerek olyannyira komplexek, hogy nem lehet leírni sem felfogni vagy megérteni őket egyetlen egy rendszerként. Ezért bontjuk fel olyan méretű részrendszerekre, amelyeket képesek vagyunk kezelni és megérteni. Elképzelhető, hogy még tovább bontjuk ezeket a részrendszereket is, hogy további részleteket tudjunk leírni. Gyakran valamilyen hierarchikus diagram segítségével ábrázoljuk a köztük fennálló összefüggéseket, egyes részrendszerek több másik (rész)rendszerhez tartozhatnak így egy háló formájában és nem faszterkezetben tudjuk leírni a kapcsolódásukat egymáshoz.

A már idézett definíciónk szerint (Isd. Definíció 1-3.) minden *részrendszert* önmagában rendszernek tekinthetünk és ezért további részrendszerekre bonthatjuk.

A kommunikáló részrendszerek

Egy nagy rendszer leírását egyszerűsíthetjük azzal, hogy részrendszerekre bontjuk, de ezért komoly árat kell fizetnünk, nevezetesen meg kell határoznunk a köztük fennálló kapcsolatokat, *felületeket (interface)*, amelyeken keresztül a *kommunikáció* lezajlik. Egy *felület* alatt azokat a bemeneteket és kimeneteket értjük, amelyeket egynél több rendszer közösen használ. Nyilvánvalóan, a bemenetek / kimenetek leírásának mindkét vagy több rendszerben létezniük kell és azonosaknak is kell lenniük. Ez különösen fontos számítógéprendszerek, - folyamatok és - programok esetén.

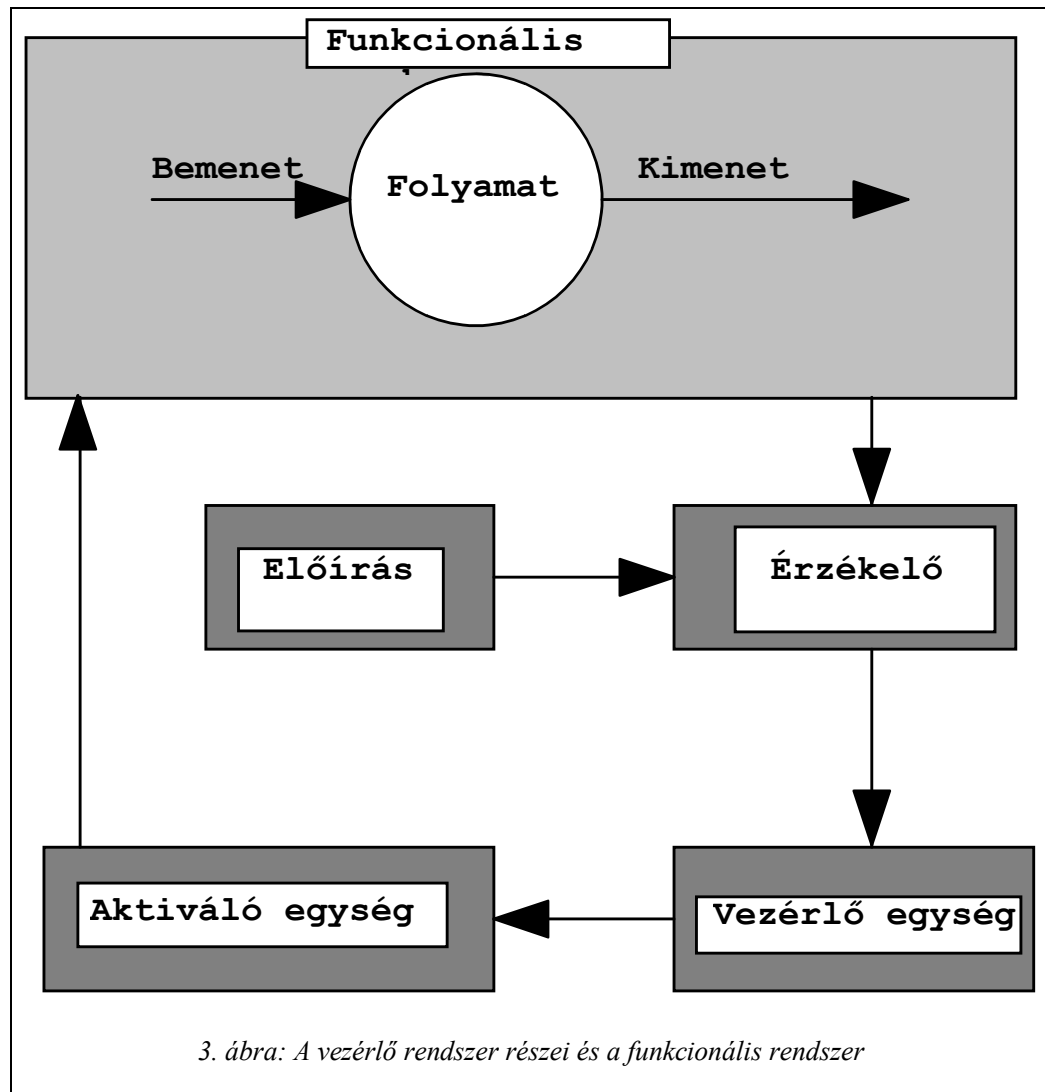
A részrendszerek csatolása

Amikor a rendszerek kölcsönhatását részletesebben vizsgáljuk, akkor időnként azt találhatjuk, hogy az egyik rendszer kimenetét egy másik rendszer azonnal felhasználja. Az ilyen eseteket nevezzük *szoros csatolásnak*. Vannak esetek, amikor ez egyáltalán nem kívánatos, mert például a fogadó rendszer csak bizonyos diszkrét állapotokban, vagy bizonyos időpontokban tudja a saját bemenetét (a másik kimenetét) feldolgozni. Egy másik ok lehet a két feldolgozási folyamat sebességének különbsége lehet.

Ekkor egy lehetséges megoldás az, ha a két rendszer szoros kapcsolatát megszüntetjük, a két rendszert *szétkapcsoljuk*, azaz a két rendszer egymástól függetlenül működik, legalábbis egy ideig. Ezt úgy valósíthatjuk meg, hogy *buffert* használunk, ahol az első rendszer kimenete várakozik addig, amíg fel nem használják.

Determinisztikus rendszerek

Bizonyos rendszerek viselkedését meg tudjuk jósolni, azaz egy adott bemenetre tudjuk, hogy milyen kimenetet fog a rendszer előállítani, vagyis létezik egy szabály, amely a bemenetet a kimenethez rendeli. Az ilyen rendszerek általában egyszerű folyamatokból állnak, és ahhoz, hogy a rendszer célja megvalósuljon elegendő a megfelelő bemenetről gondoskodni.



3. ábra: A vezérlő rendszer részei és a funkcionális rendszer

Nem-determinisztikus rendszerek

Nem-determinisztikus vagy *sztochasztikus* rendszerek viselkedése előre nem jósolható meg, azaz nem tudjuk azt, hogy hogyan idézhetnénk elő a rendszer céljának megvalósulását, sőt még azt sem tudjuk, hogy egy adott bemenetre milyen kimenetet fog előállítani. Ez a helyzet vagy nagyon bonyolult folyamatoknak a következménye, vagy nagyon kevésbé megértetteknek, vagy ismeretlen esetleg meghatározhatatlan bemeneteknek, vagy ilyen bemenetek kombinációjának. Például ilyen rendszer az időjárás, Magyarország gazdasága, stb.

Rendszerek viselkedésének kézbe tartása

Ha azt akarjuk garantálni, hogy a rendszerek céljai megvalósuljanak, folyamatosan ellenőriznünk kell a működésüket. A bemenetek lehet, hogy pontatlanok, vagy a kevésbé megértett folyamatok között összeütközések merülnek fel, ezért a rendszerhez hozzá kell kapcsolnunk egy *vezérlő rendszert*, amelyik ellenőrzi az eredeti rendszert (ezt most funkcionális rendszernek hívhatjuk), vajon helyesen viselkedik, működik-e.

Kimenet ellenőrzés

Valamilyen eszközzel észlelik a rendszer kimenetét és összevetik valamilyen előre rögzített előírással. Bármilyen eltérés egy helyreigazítási tevékenységet, korrekciót indít el, amelyet a rendszer bemenetként kap meg, ezzel működésbe hozva a funkcionális rendszert és az előírásokhoz közelebb álló kimenetet generál. A vezérlő rendszer azonban megváltoztathat folyamatokat, önszervező (adaptív) rendszereknél megváltoztathatja a rendszer célját.

Negatív visszacsatolás azt jelenti, hogyha a rendszer kimenete eltér az előírttól (valamilyen mintavételezés során), akkor a vezérlő rendszer megpróbálja az eltérést csökkenteni, a kimenetet az előíráshoz közelíteni. A pozitív visszacsatolás ennek az ellenkezőjét jelenti - ha a kimenet, annak legfontosabb jellemzői eltérnek az előírttól, akkor a rendszer megismétli az eljárást ezzel tovább növelve az előírttól az eltérést.

Bemenet ellenőrzés

A vezérlő rendszer gyakran ellenőrzi a rendszer bemenetét ugyanúgy, ahogy a kimenetét. Ezt szokták szűrésnek hívni.

A vezérlés nehézségei

A vezérlő rendszert úgy tekinthetjük mint olyant, amely csökkenti a rendszer bizonytalansági fokát. Azonban egy bonyolult rendszer vezérlése gondot okozhat, ugyanis leegyszerűsítve, minden lehetséges rendszerállapothoz tartoznia kell egy vezérlési állapotnak. Ráadásul, a rendszer elemektől vezérlési-információkat kell kapni és továbbítani nekik, ami lényegesen megnöveli a járulékos információ-feldolgozási feladatokat. Egy viszonylag nyitott vagy nem megjósolható bemenetekkel dolgozó rendszer állapotainak meghatározása gyakorlatilag lehetetlen. Ilyen esetekben a gyakori megoldás egy ember-számítógép hibrid rendszer létrehozása, amelyben a számítógéprendszer reagál az előre megállapítható esetekre, míg az ember a nem várt esetekben hoz döntéseket.

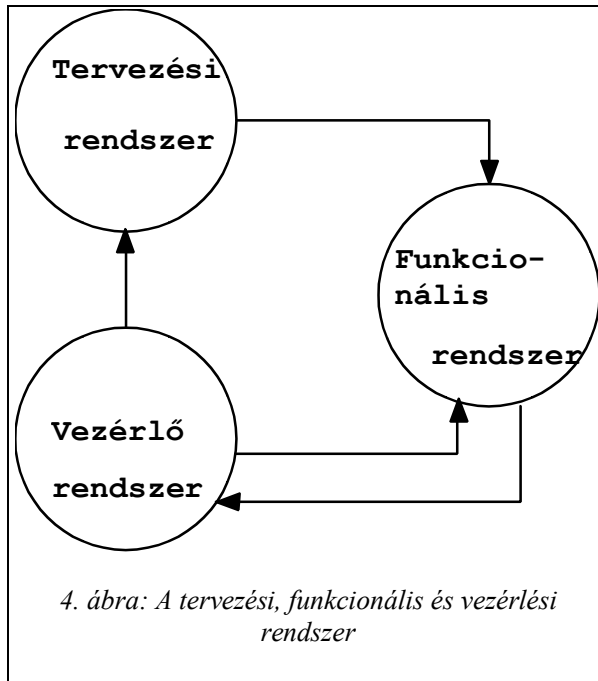
Az állapottér

Az állapottér fogalma egy a rendszerek viselkedésének leírásához kapcsolódó kiegészítő fogalom, amely a folyamatokat és egyéb elemeket (objektumok, entitások, stb.) is figyelembe veszi. Ekkor a rendszert úgy fogjuk fel, hogy bármilyen időpillanatban egy adott állapotban van, ezt az állapotot azok az elemek határozzák meg vagy jellemzik, amelyek ebben az adott időpontban a rendszeren belül léteznek. Ezek az elemek nem csak a fizikai valóságban léteznek lehetnek, hanem például egy adott személy kora is lehet egy lényeges tényező / elem / objektum az adott rendszerben.

Ezt a megközelítést csak az ún. *diszkrét* rendszereknél lehet alkalmazni, azoknál, amelyeknek megkülönböztethetően azonosítható állapotaik vannak. Az információrendszerek diszkrét rendszerek.

Az aktualizálási folyamatok a folyamatok fontos típusát alkotják, ebben a megközelítésben egy olyan transzformációt jelentenek, amelyek a rendszert az egyik állapotból a másikba viszik.

A tervezési rendszer



4. ábra: A tervezési, funkcionális és vezérlési rendszer

Egy szervezeten belül rendszeresen felülvizsgálják és újraterveznek rendszereket, ezért érdemes egy tervezési alrendszert is beilleszteni a funkcionális és vezérlési alrendszer mellé azért, hogy egy átfogóbb keretbe helyezzük őket. A vezérlő rendszer észleli a funkcionális rendszer kimenetének eltérését a célkitűzéstől, ezután egy a tervezési rendszer számára szóló bemenetet állít elő azért, hogy megváltoztassa a funkcionális rendszer folyamatait, bemeneteit, kimeneteit vagy célkitűzéseit. A tervezési rendszer olyan kimeneteket állít elő, amelyek az új folyamatok vagy vezérlési módok tervét tartalmazzák.

A rendszerszemléletű megközelítés hátrányai

A valós világ leírása különböző nem-egyértelmű leírása lehet az eredménye ennek

a megközelítésnek; a módszer eltérő alkalmazása különböző eredményekre vezethet.

A létrejött modell nem teljes, vagy nem pontos, minthogy nincs pontos előírás arra, hogy milyen részletességűnek kell a leírásnak lennie.

Nincs kifejezetten javasolt módszer, vagy általános egyetértés abban, hogy az esetleg létező módszerek közül melyik a legmegfelelőbb.

" Az oroszán és a ló rendszerszemléletű megközelítésben ugyanaz, mégis egy kicsit másképpen kell velük bánni".

A rendszerszemléletű megközelítés előnyei

A rendszer fogalmának fentebbi meghatározásainak megfelelően közelítjük meg a valós világot vagy annak egy részét.

Így ez a megközelítés egy informális eszközt nyújt a helyzet megértéséhez és leírásához, az ösztönösen ismert fogalmak segítségével, mint például a folyamat, bemenet, kimenet.

A folyamatok *hierarchikus lebontása (dekompozíciója)* a bonyolult, összetett folyamatok elemzésének hasznos eszköze.

A vezérlő rendszer fogalma illetve ennek fontossága rámutathat arra, hogy egy ilyen hiányzik, nem hatékony, esetleg eredménytelen.

A rendszer ilyen jellegű leírása általában elég egyszerű ahhoz, hogy kommunikációs eszközként használjuk a személyek közötti párbeszéd lebonyolításához, természetesen megfelelő magyarázat kíséretében.

Az adatközpontú megközelítés

Ebben a megközelítésben elsősorban és hangsúlyosan azokkal az adatokkal foglalkoznak, melyek a rendszeren belül és a rendszerrel kapcsolatban felbukkannak. Tehát meghatározzák

az adattípusokat és attribútumokat majd egy adatmodellt állítanak elő, amely leírja a az adattípusok közötti kapcsolatokat. Ha helyesen és a leendő felhasználók aktív részt vételével hajtják végre, akkor azokat a dolgokat pontosan sikerült meghatározni, amelyekről a szervezet nyilvántartást akar vezetni. A modellt általában normalizálják a relációs elmélet szabályainak megfelelően, racionalizálják és végül a megvalósításhoz szükséges adatbázis tervét alakították ki. A folyamatok meghatározása ezután következik - az adattípusok egyes példányainak aktualizálása, törlése a modellnek megfelelően. Ez a megközelítés csökkenti az adatok redundanciáját, a kétértelműségeket, ön-ellentmondásokat (inkonzisztencia).

Egyes szerzők szerint ha az adatok integrációja (harmonizálása és összhangba hozása) történik meg először, akkor a szervezeti folyamatokra és tevékenységekre ez egy szükségtelen korlátozást jelent, előkészítés nélküli kényszer korlátokat jelent a folyamatokra, illetve ezek összehangolására. Más szerzők ezt kifejezetten előnyösnek tartják, mivel az adatszerkezet időben, stabilan változó, módosítása nagy költségeket igényel szemben a folyamatokéval, ezért az 'egészséges' adatmodellnek kell az integráció és harmonizációs munka központjában állnia.

A funkcionális megközelítés

A rendszert ebben a megközelítésben a funkciók hierarchiájának fogják fel. Funkciók alatt bizonyos bemenetek bizonyos kimenetekké történő *transzformációját*, átalakítását értik. A rendszer teljes egészét mit egy funkciót fogják fel és lépésenként bontják, finomítják részfunkciókká. Természetesen ebben a folyamatban a funkciók be- és kimeneteit valamint a rész- vagy alfunkciókat meghatározzák. A részfunkciók definíciói ekkor tisztázódnak, pontosan és egyértelműen definiálják őket.

A rendszerszemléletű és a funkcionális megközelítés között az alapvető különbség az, hogy a rendszerszemléletű megközelítés először a rendszer határait definiálja a bemenetek és kimenetek értelmében, míg a funkcionálisban először a funkciókat és a funkciók meghatározása során / után a be- és kimeneteket határozza meg; valójában két teljesen eltérő megközelítés.

1.3 A módszertanok osztályozása

Yourdon szerint egy módszertant egy lépésről-lépésre vonatkozó csatatervnek, míg egy módszert az átfogó módszertanban kijelölt legfontosabb végrehajtandó technikai lépések sorozatának tekinthetünk. Ez a megfogalmazás tartalmazza a módszertan *meta*, leíró jellegét (*terv*) és azt is, hogy elemibb dolgokból épül fel, amiket módszereknek tekinthetünk, és amelyek a konkrét, egyedi megvalósulásuk, példányuk a módszertanban előírtaknak megfelelően jön létre minden konkrét (projekt) esetben.

A módszertanokat a következőképpen is osztályozhatjuk:

- tudományos alapúak: idetartoznak a hagyományos, az adat-központú, strukturált vagy objektum-orientált módszertanok.
- Ezeknek az alapfeltételezése, hogy a 'Valóság' létezik, és megfogható, elemezhető elemekből áll, amelyekre szilárd nem változó törvények, tények érvényesek. Az egyik alap jellegzetességük, hogy az "oszd meg és uralkodj" elvét alkalmazva az elemzendő rendszert részekre bontják, amelyek könnyebben és jobban kezelhetők. Ennek a következménye viszont az, hogy ezek a módszerek, módszertanok nehezen alkalmazhatók a puha, nem szabatos, homályosan megfogalmazott, nem strukturált problémák elemzésére. Tehát ahol az elemzendő problématerület 'fuzzy', nem körvonalazott, eléggé pontosan és ezért nehéz azonosítani, felismerni és ezt többé-kevésbé formálisan megfogalmazni ott más módszertani megközelítéseket kell alkalmazni.

- rendszer szemléletűek: az általános rendszerelmélet alapú, a résztvevői megközelítés, az emberi tevékenység, munkafolyamat elemzésére alapuló módszertanok tartoznak ide.
- Az ún. 'puha megközelítésű' ('soft approach') információrendszer fejlesztési módszerek tartoznak ide. Az alapvető filozófiai különbség az, hogy ezek a módszerek a 'Valóság'-ot szubjektív módon létezőnek tekintik, az egyes szereplők, személyek érzékelésében, felfogásában élőknek. Ezért egy 'Szervezeti Valóság' leírást az egyes valóság felfogások összenövesztésével, összehangolásával lehet elérni. A nézetek összeillesztése természetesen feltételezi az egyes szereplők aktív részvételét. A rendszer szemléletű megközelítés a rendszert mint egészet fogja fel (holisztikus), és a szinergizmus elvére alapozva — azaz a kis rendszerek összenövése egy nagyobb rendszerré teljesen új tulajdonságok megjelenéséhez vezet, olyanokhoz, amelyek az alkotó részekben nincsenek jelen, vagyis az egész több mint a részeinek egyszerű összege. Ezen módszereket támogatók szerint pontosan ez hiányzik a tudományos megközelítésből, a részekre bontás során elveszítik annak a lehetőségét, hogy felismerjék ezeket a 'újonnan keletkező' rendszer sajátosságokat.

A módszertanok osztályozására, jellemzésére kifejlesztettek finomabb elemzési kategóriákat is ([Layzell89]).

Filozófia: *egy módszertan filozófiája alatt egy olyan általános állítást értünk, amely a módszertan legalapvetőbb megközelítésének indokát, módját tartalmazza. Gyakran az elemzés/tervezés alapvető stratégiájának megfogalmazását, a módszertan központi fogalmát írja le. (A rendszer modellezés mögött meghúzódó filozófia.)*

Például:

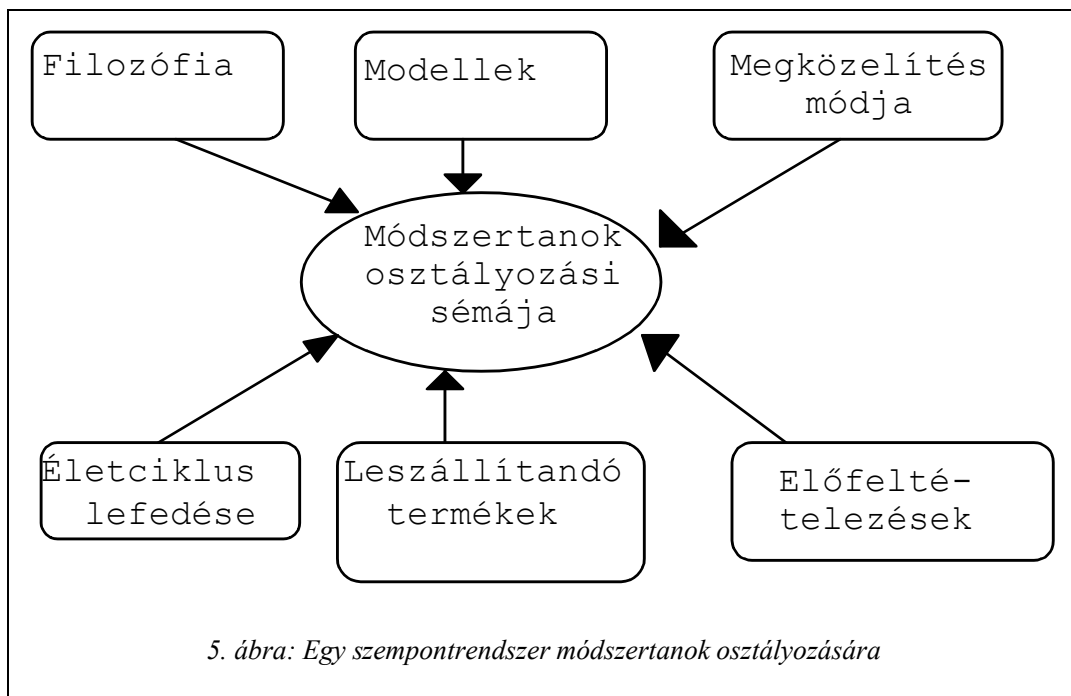
- funkcionális dekompozíció,
- alulról - felfelé vagy felülről - lefelé haladó,
- strukturált,
- rendszer szemléletű
- objektum-orientált, stb.

Megközelítés: a módszertan javasolta munkafolyamat jellemzése. Bizonyos esetekben ez a lépések megváltoztathatatlan, rugalmatlan sorozatát jelentheti, míg más esetekben, a módszertan csupán elemző eszközök, modellező nyelvek halmaza, amelyből a módszertan felhasználója tetszése szerint választhat anélkül, hogy egy szigorúan előírt munka vagy technológiai folyamatot kellene követnie. (Az eszközök használatának eljárásai, a fejlesztési folyamat tervezésének, irányításának, ellenőrzésének módja, a feladatok és a fejlesztésben résztvevők összerendelése.)

Modellek: a módszertanban használt modellezési eljárások, amelyekkel az elemzett rendszerről rögzítik a tényeket. A modellekben tartalmazzott feltárt tényeket a felhasználó tudomására a módszertanban előírt, leszállítandó termékek formájában hozzák, egy előre meghatározott jelöléstechnikát alkalmazva. (Eszközök és technikák halmaza, amelyeket a dokumentációk előállítására használnak)

Például:

- adatmodell,



- adatfolyammodell,
- eseménymodell.

Életciklus lefedése: az információrendszer fejlesztés életciklusának mekkora részét, milyen hányadát fed le az adott módszertan. Kiterjed-e a teljes életciklusra vagy csak egy meghatározott részben nyújt segítséget.

Leszállítandó termékek: A módszertan kézzelfogható végeredménye, amely szorosan kötődik a használt modellekhez, modellezési eljárásokhoz.

Előfeltételezések: *a módszertanok (illetve kifejlesztők) különböző előfeltételezésekkel élnek, például azokra a rendszerekre vonatkozóan, amelyekkel a módszertan foglalkozni szándékozik. Ezeket a feltételezéseket időnként nyíltan megfogalmazzák, máskor a módszertanban és annak gyakorlatában rejtve marad. Például információrendszerre vagy valós-idejű rendszerre alkalmas módszertanok.*

Az egyes módszertanok rövid ismertetésénél utalni fogunk erre az osztályozási szempontrendszerre, megpróbáljuk elhelyezni a módszertanokat ezekben a dimenziókban.

A módszertanok célkitűzései

Az információrendszer fejlesztési módszertanok célja két oldalú, egyrészt a fejlesztési folyamat irányításához és tervezéséhez szükséges vezetési/irányítási struktúrát próbálja meghatározni, másrészt a szoftver minőségét kísérli meg növelni/javítani a rendszerspecifikáció készítésére bevezetett formális módszerek segítségével.

Az első pont annak a felismerése, hogy a jelenlegi nagy, bonyolult rendszerek teljes megértése, a követelmények és az egyéb szempontok felfogása messze meghaladja egyetlen ember képességét. Ez teszi szükségessé a csoportmunkát és ily módon a jól definiált lépések sorozatát, amely a megfelelő koordinációt teszi lehetővé a csoport tagjai között és egyúttal a munka ellenőrizhetőségét és átláthatóságát is biztosítja.

A második pont a kifejlesztendő rendszer valódi megértését, átlátását célozza meg, adekvát absztrakciós szintek kialakításával, a követelményspecifikációtól kezdve a programkód generálásig a kapcsolódó dokumentációkkal, amelyek a csoport tagjai közötti hatékony és eredményes kommunikációhoz szükségesek.

A módszertanok között a következő közös jellemzők fedezhetők fel:

- a megvalósítandó rendszer logikai modelljét készítik el, majd ezt alakítják át fizikai modellel ökölszabályok, tervezési heurisztikák alkalmazásával,
- diagramtechnikákat, grafikus ábrázolási eljárásokat használnak, lehetőség szerint kerülnek a tisztán szöveges leírásokat,
- a modellek helyességének ellenőrzésére, világos, egyértelműen lefektetett szabályrendszerek vannak,
- a specifikálást eszközök támogatják, diagramtechnikák, szöveges specifikációs nyelvek és a metaadatok formalizált rögzítése adatszótárban.

Összefoglalva, a mai modern rendszerfejlesztési módszertanokat az különbözteti meg lényegesen a hagyományos eljárásoktól, hogy sokkal nagyobb hangsúlyt helyeznek a fejlesztés korai szakaszaira. Ezért ezeknek a módszertanoknak az a fő célja, hogy a követelményspecifikáció és a rendszertervezés fázisában a felbukkanó hibák számát minimalizálják és ezáltal növeljék a fejlesztő csoport teljesítményét és a leszállított szoftver minőségét.

1.4 Információrendszerek szervezetekben

Információ a kommunikáció bármely formája lehet, akár formális akár informális, amely egyben egy rendszer koherens viselkedése megértésének kulcsa.

Definíció 1-4 Információrendszer

Információrendszer a szervezet olyan része, mely információt szolgáltat, létrehoz, tárol, szétválogat, használ és eloszt. Emberi, műszaki és pénzügyi / gazdasági alkotórészekből, erőforrásokból áll. Tulajdonképpen eredendően egy humán rendszer (szervezet, manuális rendszer), mely esetleg tartalmaz egy számítógéprendszert, és ez az információrendszer bizonyos jól meghatározott részeit, kiválasztott elemeit automatizálja. Célja, hogy egy

szervezet vezetési / irányítási funkcióit, valamint a mindennapi működését egyaránt támogatják

Információrendszert a következők jellemzik:

bemenetek

kimenetek

feldolgozási folyamatok

tárolt adatok (adat)állományok)

A rendszerelemző feladata a fentebbi jellemzők meghatározása.

Egy szervezeten belül további (rész)rendszereket lehet felismerni, és hasznos ha a rendszerelemző tudatában van ezek létezésének, még akkor is ha ezek részletekbe menő leírása nem a feladata.

Funkcionális Rendszer a szervezetnek az a részrendszere, amelynek keretében végzik az emberek azokat a tevékenységeiket, amely a szervezet céljainak megvalósítását szolgálja. Ez az a részrendszer, amely számítógéprendszer tartalmazhat, azonban az emberi tevékenységnek ez a számítógéprendszer csak egy kis részét fedi le. Más egyéb segítő adminisztratív, emberi rendszerfelületekről kell gondoskodni, amelyek lehetővé teszik, hogy egy számítógép rendszer helyesen működjék (vagy hogy egyáltalán működjék).

Szociológiai Környezet az emberek értékeivel, értékrendszerével foglalkozik. Mit tartanak jónak, rossznak egyenrangú vagy magasabb rangú kollégáikkal való kapcsolatukban. Egy számítógép rendszer bevezetése erőteljesen befolyásolja ezeket a viszonyokat, a munkaköröket és a hatás - és feladatköröket.

Politikai Rendszer az emberek jólétével, szociális problémáival foglalkozik. Milyen a szakszervezetek egymás közötti kapcsolata és viszonya a vezetéshez? A számítógép hatása a személyes érdekekre, a karrier elképzelésekre, a szervezeten belül elfoglalt társadalmi státuszra komoly hatást gyakorol és kritikus tényező lehet a rendszer elfogadása, befogadása, a későbbiekben pedig a biztonsága szempontjából.

Az információk között is célszerű különbséget tenni, két legfontosabb osztálya a feldolgozandó információknak:

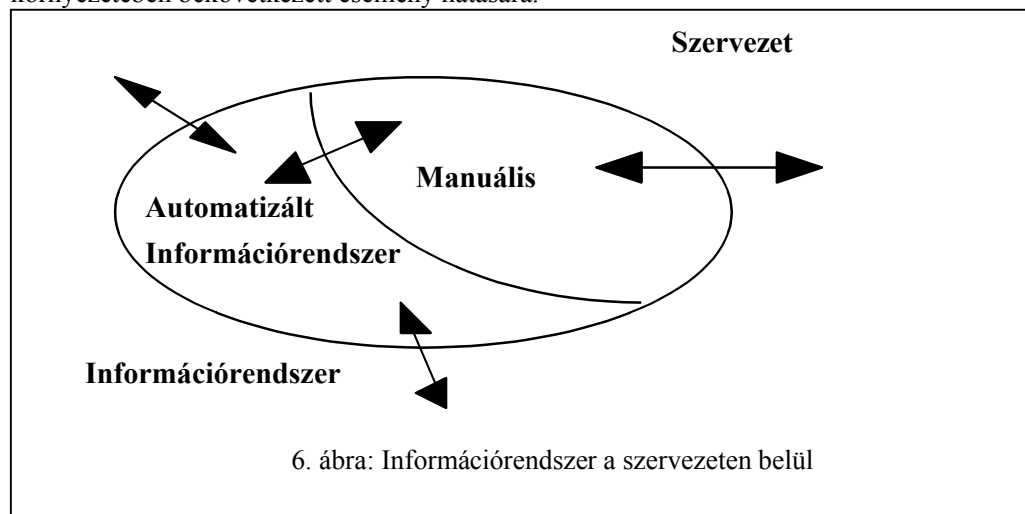
Működési információk, azok, melyeket rutinszerűen használ a szervezet, és a napi feladatainak ellátásához feltétlenül szükség van

Vezetési / Irányítási információk, azok, melyek a különböző vezetői szintek számára a döntési folyamatban fontos szerepet játszanak, és kevésbé rutinszerűek mint a működési információk.

Informatikai szempontból az automatizált adatintenzív, tranzakció-orientált rendszerek érdekesek.

Adatintenzív rendszerek, azok a rendszerek, melyek az *állandó jellegű*, működési információkkal dolgoznak, amelyek a szervezet számára *közösek* és *integráltak*. Az állandóság itt azt jelenti, hogy egy bizonyos ideig tárolni kell még, mert ezek az adatok a szervezet folyamataival, tevékenységeivel függenek össze. Az integráció azt, hogy az adatok teljes halmaza úgy tekinthető mint több kisebb, egyébként különálló adathalmazok egyesítése, továbbá egymással összhangban vannak és nincs köztük ellentmondás. A közös adatok azt jelentik, hogy az integrált adathalmaz egyes elemeit sok különböző felhasználó együttesen használhatja.

Tranzakció-orientált rendszerek, olyan rendszerek, amelyek az adatok egy adott állapotának az átalakításával (*transzformáció*) foglalkoznak, a szervezetben illetve környezetében bekövetkezett esemény hatására.



Az informatikában megkülönböztetnek *Vezetői Információs Rendszereket* (**MIS, Management Information Systems**), amely tulajdonképpen gyűjtőfogalma az összes alkalmazási (rész)rendszernek, valamint *Döntéstámogató Rendszereket* (**DSS, Decision Support Systems**).

Az adatintenzív tranzakció-orientált rendszereket innentől kezdve információrendszernek fogjuk hívni a rövidség kedvéért. Néhány alapvető fogalom, amely implicit módon hozzátartozik ezekhez az információrendszerekhez a következő: az adat átalakításoknak

konzisztensnek (ellentmondásmentesnek) kell lenniük, azaz bizonyos előírt szabályoknak meg kell felelniük az átalakítás utáni állapotoknak,

febonthatatlanság azt jelenti, hogy a tranzakció vagy teljes mértékben végrehajtható vagy egyáltalán nem,

tartósság, vagyis az adat-átalakítás után az eredmény nem tehető semmissé.

Ahogy ezt már előbb megállapítottuk az információrendszerek *diszkrét* rendszerek, azaz diszkrét elkülöníthető állapotaik vannak, még ha nagy számú is; ezek az állapotok az adatok, adatelemek állapotainak, értékeinek kombinációjával foghatók meg. A tranzakciók ezeken az adatokon okoznak állapotváltozást, a rendszert az egyik diszkrét állapotából a másik állapotba viszik át.

Információrendszerek fejlesztése

Machiavelli (1513):

Semmi sem lehet nehezebb, a sikere kétségesebb, sem veszélyesebb mint megtervezni és végbevinni egy új rendszer létrehozását. A kezdeményezőnek minden olyan ember ellensége, aki bármely hasznot is húz a régi rendszer megőrzéséből és csupán langyos védelemre számíthat mindazok részéről, akik az új rendszer bevezetésével nyernének.

A fentebbi idézet az új rendszerek kifejlesztésének nehézségével szembesíti a leendő rendszer készítőket. Tehát nemcsak a rendszer bonyolultsága okoz problémát, hanem más tényezők is. Ebben a szekcióban az információrendszer fejlesztéssel kapcsolatos legfontosabb fogalmakat fogjuk áttekinteni, amelyek természetesen különböző módszertanokhoz is kapcsolódnak.

1.5 Információrendszerek modelljei és ábrázolásai

Modellek

A leendő információrendszer részletes, megvalósítható leírását, modellekben fogalmazzuk meg. A tapasztalat azt mutatja, hogy egy lépésben - az információrendszerek komplexitása miatt -, egyetlen formalizmust használva nem lehet olyan pontossággal lefedni a rendszer összes lényeges oldalát, mely kielégítő lenne a rendszer kivitelezéséhez, tehát a bonyolultság kezelésére használunk modelleket.

Az elemzés és a leendő információrendszer különböző oldalainak leírására a következő modellek használhatók:

- *szervezeti* modell, amely a munkamegosztást és a munkamódszerét, - módját írja le a valóságban; az esetleg létező Szervezeti és Működési Szabályzat, a munkaköri leírások hasznos kiinduló pontul szolgálhatnak;
- *információ* modell, ez mutatja be a rendszer információit, azok tartalmát és származtatását a valósághoz kapcsolódóan;
- *adat* modell, amely azokat a valóságban létező fogalmakat, objektumokat, entitásokat fogja össze, amelyekről eltárolnak bizonyos *információkat* és a köztük fennálló *kapcsolatokat*;
- *folymat* modell, amelyben a való világ szerkezete, tevékenységeinek irányítási struktúrája lesz megfogalmazva.

Természetesen ezek a modellek egymással kapcsolatban állnak, explicit vagy implicit módon megkövetelik az integráltságukat, az elemeik között fennálló ellentmondásmentességet és összhangot, bizonyos kis mértékű átfedések és redundancia viszont elfogadható.

Ábrázolások

Fogalmi vagy logikai ábrázolás. Ez az ábrázolási mód a fentebbi négy modellt absztrakt, de a felhasználók számára érthető, döntően nem műszaki szakkifejezések segítségével írja le.

A *szervezeti* modell ebben az ábrázolásban a szervezet tevékenységeit tartalmazza, valamint az információkhoz illetve anyagokhoz való viszonyát.

Az *információ* modell azokat az információkat gyűjti össze, amelyek a szervezeten belüli káosz csökkentéséhez (az entrópia növekedés fékezéséhez) vezetnek, valamint az információk forrását és származtatási / levezetési módját.

Az *adat* modell, a szervezet tevékenységei közötti kommunikáció megvalósításához eltárolandó adatok meghatározását és az adatok között fennálló kapcsolatokat adja meg.

A *folyamat* modell a szervezeti tevékenységekben végrehajtott folyamatokat mutatja be, amelyek a szervezet működtetéséhez, irányításához szükségesek és a szervezeti valamint jogi szabályozást jelenítik meg.

A *műszaki vagy fizikai ábrázolás*. A négy modellnek ebben az ábrázolásában a rendszernek azt a részét jelenítjük meg, amelyet számítógép fog végrehajtani vagy segíteni, és műszaki / technikai vagy fizikai kifejezéseket használnak. Azt mutatja meg, hogy a felhasználó számára, hogyan fog kinézni a rendszer műszaki vagy fizikai szempontból.

A *szervezeti* modell a számítógépesített tevékenységek számítógép tranzakcióit, programjait fogja ábrázolni, beleértve a szervezet telephelyeit összekötő számítógép hálózatot és az ezekhez kötődő, a felhasználók által elvégzendő, de nem számítógépesített tevékenységeket.

Az *információ* modell az információk megjelenítését, megjelenítési formáját mutatja be.

Az *adat* modell az eltárolandó adatok leírását az adatbáziskezelők műszaki nyelven fogalmazza meg.

A *folyamat* modell a program terveket önti formába, a bemenetek, kimenetek, és a moduláris szerkezetük értelmében. Továbbá az egymással kommunikáló számítógép folyamatokat az adott adat kommunikációt kezelő rendszernek megfelelően mutatja be.

A *megvalósítás ábrázolása*. A négy modellnek ez az ábrázolása azt írja le, hogy a leendő rendszer hogyan valósul meg számítógépeken, terminálokon, hálózatokon, adatállományokon és adatbáziskezelő rendszereken, stb.

A *szervezeti* modell a tevékenységeket a szervezeti környezetbe ágyazva érzékelteti: a szervezeti egységek a jelentési kötelezettségek, az alá-fölé rendeltség hierarchiájában, a szervezeti egységek és az alkalmazottak funkciójából származó *feladatok (task)* értelmében. Továbbá az összes feladatra vonatkozóan a felhasználók által követendő eljárásokat. A kommunikációs hálózat szintén ehhez az ábrázolási módhoz tartozik.

Az *információ* modell az információkat a programozási nyelvek formalizmusának megfelelően adja meg, azaz deklarációk formájában.

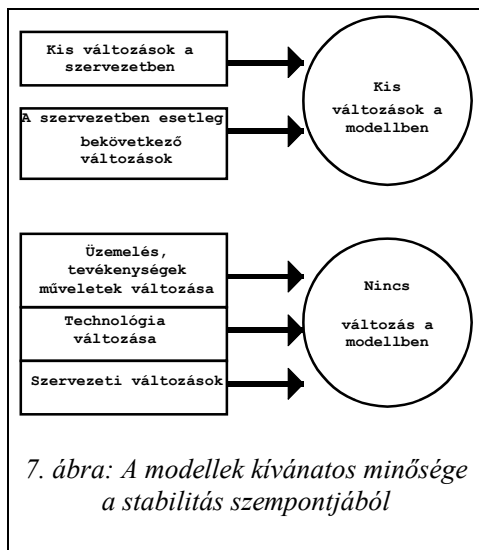
Az *adat* modell azt mutatja be, hogy a háttértárakon (lemez, szalag, egyéb) milyen módon tárolják el az adatok szerkezetét és a köztük fennálló kapcsolatokat.

A *folyamat* modell az adatfeldolgozást valamilyen programozási nyelv formájában jeleníti meg, ebbe beleértve az adatkommunikációt.

A modellek stabilitásának elemzése

Definíció 1-1 Stabilitás

A modell stabilitása azt jelenti, hogy a modell képes legyen a külvilágban bekövetkező megrázkódtatásokra, zavarokra, kiszámíthatatlan jelenségekre úgy reagálni, hogy azokat befogadja, igazodjon hozzájuk (adaptivitás). (12. ábra)



7. ábra: A modellek kívánatos minősége a stabilitás szempontjából

A stabil modell olyan, hogy a jövőben nincs szükség indokolatlan módosítására, a karbantartás hatásának inkább mérséklőnek, fékezőnek mintsem felerősítőnek kell lennie. A stabilitás elemzésnek a célja az, hogy megerősítse azt, hogy a szervezet életében bekövetkező kis változások legfeljebb a modell csekély változtatásához vezetnek. (Bizonyos változások a szervezetben nem követelik meg a modell megváltoztatását, de ezt nem várhatjuk el minden esetben, kis mértékű változtatásokat el kell tudni viselni.)

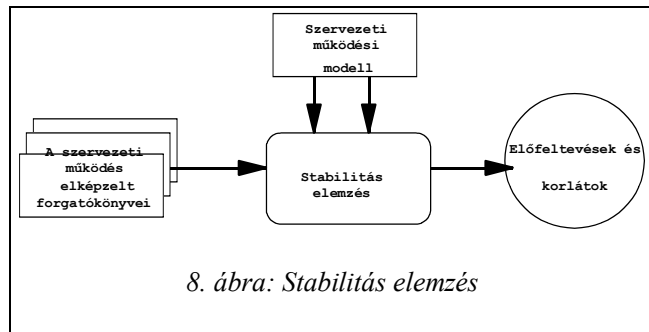
A szervezet modelljének nem szabad sem az alkalmazott technológiából származó korlátozásokat tartalmaznia, sem az eljárásrendből és a szervezet felépítéséből származókat, függetlennek kell lennie az osztályok közti felelősség felosztástól is. Így bármilyen változás az

előbbieken nem vezet a modell megváltozásához. Ha például a modell feltételezi, hogy a befizetések postait csekken érkeznek, akkor a modell elavulttá válhat, amikor a szervezet áttér az elektronikus pénzügytatalásra.

A stabilitás elemzés a következőket jelenti (14. ábra). Az elképzelt jövőnek megfelelő forgatókönyveket készítenek a szervezet működéséről: a szervezeti célkitűzések módosulása, a működési szabályok változása, a külső környezet változásai (beleértve a lehetséges törvény módosulásokat, új törvényeket). Ezeket összevetik a szervezet modelljével és megvizsgálják, hogy a modell mennyire tud megbirkózni ezekkel a hipotetikus változásokkal.

Az egyes forgatókönyvek bekövetkezésének valószínűségét is meg kell becsülni. Sokszor hasznos ha a modellt úgy módosítják, hogy az képes legyen alkalmazkodni a azokhoz a helyzetekhez, amire számítanak. Természetesen nem lehet az összes lehetséges forgatókönyvre felkészülni (azaz előre gondoskodni a lehetséges problémák leküzdéséről). A stabilitás elemzés így egyenesen elvezet a szervezet modelljének felülvizsgálatához és ugyanakkor kiemeli és láthatóvá teszi a modellben benne levő előfeltevéseket és korlátokat. Ezeket a felismert feltételeket és korlátokat közölni kell a vezetéssel azért, hogy figyelembe tudják venni a tervezésnél.

De magától értetődő módon a stabilitás elemzés akár a többi ellenőrzési, megerősítési feladat és maga a modellezési tevékenység is egy végtelen feltáró, felfedező jellegű munka és nem pedig egy véges lépéssorozatból álló konstrukciós feladat. Vagyis más szavakkal, bármennyit is dolgozik a rendszerelemző mindig van lehetőség egy kicsit többet csinálni, és lehetséges, hogy még mindig rejtve maradt valamilyen nagyon fontos tény. Ezért annak a meghatározása, hogy mikor fejezzék be az elemzést sokkal inkább vezetői döntés semmint műszaki, informatikai.

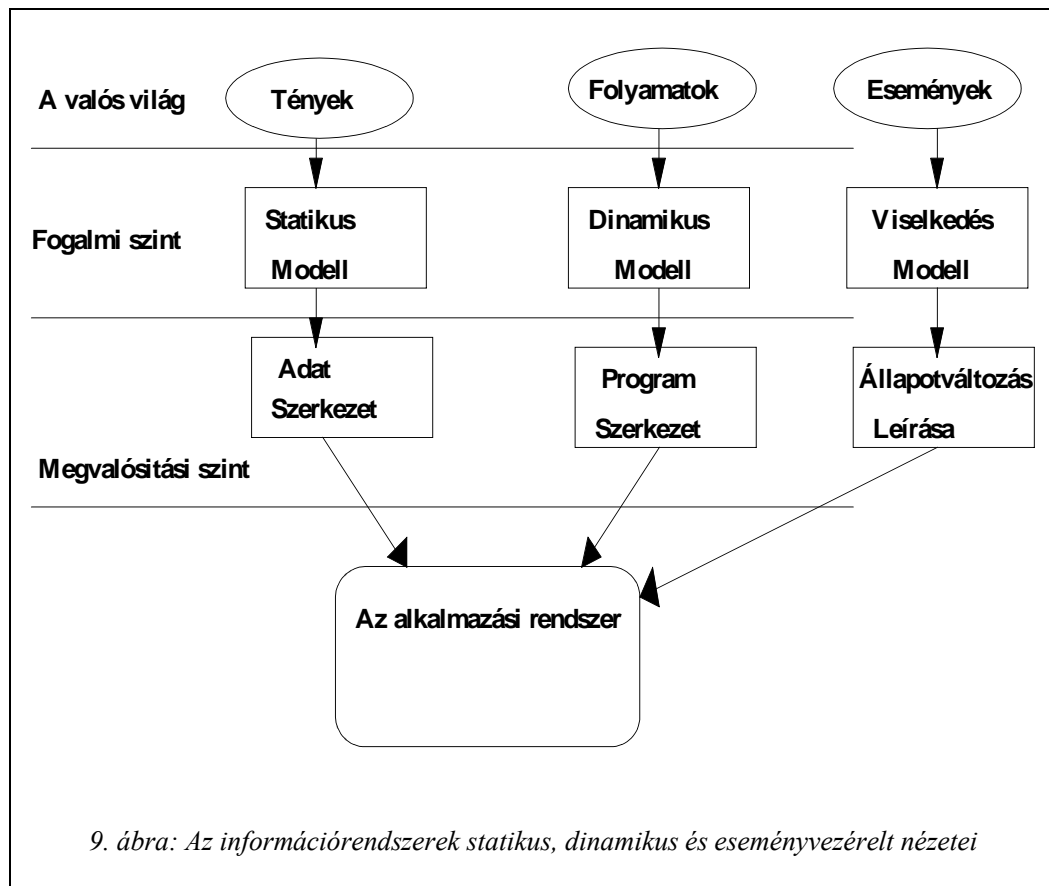


8. ábra: Stabilitás elemzés

1.5.1.1 jövő modellezése

A gazdasági racionalitás és a rugalmasság között ésszerű kompromisszumot kell kötni - a jelenlegi követelményekhez történő igazodás csorbíthatja az adaptációs képességet a jövőbeni követelményekhez. Itt fellép egy időzítési probléma, a lehetséges jövőbeni követelményeknél, mind a funkcionális mind a nem-funkcionális követelményeknél.

Ha valamit most nem kérnek, de lehet, hogy szükség lesz rá a jövőben, kell-e modellezni? Vagyis érdemes-e a modellünket most elbonyolítani azért, hogy bizonyos módosításokat később könnyebbé tegyünk, például elébe menjünk az adatok mennyisége növekedésének.



9. ábra: Az információrendszerek statikus, dinamikus és eseményvezérelt nézetei

Megalapozott döntéshez szükség van a módosítások költségére. Relációs adatbáziskezelő alkalmazásával, vagy modern CASE eszközök használatával könnyebb újabb objektumokat a modellhez hozzáilleszteni mint a meglévő modellben fennálló szemantikai és összhangot

garantáló kapcsolatok jelentését megváltoztatni. Ugyancsak tudni kell, hogy mi a valószínűsége a leendő követelmények megjelenésének, milyen gyakran fordul ez elő, és mennyire pontosan lehet ezt előre specifikálni? (Ha valószínűtlen, távoli és homályos, felejtsek el!)

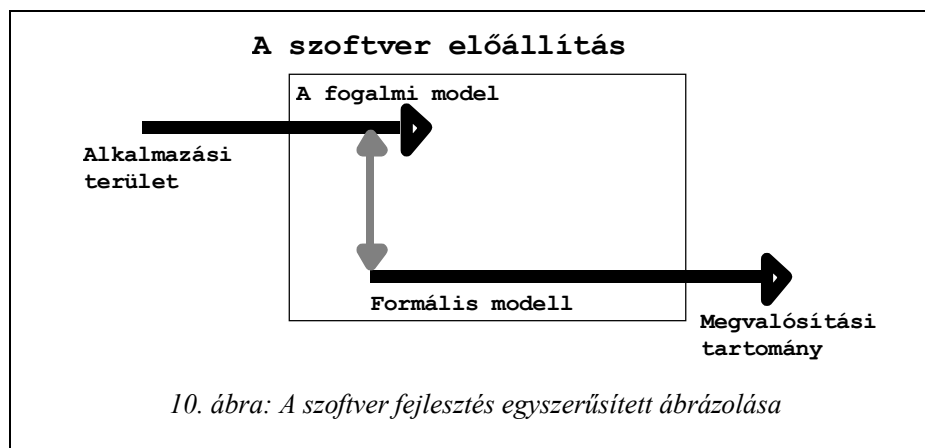
Az információrendszerek fejlesztésének főbb elvei

Az információrendszerek fejlesztésének egyik lehetséges módja az életciklus megközelítés (ld. 0) a másik a prototípuson alapuló evolúciós. Gyakran használják a prototípusokat az életciklus modelleken alapuló fejlesztések kiegészítéseként, ezért nem zárjuk ki teljesen a prototípusra alapuló módszereket, de ebben a jegyzetben főleg a strukturált, életciklus modellekre koncentrálnunk. A fejlesztés két nagy fázisát különböztethetjük meg:

A részletekre bontás szakasza, amikor a rendszert felbontjuk értelmes részrendszerekre, ezek kapcsolatát elemezzük és leírjuk.

A szintézis szakasza (kivitelezés), amikor a rendszert alkotó részeket összekapcsoljuk, szintézis útján egy új rendszert hozunk létre.

Az információrendszereket felfoghatjuk tehát úgy, mint a valóság egy szeletének az



ábrázolását, az adott szervezet egy darabjának a leírását, vagyis a létező tények és a valóságban végbemenő folyamatok megfogalmazásának. Ezért az információrendszerek fejlesztése tulajdonképpen modellezési, modell leírási probléma.

Ezeket a modelleket besorolhatjuk két vagy három főbb kategóriába.

A rendszerek dinamikus oldalának lehet tekinteni folyamatokat, tevékenységeket, adatfolyamokat, tranzakciókat. Ezt a nézetet a használat nézőpontjának nevezhetjük, azaz azt tükrözi, hogy 'hogyan segítik a szervezet működését'. A statikus oldalhoz sorolhatjuk a tárgyakat, objektumokat, kategóriákat és a köztük fennálló kapcsolatokat, azaz azokat, 'amik segítik a szervezet működését'. Egyre terjed az a felfogás, hogy érdemes külön venni és megkülönböztetni a szervezeti eseményeket, az ezek által kiváltott informatikai eseményeket, amelyek valamilyen változtatást, átalakítást kezdeményeznek az információrendszerben, és az események által beindított folyamatokat. Ezt az oldalt a rendszer viselkedésének nevezhetjük, vagyis az eseményekre adott rendszer választ tükrözi, vagyis 'hogyan reagál a rendszer a szervezet környezetében bekövetkező eseményekre'.

A fentebbi ábrán (10. ábra) az automatizált, számítógépes információrendszerek szoftver fejlesztés folyamatára egy sematikus ábrázolását mutatjuk be. Ez az ábrázolás rámutat annak a ténynek a fontosságára, hogy először egy olyan modellt kell kialakítani, amely az alkalmazási terület megértését tükrözi vissza (vagyis a jellegében kognitív) azután egy sor

átalakításon keresztül egy sokkal formálisabb modell sorozatot alakítanak ki, amelyek megfogalmazását már befolyásolják a fizikai megfontolások.

Emberi szerepek a fejlesztésben

A rendszerfejlesztés során sok emberi szereplőnek kell részt vennie ebben a folyamatban. Erre már több ponton utaltunk. A következő lista nem biztos, hogy teljes, de legalábbis megpróbálja a felismert és azonosított szerepeket lefedni a jelenlegi tudásunk szerint.

- igazgatóság/vezetőség részéről kinevezett felelős,
- fejlesztési koordinátor,
- rendszerszervező (üzleti elemző, business analyst),
- rendszertervező,
- felhasználói képviselő / átvevő,
- felhasználó
- kivitelezési terv átvevője,
- kivitelező,
- erőforrás menedzser.

A gyakran használt *fejlesztő* szerepkörét ebben a terminológiában a tervező és a kivitelező fogalma fedi le. A *megvalósító* több értelemben is használt fogalmát a kivitelező szerepköre foglalja magában itt. Gyakran a megvalósító és megvalósítás alatt azt értik, hogy egy megtervezett és már elkészített rendszer egy példányát üzembe helyezi. A *projektirányító* szerepköre ebben a felsorolásban felbomlik a fejlesztési koordinátor és az erőforrás menedzser szerepére. Röviden megpróbáljuk ezeket a szerepeket körülírni. Azt látni kell, hogy a különböző információrendszer fejlesztési és projektirányítási módszerek saját elnevezéseket alkalmaznak a fentebbi szerepkörökre, de ez az osztályozás elméleti szempontból elég átfogónak tűnik.

1.5.1.2 Igazgatóság/vezetőség részéről kinevezett felelős

Az igazgatóságnak, igazgató tanácsnak felelős személy, akinek a feladata az egész információrendszer sikeres kialakítása (felelős az egész 'IR adaptáció'-ért lsd. Euromethod).

1.5.1.3 Fejlesztési koordinátor

Nagy információrendszer fejlesztési projekteknél, ahol a projekt szerepköröket különböző személyek és csoportok töltik be, szükség van egy koordinátorra, aki megvalósítja a projekt különálló részei között az összeköttetést napról napra és az igazgatósági felelősnek megküldi a jelentéseit. Egyes szervezeteknél, a fejlesztési koordinátor egynél több projektet is felügyelhet.

1.5.1.4 Rendszervező (üzleti elemző, business analyst)

A szervezetet, a szervezet működését, tevékenységét elemzi annak érdekében, hogy létrehozson egy olyan kiindulópontot, amely az informatikai rendszer kialakításának alapjául szolgálhat. Főtevékenysége az üzleti és / vagy működési szabályok feltárása, a szervezeti, szervezési, gazdasági, szociális, szociológiai szempontból. Erre a feladatra sok technika áll rendelkezésre.

1.5.1.5 Rendszerelemző (system analyst)

Nehéz elhatárolni a tevékenységét a fentebbi szereplőtől, de talán abban lehetne megragadni, hogy az informatikai előképzettség és tapasztalat elengedhetetlen az információrendszerek területén. Főtevékenysége az üzleti és / vagy működési szabályok feltárása és dokumentálása az informatika keretén belül elfogadott eljárásokkal, technikákkal, lehetőleg széles körben olvasható jelölésrendszert használva. Gyakran a rendszerszervezési és rendszerelemzési feladatokat ugyanaz a személy látja el.

1.5.1.6 Rendszertervező

Ez a személy felelős a tervezési termékek előállításáért, azaz a leendő rendszer különböző szintű specifikációjáiért, amelyek alapján a rendszer kivitelezése létrehozható. Ez kifejezetten informatikai szakképzettséget igényel, specializálódva az információrendszerek készítésére, és a kapcsolódó technológiákra; mérnöki szabotosságú, tudományosan megalapozott eljárások, technikák alkalmazására kell képesnek lenni. Vannak olyan képzett informatikusok, akik egyszerre képesek a rendszerszervező, - elemző és - tervező szerepkörét betölteni.

1.5.1.7 Felhasználói képviselő / átvevő

Ahogy a neve is mutatja a felhasználói érdekeket képviseli elsősorban és esetleg az igazgatóság érdekeit is. Főfeladata a rendszer átadás / átvételhez szükséges felhasználói szintű dokumentáció leellenőrzése és elfogadása a rendszer kivitelezés megkezdése előtt.

1.5.1.8 Felhasználó

A felhasználó az a személy, aki a leendő információrendszert használja fogja, amint az rendelkezésre fog állni. Tipikusan sok felhasználó van, esetleg több ezer. Ilyen esetekben meg kell különböztetnünk a felhasználók típusait, szerepköreit. A szervezetet jól ismerő, az adott területen nagy jártasságot szerzett felhasználókat kell a tervezési folyamatba bevonni, a konzultációk, interjúk, minőségi szemlék során.

1.5.1.9 Kivitelezési terv átvevője

Ennek a szerepkörnek az a feladata, hogy a rendszer kivitelezőjének specifikációját felülvizsgálja a kivitelező szempontjából, és leellenőrizze, hogy minden előírásnak, szabálynak, szabványnak, ami a rendszerre vonatkozik, megfelel.

1.5.1.10 Kivitelező

Az a valaki, szervezet, aki a rendszer kivitelezését a terv specifikációkkal összhangban végrehajtja. Hagyományosan, a kivitelező egy alkalmazási program fejlesztő, programozó lehet, de a magas szintű nyelvek terjedése (4GL, stb.) a programozási tudás iránti igényt csökkentette.

1.5.1.11 Erőforrás menedzser

Az erőforrás menedzser felelős a szükséges erőforrások biztosításáért és ezen keresztül a fejlesztés zökkenőmentes előrehaladásáért. Ezt a szerepet össze lehet kapcsolni az igazgatósági képviselővel vagy a fejlesztési koordinátoréval.

1.6 Módszertanok a gyakorlatban

Az információrendszerek (iparszerű) készítésének nagyon fontos segédeszközei a szoftveréletről több fázisát, illetve egyes fázisait támogató módszertanok. Hosszú évek kutatásai/fejlesztései során alakultak ki a jelenlegi módszertanok jellemzői az ún. strukturált

módszerek alkalmazása, és a különböző diagramtechnikák, amik közérthető ábrák felhasználásával jelentősen megkönnyítik a műszaki/technikai szakemberek és végfelhasználók közötti kommunikációs szakadék leküzdését.

Egy felmérés szerint az Egyesült Államokban, Németországban, Nagy-Britanniában és Franciaországban a rendszerelemzési technikák nagymértékben elterjedtek. A felmérésben szereplő fejlesztésekben az ilyen technikák országokénti elterjedtségére a következő adatokat kapták:

Nagy-Britannia	33%
Franciaország	28%
Németország	16%
USA	17%

Az önkormányzatokra és az országos kormányzati szférákra vonatkozó megfelelő adatok a következők:

Nagy-Britannia	50%
Franciaország	20%
Németország	0%
USA	50%

Nagy-Britanniában, Franciaországban és Hollandiában egyértelműen vezető szerepet játszanak a kormányzati szabványok, nevezetesen az **SSADM**, a **MERISE** és az **SDM**. Vagyis nemcsak az országos és helyi kormányzatokban folyó rendszerfejlesztésekben, hanem más alkalmazási területeken is használják ezeket a módszereket, például pénzügyi és biztosítási rendszerek, ipari gyártórendszerek, kis- és nagykereskedelem, stb. Egy felmérés szerint az **SSADM** (40%+), a **MERISE** (45%), Hollandiában az **SDM** (40%) részesedést el.

A strukturált rendszerelemzési / - fejlesztési technikákat döntően nagy és közepes projektekben használták, de néhány kisebb munkában is. A fejlesztői csoportok létszáma 30 főnél kevesebb volt a felmérésben szereplő rendszerek 25%-nál.

Strukturált rendszerelemzési / - fejlesztési technikákat alkalmazók több mint fele használt valamilyen CASE eszközt.

Ezek a módszertanok nagyon sokat segítenek az ún. szoftver krízis leküzdésére, amit a következő statisztika illusztrál:

A fejlesztők, cégek által valamilyen formában átadott alkalmazás fejlesztések, szoftverrendszerek használatba kerülésének statisztikája a Pentagonnál:

47%	leszállított, soha nem használt
29%	kifizetett, de soha le nem szállított
19%	átdolgozott, vagy kidobott
3%	használták a változtatások elvégzése után
2%	volt úgy használva, ahogy leszállították

(forrás: U.S.A: hadseregének statisztikája)

A hardver elveszti meghatározó szerepét. A legutóbbi időig a kiválasztott hardver jellemzői határozták meg a rendszerfejlesztés irányelveit. Azaz a hardverrel kapcsolatos költségek diktálták a feltételeket, aminek eredménye gyakran az volt, hogy a felhasználói követelményekből kellett engedni. Az elmúlt évek hihetetlen gyorsaságú hardver áresése következtében a hardverrel kapcsolatos szempontok nem játszanak elsődlegesen meghatározó szerepet, nem akadályozzák és nem is korlátozzák a megfelelő megoldás kialakítását.

A hardver várható élettartama jelenleg	5 év,
egy alapszoftveré (operációs rendszer, stb.)	10 év,
alkalmazást készítését segítő szoftver (adatbáziskezelő, stb.)	15 év,

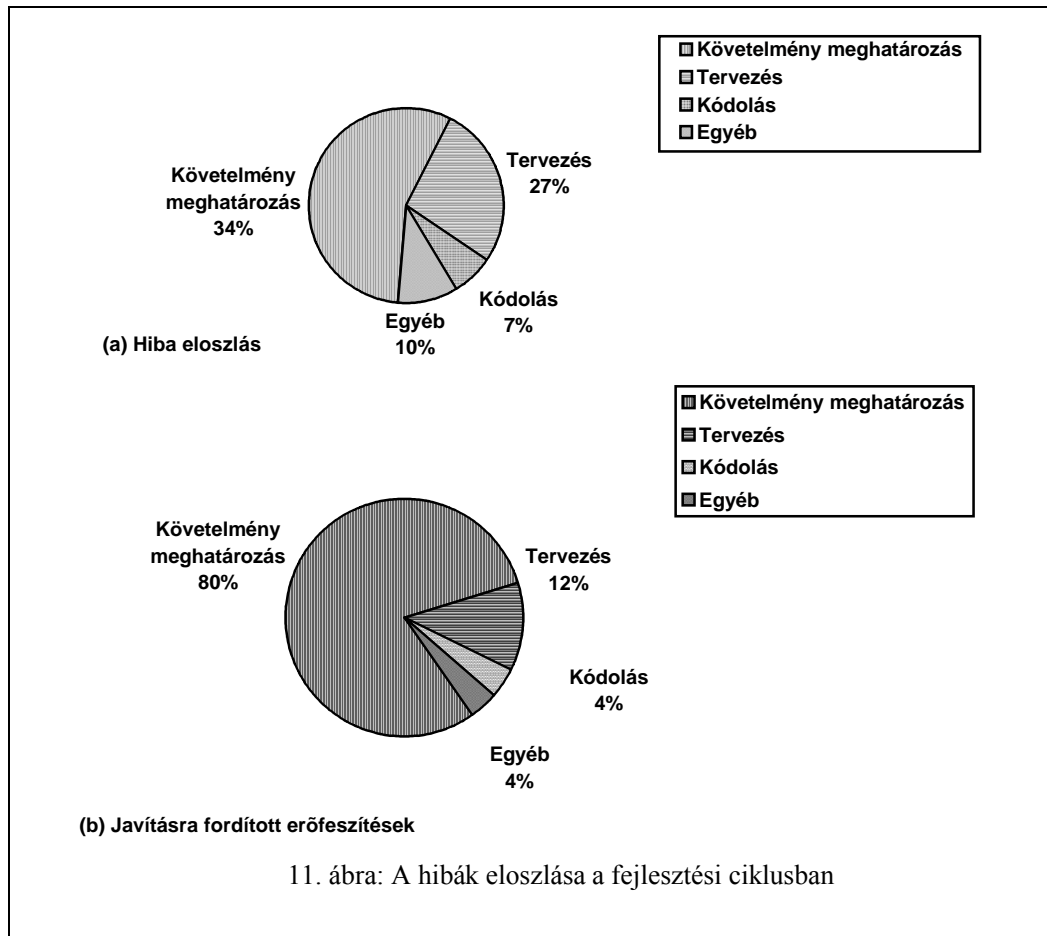
egy információrendszeré

30 év .

(forrás: Guidelines for an informatics architecture - Commission of the European Communities).

Új követelmények az alkalmazásokkal szemben. Az információrendszerek fejlesztőinek újabb és újabb alkalmazási területekkel kell szembenéznük, mint például az irodaautomatizálás, döntéstámogató rendszerek és szakértő rendszerek. Ezenkívül a döntési folyamatok jó előkészítése, megalapozása felkeltette az igényt az integráltság iránt, ami pedig az információrendszerek adatainak megosztására, konkurens elérésére és ellentmondásmentességük fenntartására világított rá.

Felhasználók elégedetlensége. Az információrendszerek fejlesztésével kapcsolatos problémák leginkább a felhasználókkal való együttműködés során jelentkeznek, a végső eredménnyel azonban a felhasználók sokkal többször elégedetlenek mint elégedettek. Az olyan jelenségek mint pl. a késedelmes leszállítás, költségtúllépés, rugalmatlanság és a nem megbízható rendszerek, amelyek rendszeresen visszatérő felhasználói panaszoknak számítanak.



A hagyományos információrendszer-készítés hátrányait a következőkben összegezhethetnénk:

gyenge követelményspecifikáció, ami jelentős mértékben a szöveges leírásokban elkerülhetetlenül előforduló kétértelműségek következménye,

gyenge rendszertervek, ami annak a következménye, hogy nem voltak szabályok, heurisztikák, ökölszabályok a követelmény specifikáció átalakítására jó rendszertervvé,

a rendszer nehezen volt karbantartható,

exponenciálisan egyre növekvően többet és többet kellett a rendszerek karbantartására fordítani a gyenge tervezés következtében, emiatt az új alkalmazásokra egyre kevesebb erőforrás jutott, ami az új felhasználói követelmények növekvő hátralékában jutott kifejezésre, azaz nagyon sok felhasználói kívánságnak, kérésnek semmilyen esélye sem volt arra, hogy valaha is megvalósul.

a fejlesztést nehéz volt formális minőségbiztosítási eljárásrendbe illeszteni.

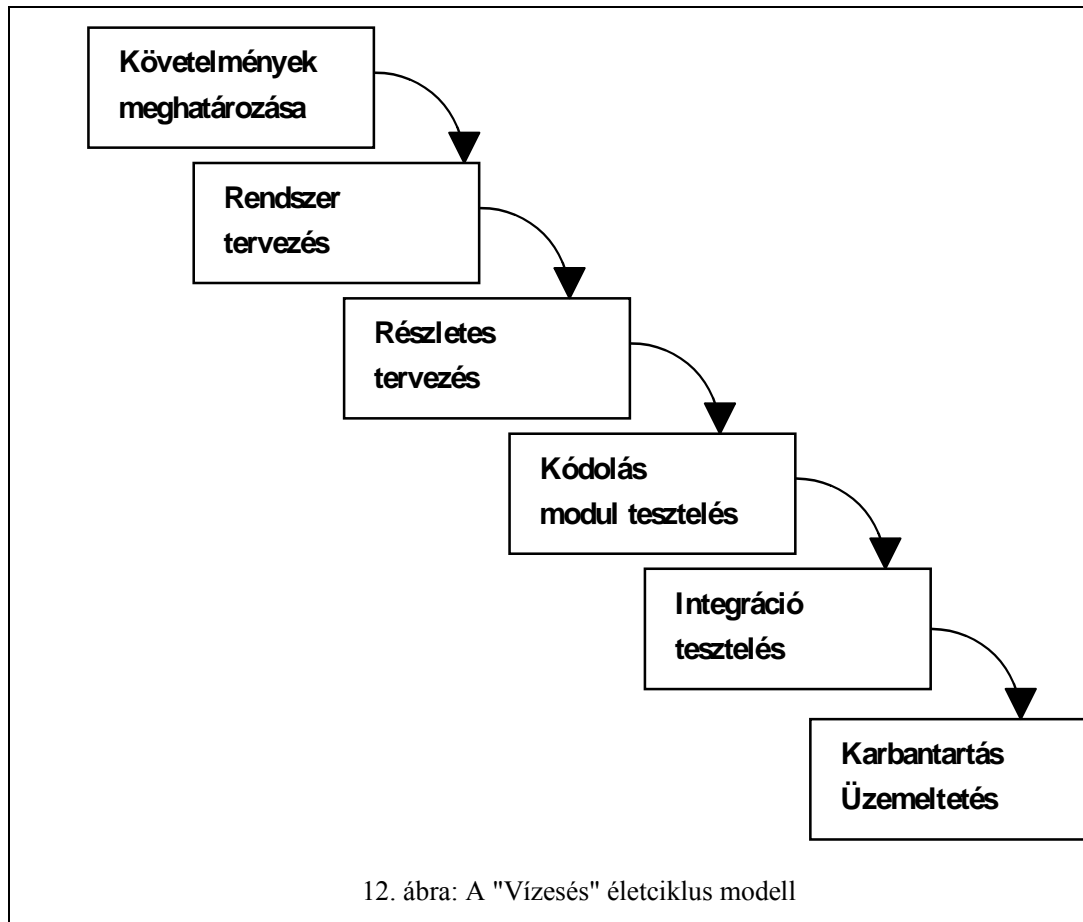
Egyes felmérések azt mutatták a 80-as évek elején, hogy a számítástechnikában alkalmazottak 92%-a foglalkozik a rosszul tervezett rendszerek tesztelésével és hibakereséssel.

A fentebbi okok együttesen vezettek oda, hogy az információrendszerek 'informális', hagyományos fejlesztése alkalmatlan a jelenlegi információrendszerekkel szemben fennálló igények kielégítésére és ezért szükség van tudományosan megalapozott rendszerfejlesztési módszertanok kialakítására.

1.7 A fejlesztés szakaszai

Életciklus modellek

Nagyon röviden ismertetjük az elfogadott rendszeréletciklus modelleket, mert ezekre a fogalmakra szükségünk van, de ezek részletes megvitatása más tárgyhöz és jegyzethez tartozik.



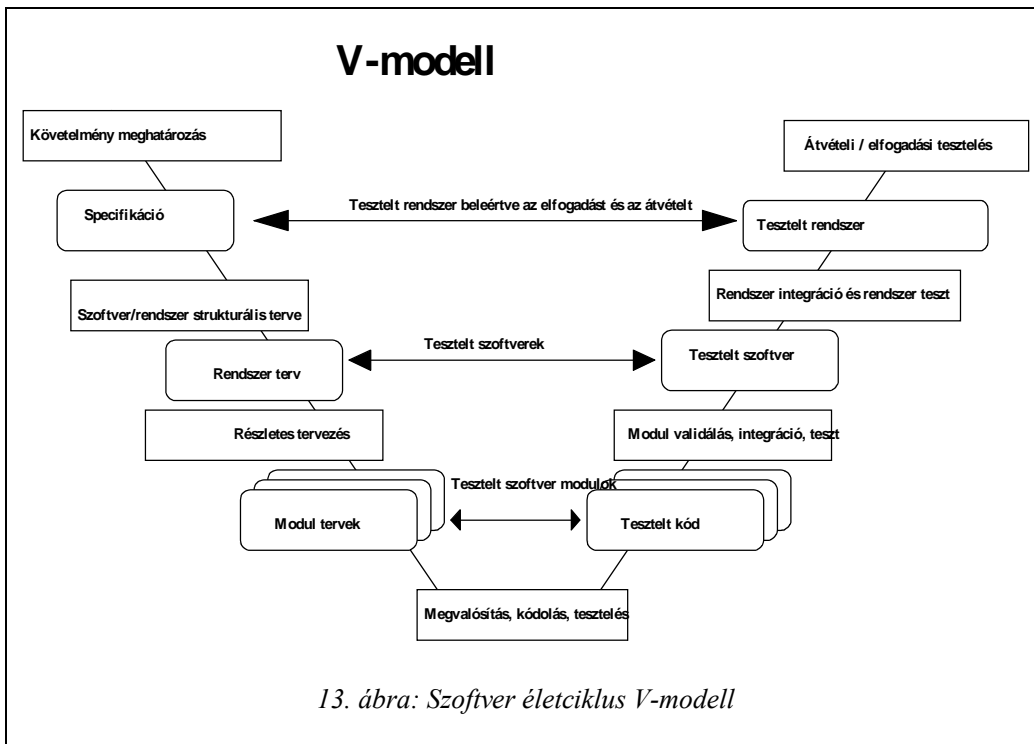
Az alkalmazás követelményeinek meghatározása

- Rendszer tervezés
- A részletes tervezés
- Kódolás és hibakeresés, modul tesztelés
- Integrációs tesztelés
- Működtetés és karbantartás
 - A modell alkalmazása:
 - Hagyományos - bármely szakaszból bármely másikba vissza lehet lépni
- Szoftver mérnöki - visszalépés csak a ciklus legelejére lehetséges
- Prototípus - A követelmény specifikáció és a részletes tervezés között ciklikus visszalépés
- CASE - A fejlesztés minden lépésébe bekerül
 - A V-modellnek van néhány előnye a vízesés modellel szemben (ld.20. ábra):
 - az egyik szakaszban létrehozott termékek közvetlenül átkerülnek a rákövetkező szakaszba;

- azokat a termékeket, amiket le kell ellenőrizni, tesztelni a különböző szakaszokban, eltérő szintű hibaellenőrzési eljárásokkal kapcsolja össze, világosan és expliciten kijelölve ezeket a szinteket.

Az ábrán a lekerekített sarkú téglalapok utalnak az elkészítendő termékekre, a téglalapok a szakaszokra. Az idő előrehaladását úgy lehet képzelni, hogy fentről lefelé haladva a legszélső baloldali elemtől a legszélső jobboldali elemig haladunk.

A bonyolult rendszerek, szoftverek kifejlesztésére alternatív életciklusként dolgozta ki Boehm a spirál életciklust (lsd. 19.ábra). Ezt a spirált legcélszerűbb egy polár koordináta rendszerben felfogni, ekkor a radiális irány mutatja a fejlesztésben addig felmerült, akkumulált költségeket, a sugár által bezárt szög illetve az ezáltal meghatározott szektor pedig a fejlesztés előrehaladását reprezentálja. A modell szerint a spirál minden egyes ciklusa a fejlesztés előrehaladását jelenti, lényegében ugyanazokat a fejlesztési lépéseket ismételve, de

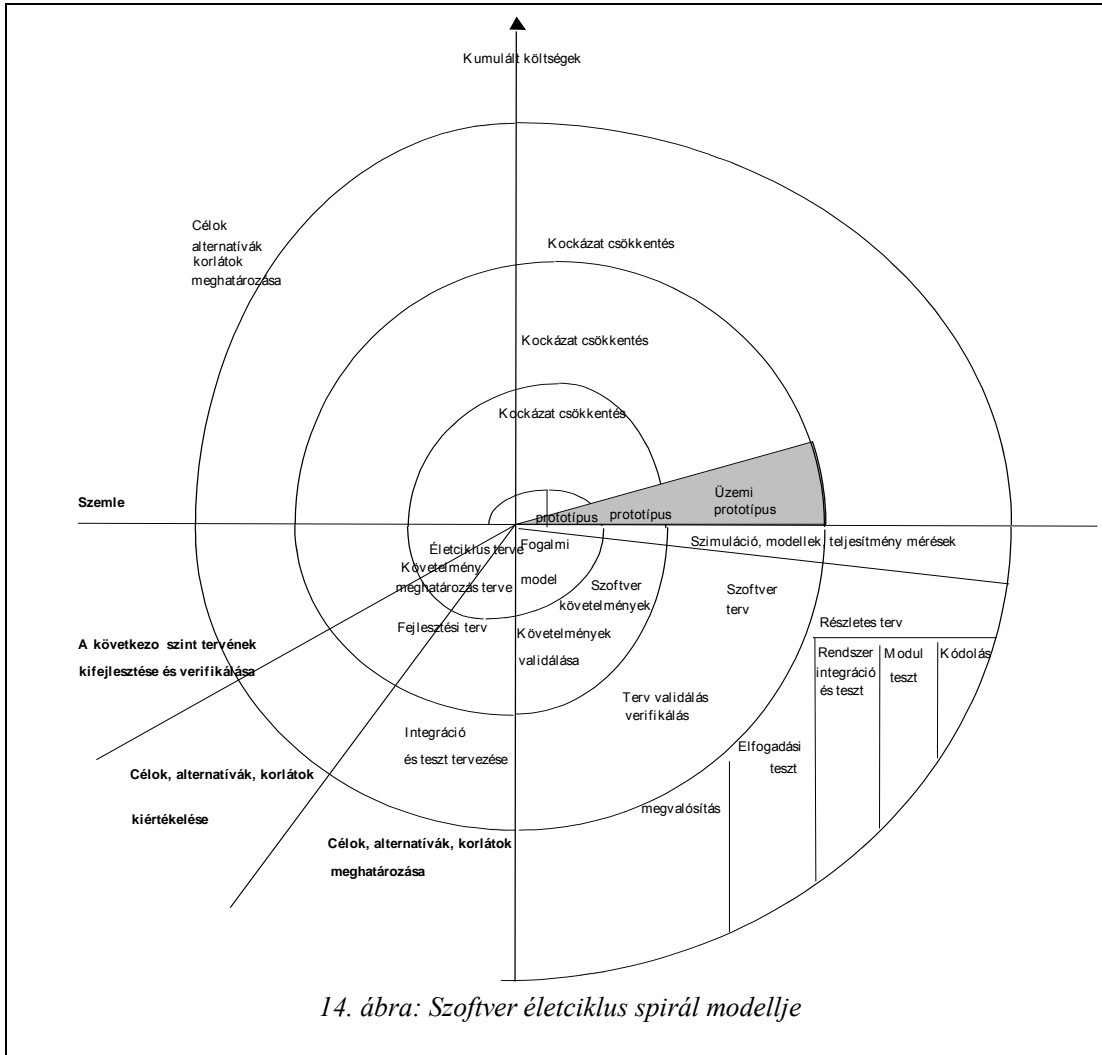


természetesen minden egyes körben az előzőleg létrehozott termékek finomítását, finomabb kidolgozását hozva létre a termékek minden részére és kidolgozottsági szintjére vonatkozóan. A leendő rendszer működését globálisan leíró dokumentumból kiindulva egészen az egyes programok lekódolásáig.

Minden ciklus a spirál balszélső legfelső negyedében kezdődik a következők meghatározásával:

a végtermék(ek) mely részét és milyen célból kell kidolgozni (teljesítmény, funkcionalitás, adaptivitás azaz a változásokhoz történő rugalmas alkalmazkodás képessége, stb.);

- a végtermék(ek) ezen részének megvalósíthatóságának elemzése, alternatív módok elemzése (alternatív tervek, újra-felhasználhatóság, beszerzés, stb.);



- az alternatívák korlátai és hatása az alkalmazásra (költségek, ütemtervek, kapcsolatok és felületek más alkalmazásokhoz, stb.).

A következő lépés az alternatívák kiértékelése tekintettel a célokra és a korlátokra. A bizonytalanságra, instabilitásra utaló jeleket fel kell ismerni, mivel ezek a projekt kockázatát nagy mértékben fokozzák, ezért olyan költségtakarékos megoldásokat kell találni, amelyek a felbukkanó kockázatok kezelésére alkalmasak, mint például a prototípus készítés, szimuláció, felhasználók intenzív bevonása, elemző / analitikus modellezés, vagy különböző kombinációi ezeknek és esetleg egyéb más kockázat csökkentős technikáknak.

Miután a kockázatokat kiértékeltek, a felismert kockázatok viszonylagos egyensúlya fogja a következő lépést meghatározni, pl.:

- evolúciós fejlesztés (prototípus), a végtermék globális tulajdonságait minimális befektetéssel specifikálják, megtervezik a prototípus fejlesztés következő lépését, majd egy sokkal részletesebb prototípust fejlesztenek ki, mellyel a fontosabb kockázatokat mérséklük;
- inkrementális fejlesztés, azaz a klasszikus életciklus fejlesztés a következő szakaszra, vagyis az előző szakasz eredményeire támaszkodva annak bővítését hozzák létre (fogalmi modell, követelmény meghatározás, strukturált tervezés, stb.);

A specifikáció készítés minden lépését egy *validációs* és *verifikációs* lépés követi, majd a következő (spirál)ciklus terveinek az előkészítése. Minden dokumentum kifejlesztésének utolsó szakaszát a jobboldali legalsó negyed mutatja.

Minden ciklus végén egy átfogó *szemlélet* hajtanak végre, ez jellemző tulajdonsága a spirál modellnek, az ábrán pedig a baloldali tengely érzékelteti ezt a tevékenységet. Ez a szemle az előző ciklusban előállított összes terméket vizsgálja, beleértve a következő ciklus ütem - és erőforrás-tervét. A szemlézésben az összes érintett és érdekelt fél, felhasználó, szervezet részt vesz. A másik fontos tulajdonsága ennek a modellnek, hogy nagy hangsúlyt helyez az esetleg megjelenő kockázatok összes aspektusának kezelésére és alkalmas döntési mechanizmus kialakítására.

Ezt a modellt különösen szeretik használni:

- Gyors fejlesztéseknél (Rapid Application Development, DSDM, Dynamic System Development Method);
- Prototípus alapú, evolúciós megközelítéseknel;
- Szakértő és ismeretbázisú rendszereknél;
- specifikáció orientált, automatikus transzformációt alkalmazó, szimulációt használó megközelítéseknel
- Inkrementális fejlesztéseknél.

Információrendszer adaptációk készítésének szakaszai

Az előző fejezetek néhány szoftver fejlesztési életciklus modellt ismertettek röviden. Ezekből is látható volt, hogy számtalan szakaszolási módja van a fejlesztésnek. Az információrendszerek fejlesztése a konkrét szoftver fejlesztésnél még nagyobb területet fog át, így itt is módszertanonként különböző szakaszolással találkozhatunk:

- információrendszerek stratégiai tervezése,
- rendszerelemzés,
- rendszertervezés,
- rendszerkészítés.

Az Euromethod (ld. [CCTA95B], [Turner96], [Euromethod94]) ezt összefoglaló névvel információrendszer adaptációnak nevezi (IR-adaptáció).

1.7.1.1 Információrendszerek stratégiai tervezése

A szervezet tevékenységét működését elemzi, de nem olyan részletességgel, mint amikor egy konkrét rendszert akarunk megtervezni, amely egy adott tevékenység egészét vagy annak egy részét segítené. Ekkor a már létező rendszerek elemzésére is szükség van, abban az értelemben, hogy milyen hasznot hajtanak, nyújtanak-e valamilyen előnyt a szervezetnek. Az

információrendszerek stratégiai tervének meg kell jelölnie azokat a rendszereket, amelyeket létre kellene hozni és azt a sorrendet, amelyben a kivitelezésük megtörténhetne. A rákövetkező szakaszok a szervezet illetve a működés, a tevékenységek egyre szűkebb körére koncentrálnak. Ennek a szakasznak a végterméke, a megrendelőnek *átadandó termék*, a leendő információrendszerek terveinek egy portfóliója, ezt tulajdonképpen *tervezési terméknek* tekinthetjük.

1.7.1.2 Rendszerelemzés

Ez a szakasz a szervezet egy meghatározott működési területének helyzetét vizsgálja meg és egy helyzetfelmérési tanulmányt készítene. A létező információrendszereket tanulmányozzák, akár manuális akár automatizált rendszerről is legyen szó. Elemzik, hogy valójában mit is csinálnak a szervezetben, és tulajdonképpen mit kellene csinálni ahhoz, hogy egy sokkal fejlettebb információrendszer működését támogassák.

Ebben a szakaszban megint keletkezik egy a megrendelőnek *átadandó termék*, amit ugyanakkor a *rendszerelemzés termékének* tekinthetünk. Ez a szakasz tulajdonképpen leíró és nem előíró jellegű.

1.7.1.3 Rendszertervezés

A rendszertervezési szakasz egy a leendő információrendszer vonatkozó előírást állít elő, általában elektronikus formában. Az alkalmazási terület kiterjedése sokkal szűkítettebb, mint a megelőző rendszerelemzési szakaszban vizsgált területé.

Gyakran a tervezési szakasz eredményeként megjelenő *tervezési termék* független a rendszer létrehozása során használandó eszközöktől. Azonban sokszor már ekkor lehet tudni, hogy mik lesznek a készítés során használt eszközök és ezeknek a tulajdonságait figyelembe lehet venni, különösen teljesítmény tervezési szempontból.

1.7.1.4 Rendszerkészítés (létrehozás)

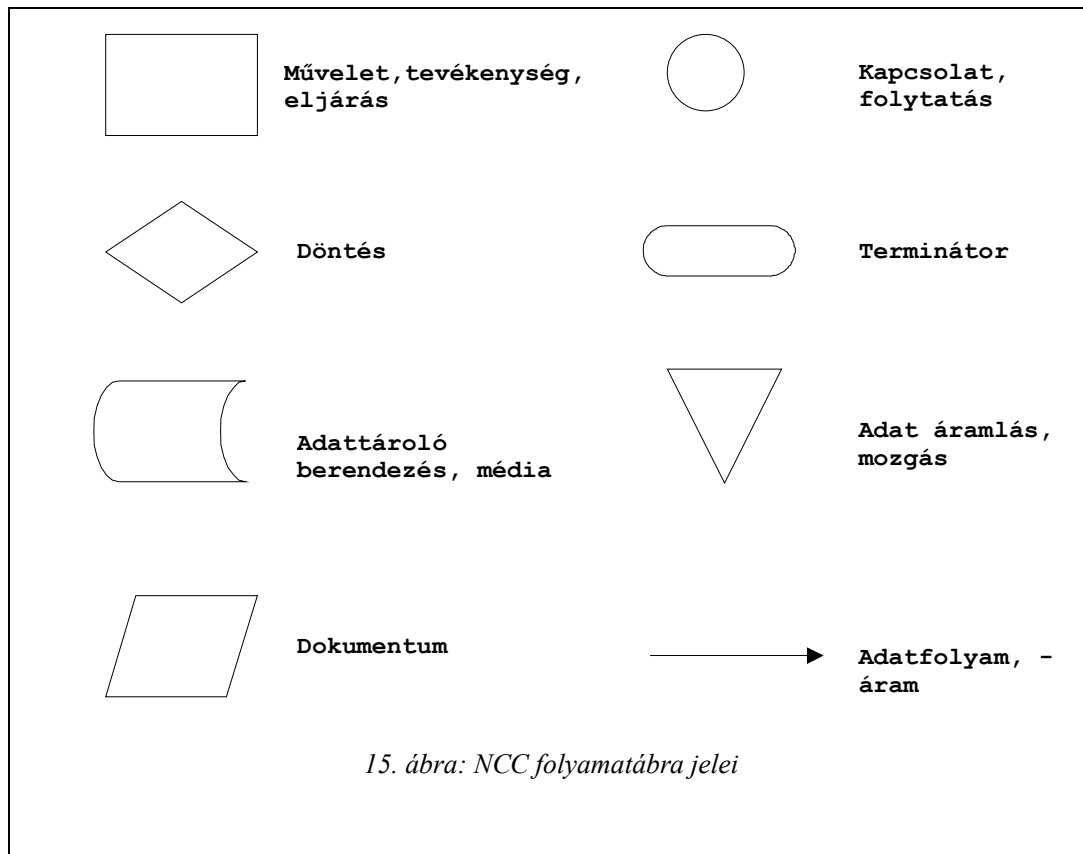
A rendszerkészítési tevékenység tulajdonképpen nagymértékben függ a rendelkezésre álló hardver és szoftver környezettől. A rendszerfejlesztési környezet általában a következő eszközöket tartalmazhatja:

- adatbázis-kezelő rendszer;
- adatszótárak, repozitóriumok;
- képernyő tervező eszközök, grafikus felhasználói felület tervező eszközök;
- tranzakció feldolgozó eszközök;
- programozási nyelvek;
- alkalmazás generátorok.

A szoftver fejlesztési környezet kiválasztása ideális esetben a rendszertervezési szakasz befejezése után történik meg, azaz miután a rendszert minden részletére kiterjedően már megtervezték.

Folyamatok elemzése

Az előző fejezetben az előzetes adatgyűjtés és a probléma meghatározás főbb lépéseivel és technikáival ismerkedtünk. A gyakorlatban ezek a lépések gyakran egymással párhuzamosan, és folyamatosan történnek, egészen a tervezési szakaszig tartanak.



Az adatgyűjtésen túllépő elemzéshez szükségünk van egy sokkal szisztematikusabb keretre, amelyben feldogozzuk a megszerzett ismereteket. A következő fejezetben a folyamatok elemzésével foglalkozunk. Az idevágó fogalmak és technikák ismertetésekor az egyik jelöléstechnikát választjuk ki, de ez a jelöléstechnika csak a külső megjelenése ugyanazoknak az elveknek és szemantikai tartalomnak, amit esetleg egy másik módszer kissé különböző jelöléssel jelenít meg.

1.8 Bevezetés a folyamatmodellezésbe

A folyamat modellezés főbb céljai:

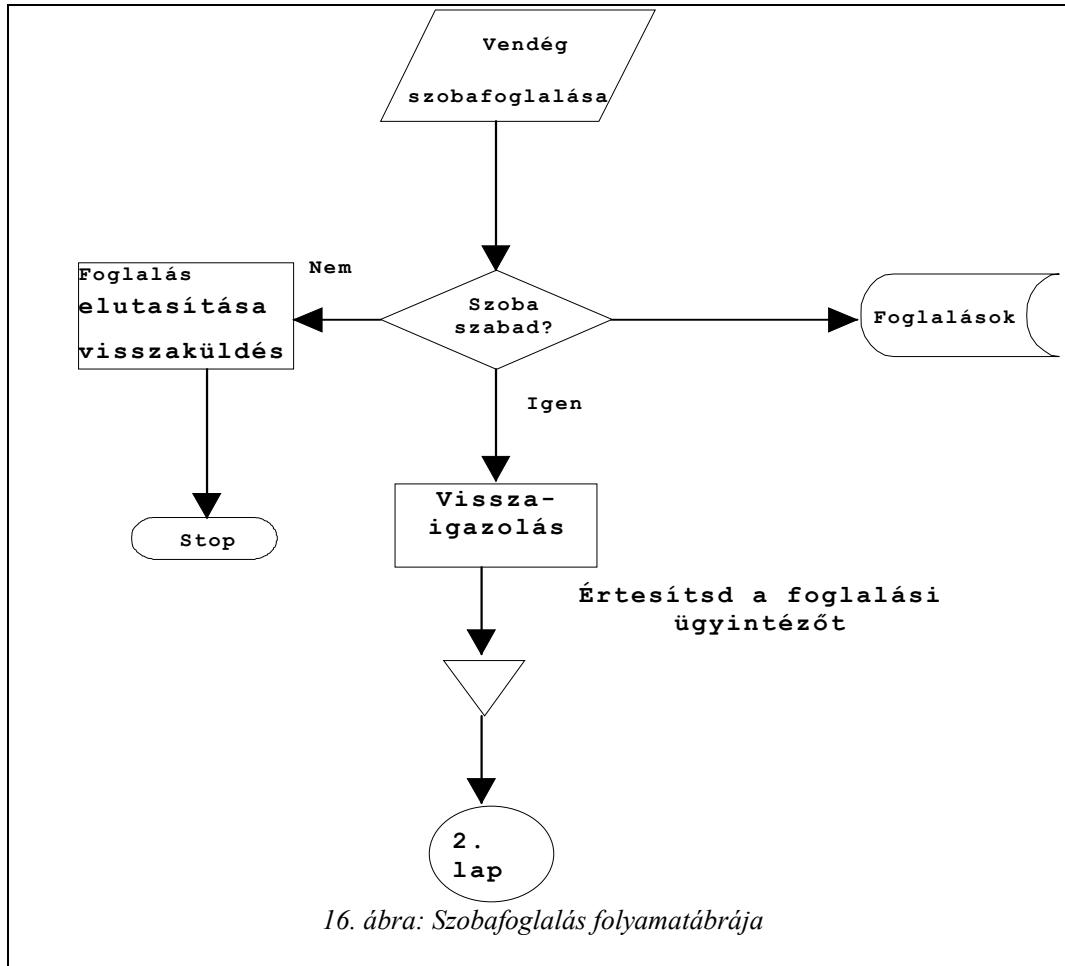
kielégítő pontossággal írja le a jelenleg működő rendszert;

- ábrázolja azt az absztrakt rendszert, amely már nem tartalmaz fizikai korlátokat;
- visszatükrözi a leendő, igényelt rendszerrel szemben támasztott követelményeket, az elképzelt rendszer működését.

Ezeket a rendszer ábrázolásokat az interjúkból, kérdőívekből, a mérési adatokból és a dokumentumok feldolgozásából absztrakció útján hozzák létre.

A legfontosabb analízis technikák, amelyeket érinteni fogunk:

- adatfolyam diagram (Data Flow Diagram, DFD), amely a folyamatok és a közöttük levő

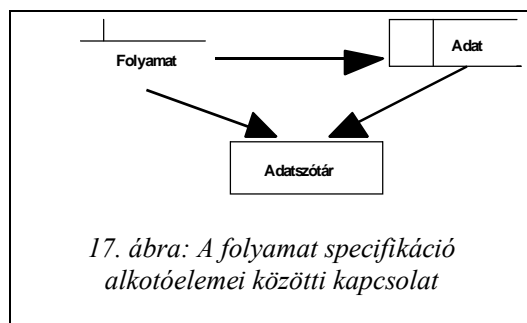


16. ábra: Szobafoglalás folyamatábrája

kapcsolatokat az adatfolyamok értelmében ábrázolja;

- adatszótár, amely az adatfolyamokat alkotó adatelemek leírását tartalmazza;
- folyamatok specifikációja, amely a folyamatok írja le részletesen.

A fentebb felsorolt modellezési elemek közötti kapcsolatot a következő ábra érzékelteti (17.ábra). Az információrendszerek középpontjában az *adatok* vannak, ennek megfelelően a folyama specifikációban centrumában is áll. Ezeknek az adatoknak általában jól definiált szerkezete van; az adatok leírását az *adatszótár* tartalmazza, nevezetesen: az adatelemek tulajdonságait, jellemzőit valamint megjegyzéseket, az analízis során feltárt és tapasztalt dolgokról. A rendszeren belül tárolt adatokat a *folyamatok* módosítgatják, változtatják az információrendszer működési céljának megfelelően.



17. ábra: A folyamat specifikáció alkotóelemei közötti kapcsolat

Bevezetés az adatfolyam modellezésbe

Definíció 1-1 Adatfolyam modell

Az adatfolyam modellezési technika az információrendszerek folyamatainak kapcsolatát írja le a közöttük áramló adatfolyamok értelmében, feltüntetve a rendszerben tárolt adatokat is.

A fentebb megfogalmazott modellezési célkitűzéseknek sok jelölés megfelel, több alternatív diagram technika terjedt el, a CASE eszközök is különbözőket támogatnak. az alábbiakban bemutatunk néhányat közülük, majd arra fogunk koncentrálni, amely a további tanulmányokat előkészíti és segíteni fogja (27.ábra).

Az ábrán látható jelölések közül az SSADM módszertannak megfelelőt fogjuk használni, aminek több oka van:

Magyarországon elfogadott kormányzati ajánlás az államigazgatásban.

Több Magyarországon forgalmazott CASE eszköz támogatja a módszertant és az általa alkalmazott jelöléstechnikát.

Nagy-Britanniában szabvány és az államigazgatásban előírt módszertan, valamint más országokban is használt módszertan.

Az informatikával illetve információgazdálkodással foglalkozó hallgatók részletesebben fogják tanulni az SSADM módszertant.

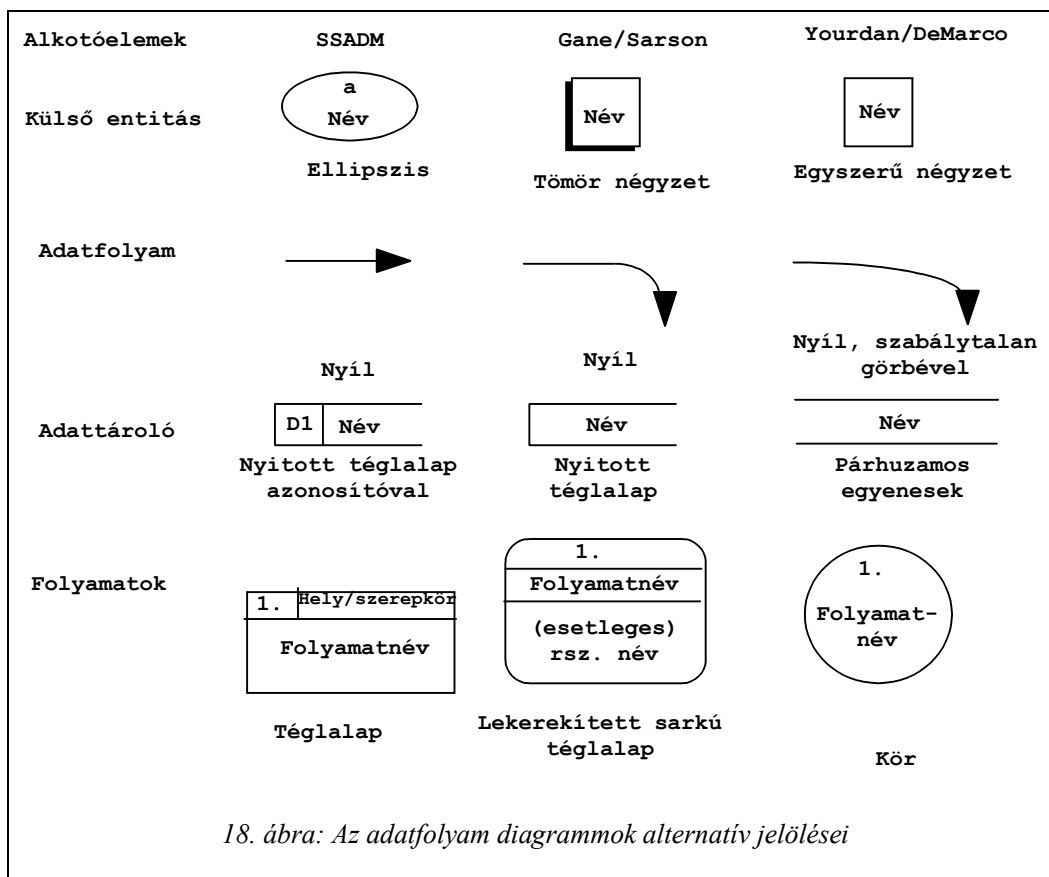
Ezért ezt a jelöléstechnikát használva ismertetjük az adatfolyam modellezést, de továbbra is utalni fogunk esetleges alternatív módszerekre és jelölésekre. Azonban a háttérben meghúzódó elméletek és elvek ugyanazok, így ez az alkalmazhatóságot és az általánosságot egyáltalán nem zavarja.

Ahogy az az ábráról is kiderül, az adatfolyam diagram legfontosabb alkotórészei a következők:

- külső entitás,
- adatfolyam,
- adattároló,
- folyamat.

1.8.1.1 Külső entitás

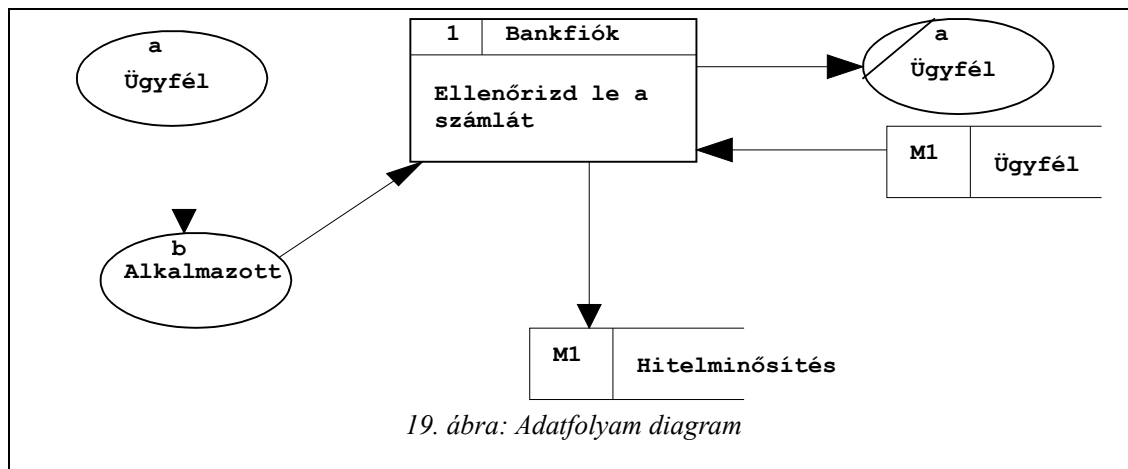
Külső entitásnak vagy (információ)forrásnak és nyelőnek nevezik azokat a személyeket, osztályokat, szervezeti egységeket, szervezeteket, más rendszereket, amelyek akár információt szolgáltatnak az elemzésnek alávetett rendszernek vagy fogadnak onnan jövő adatokat. Egy külső entitás egyszerre lehet információforrás és - nyelő. Például az ügyfél, aki



megrendelést ad fel és megkapja a kiszállított árut egyszerre információforrás és - nyelő. A külső entitás definíció szerint az elemzett rendszer határán kívül van, voltaképpen implicit módon a külső entitások felsorolása meghatározza a rendszer határát. Ha a rendszer határát egy falnak képzeljük, akkor az adatok a falon lévő lyukon keresztül léphetnek be a rendszerbe vagy hagyhatják el a rendszert, így a külső entitást a "lyuk" mögött levő valaminek képzelhetjük. Ez a kép azt is sugallja, hogy a rendszerhez "legközelebb" levő valamit kell külső entitásnak tekinteni, azt ami közvetlenül kapcsolatban van a rendszerrel és adatfolyamot küld be vagy fogad a rendszertől. Például egy bankfiókban az ügyfél valamilyen tranzakciót kezdeményez a folyószámláján a bank alkalmazottjánál, aki egy terminálon vagy számítógépen valamit végrehajt. Ebben az esetben a bank alkalmazottat kell külső entitásnak tekintenünk, aki esetleg egy másik külső entitástól az ügyféltől kap valamilyen adatot.

A pókháló szerű ábrák elkerülése végett, a külső entitás több példányban is megjelenhet a diagrammon, ezt egy "sapkával" jelöljük; továbbá az entitást az ABC kis betűivel azonosítjuk, valamint egy nevet adunk neki. Gyakran ez a név az elemzett terület leendő felhasználóinak feladataiból, munkaköréből származhat. De szembe kell nézni azzal a ténnyel, hogy több különböző személy, eltérő helyszíneken esetleg ugyanazokat az adatokat

adja át a rendszernek, ugyanazokat a felhasználói funkciókat hajtja végre, ez a rendszerelemzőt szükségtelen absztrakciókhoz vezetheti, olyan elnevezéseket használhat, amiket a felhasználók nem értenek meg. Ez arra mutat rá, hogy ez csak segít a felhasználói szerepkörök feltérképezésében, de nem helyettesíti a megfelelő technikát.



19. ábra: Adatfolyam diagram

1.8.1.2 Adatfolyamok

Ezt a nyilat - eltérően a folyamatábrákon használt nyilaktól, amelyek a vezérlésátadást reprezentálták - úgy foghatjuk fel mint egy utat, amelyen egy vagy több adatszerkezet közlekedhet föl s alá, az időpont meghatározása nélkül. Az idővel illetve ütemezéssel kapcsolatos kérdéseket a folyamatok specifikációjában részletezik. Az adatfolyam diagrammot ebben az értelemben egy vasúti térképhez hasonlíthatjuk, amelyen a vonatok által követhető utakat láthatjuk, de a menetrendet nem mellékeltek. Általában az adatfolyam egy olyan nevet kap, ami egyértelműen azonosítja azt az adatszerkezetet, amely az adatfolyamon keresztül áramlik - és értelmes a felhasználók számára.

Az adatfolyamok lehetnek egy vagy két irányúak. Különösen a két irányú adatfolyamoknál kell figyelni arra, hogy ne tartalmazzanak vezérlési információkat vagy olvasási kérelmet. A programozási logika szerint sokan kísértést éreznek arra, hogy amikor olvasnak egy adattárolóból, akkor azt az ábrán egy az adattárolóba vezető nyillal érzékeltessék, az adatfolyam adatszerkezete pedig tartalmazza a rekord kulcsát. Ez hibás! Csak az adattárolóból kivezető nyilat kell bemutatni a diagrammon ebben az esetben, mivel ez érdekes a vizsgálat és a szervezet működése szempontjából, ez jelenti az olvasás vagy adat visszakérését.

	Külső entitás	Folyamat	Adattároló
Külső entitás	Külső adatfolyam	Igen	Nem
Folyamat	Igen	Igen	Igen
Adattároló	Nem	Igen	Nem

20. ábra: Az adatfolyam ábra elemei között megengedett (adatfolyam) kapcsolatok

Alkalmanként a külső entítások között is ábrázolhatjuk az adatfolyamokat, ekkor szaggatott vonalat használunk. Annak ellenére, hogy ezek a szűkebb értelemben vett rendszeren kívül vannak sokszor segíthetik a megértést. A táblázatban az adatfolyam diagram elemei között a megengedett adatfolyamokat tüntettük fel.

A „Jelenlegi Fizikai Adatfolyam Diagrammon” az adatfolyamok a valóságban áramló információkat ábrázolják, vagyis a tényleges űrlapokat, formanyomtatványokat, dokumentumokat, iratokat, telefon hívásokat, stb.

A „Logikai és az Igényelt Rendszer Adatfolyam Diagramján” ezek az adatfolyamok azokat az adatelemeket, attribútumokat jelentik, amelyeket a folyamatok használnak bemenetként és kimenetként.

Egy folyamatban se nem keletkezhet, se nem tűnhet el adat (SSADM); dokumentumok keletkezhetnek vagy elnyelődhetnek, de kell lennie minden folyamatnál az összes bemenő adatra olyan kimenetnek, amely közvetlenül kapcsolódik hozzájuk. Ez igaz a teljes diagramra, egy folyamatra, a folyamat esetleges lebontására. Ennek az oka az, hogy a folyamatok csak az adatelemeken okozhatnak változásokat, és ezáltal a rendszer állapotában (lsd. 0). Ezt matematikailag a következőképpen fogalmazhatjuk:

Vannak olyan módszerek, ahol megengedik, hogy adatelem 'keletkezzék', és ha ezt kifejezetten specifikálják, akkor vannak CASE eszközök is, amelyek még konzisztencia ellenőrzést is támogatják. Adatelem, jelentések készítésénél, lekérdezéseknél keletkezhetnek; úgy nevezett leszármaztatott elemek jöhetnek létre, más elemek összege, átlaga, stb.

1.8.1.3 Folyamatok

Egy folyamat az adatokat módosítja, manipulálja a rendszeren belül. Általában a szervezet valamely tevékenységét jelenti, amelynek lefolyását valamilyen adat megérkezése kezdeményezi, az adatok aktualizálása után valamilyen kimenetet készít és küld tovább. Nem szabad összekeverni a számítógép programokkal! Előfordulhat, hogy bizonyos esetekben egy az egyben megfelel egy folyamat egy programnak, de ekkor is a felhasználók számára érthető kifejezéseket kell használni a leírására és nem informatikai szakkifejezéseket. Vagyis ezen a ponton szervezet működését kell visszatükrözni. Ez lehet valamilyen számítás elvégzése, új dokumentum készítése azokból az adatokból, amelyek elindították, vagy a bejövő adatokat módosítja.

Az adatfolyam ábra tehát olyan folyamatokat mutat, amelyek az adatok átalakításával foglalkoznak és nem azokat, amelyek csak a jelentésekhez szükséges formátumokat alakítják ki. Ez alól a szabály alól csak akkor lehet kivételt tenni a Jelenlegi Fizikai Adatfolyam ábrán, amikor a felhasználó szemében a jelentés készítés a rendszer nagyon lényeges részének számít.

A folyamatot egy téglalappal jelenítjük meg. A hivatkozási szám a kisebb téglalapban csupán egy azonosító, nem fejez ki sem sorrendet, sem fontosságot. A hosszabb sávszerű téglalapban arra a helyre, helyszínre lehet hivatkozni a Fizikai Adatfolyam ábrán, ahol a folyamat végbemegy. Ahogy az adatfolyam ábra egyre pontosabbá válik úgy lehet egyre pontosabbá tenni a helyszínre való hivatkozást is, ez egyébként lehet annak az alkalmazottnak a megnevezése is, aki végrehajtja ezt a tevékenységet. A folyamatnak egy tömör, világos nevet kell adni. Az egyik lehetőség az, hogy egy igei kifejezést, - felszólító módban - és főnévi kifejezést alkalmazunk. Az ige a tevékenységet írja le, a főnév pedig azt a tárgyat, amelyet a folyamat az igével kifejezett cselekvéssel manipulál. Ez a megoldás magyarul sokaknak furcsának tűnik, ezért úgy módosítható, hogy az igei kifejezés - *ás*, - *és* képzővel főnevesített változatát használhatjuk. Elemzési szempontból azonban nagyon fontos, hogy a folyamat *aktív (ige)* és *passzív (főnév)* része egyértelműen megjelenjen. (19. ábra).

1.8.1.4 Adattároló

Az adattároló az a hely, ahol az adatok nyugalomba jutnak, ez a hely az - ahogy azt a neve is mutatja -, ahol egy ideig a rendszeren belül az adatokat tároljuk. Ez lehet egy iratszekrény, iktatókönyv, indexelt kartonok, dossziék, elintézendő ügyiratok tálcája, főkönyv, vagy egy

számítógépes adatállomány. Az SSADM jelölés egy egyik rövidebb oldalán nyílt téglalapot használ, a baloldalán egy négyzettel, ami tartalmaz egy betűt és egy számot azonosítóként:

D: számítógépes adatállomány;

- **M:** manuális adattároló, pl. iratszekrény, napló, stb.;
- **T(M):** manuális adattároló. Ez egy ideiglenes adattároló, ami azt jelenti, hogy addig tartják az adatot (adatrekordot) itt, amíg *egyetlen egyszer* ki nem olvassák, azaz az olvasás *destruktív*, az olvasással együtt a tárolóból eltávolítódik vagy törlődik az adatrekord, pl. az elintézendő ügyiratok tálcája, vagy egy postaláda.
- **T:** számítógépes adattároló. Az előbbi adattároló elektronikus változata., pl. egy csak az adatok sorba rendezése miatt ideiglenesen létrehozott adatállomány.

Az adattároló nevének a benne tárolt adatok tartalmát kell kifejeznie, nem pedig a tárolás módját, tehát pl. 'Iratszekrény' helyett 'Személyzeti nyilvántartás'-t kell használni.

Az adattárolóba kerülő és tárolt adatoknak (adatrekordoknak) egyértelműen, egyedileg azonosíthatóknak kell lenniük. Maguk az adatrekordok kezdetben (Jelenlegi Fizikai DFD) lehet, hogy nem optimális szerkezetűek (nincsenek 3. normál forma alakban, 3. NF), de egy úgy nevezett kulcsuknak lenniük kell.

1.8.1.5 Az adatfolyam diagram (DFD) készítésének lépései

Az alap fogalmak és jelölések megismerése után a diagram készítésének módszerével ismerkedünk meg részletesebben. Az eddigiek alapján az adatfolyam diagram (DFD) legfontosabb céljait a következőkben foglalhatjuk össze:

hogyan lépnek az információk be a rendszerbe és hogyan hagyják el;

- mi változtatja meg az információkat;
- miben és hol tárolják az információkat.

Az adatfolyam diagrammot a rendszerelemzésben mint eszközt a következő célokra használjuk:

A rendszer határának megállapítása. A diagram világosan megmutatja a rendszer határait és kiterjedését; a külső entitások és a rendszer határát átlépő adatfolyamok mutatják ezt meg explicit módon.

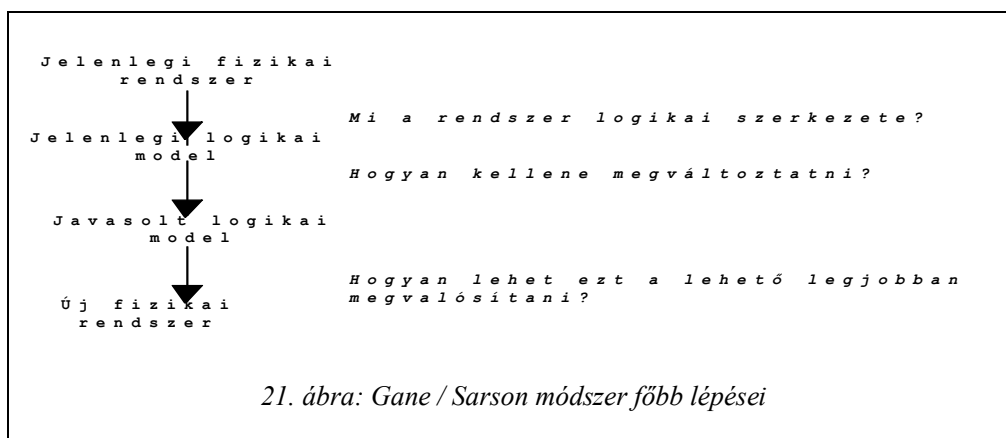
- *Az elemzés teljességének leellenőrzése.* A diagram készítésének módja, illetve összevetése más modellekkel, amelyek más nézőpontból készültek, biztosítják, hogy az összes információ-folyamot, információtároló helyet, és a rendszer összes tevékenységét feltárták.
- *További specifikációk alapja.* A módszertantól függően szolgálhat a funkciók specifikációjának, vagy logikai rendszertervnek, esetleg programtervnek az alapjául.

SSADM-ben a következő adatfolyam diagramok jöhetnek elő a fejlesztés során:

Jelenlegi Fizikai. A jelenleg működő rendszert modellezi a pillanatnyi állapotában, megvalósításában.

- *Logikai.* Tisztán logikai ábrázolása a jelenlegi rendszernek, amelyet a *Jelenlegi Fizikai* adatfolyam diagramból a fölösleges, fizikai korlátok megszüntetésével vezettek le.
- *Rendszerszervezési alternatívák.* Több alternatív rendszerszervezési tervet készítenek, amelyek mindegyike kielégíti a legfontosabb rendszerrel szemben támasztott követelményeket. Ezeket az alternatívákat áttekintő szintű adatfolyam diagrammokon ábrázoljuk.
- *Igényelt.* A kiválasztott rendszerszervezési alternatívákra támaszkodva, a módszertanban előírt mélységű adatfolyam diagram halmazt készítenek el.

Nagyjából megegyeznek a különböző módszertanok által javasolt lépések, nemcsak az adatfolyam diagramra, hanem az adatmodellekre és a feldolgozás logikájára vonatkozóan. A strukturált módszertanok egyik alap gondolata az, hogy van már működő rendszer azt értjük, meg sajátítsuk el a felhasználók szótárát, kifejezés készletét, ezzel ábrázoljuk a *Jelenlegi Rendszert* majd ezen megértési fázis után kezdjük csak el vizsgálni azt, hogy hogyan illeszthetők be ebbe a rendszerbe illetve ennek a logikai leképezésébe a követelmények. Gane / Sarson módszer arra az álláspontra helyezkedik, hogy egy projekt vezetési döntés az,



amikor meghatározzák azt, vajon melyik út a gyorsabb. Azaz a javasolt logikai rendszert a jelenlegi rendszer ábrázolásából vezessék le, vagy sokkal termelékenyebb, ha a javasolt rendszert a semmiből fejlesztik ki, "Felejtük el azt, hogyan csináljuk most!" felkiáltással.

Határozzuk meg a rendszer határát

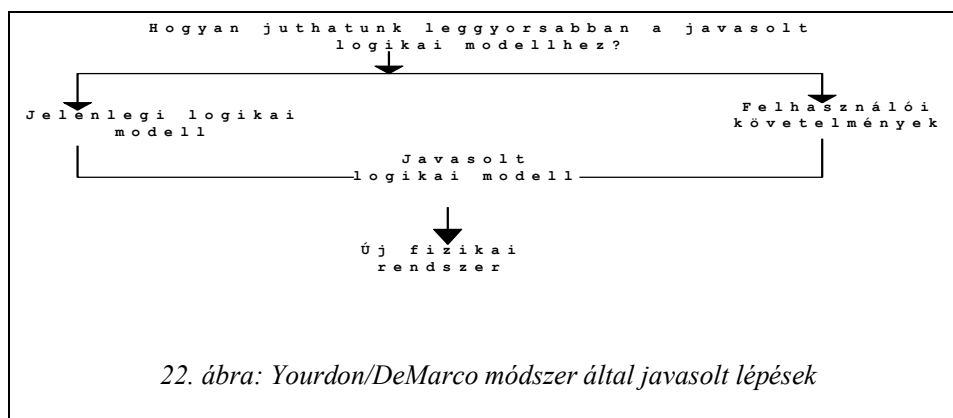
A felhasználóval egyetértésben állapítsuk meg annak a területnek a határait, amelyet vizsgálni fogunk. Ez lesz az előzetesen megállapított rendszer határ, amit a rendszer határán kívül fekvő *külső entitások*, esetleg rendszerek, folyamatok, adattárolók képében tudunk megfogalmazni.

Azonosítsuk az adatfolyamokat

Keressük meg a rendszer határát átlépő adatokat, információkat, bármilyen formában tegyék is azt (0). Adjunk nekik olyan nevet, ami kifejezi az adatszerkezet tartalmát, és nemcsak annak egy részére vonatkozik. Kerüljük a semmit mondó elnevezéseket (lista, formalap, információ, adat, stb.).

Azonosítsuk a külső entitásokat

Elemezzük a bemenő és kimenő adatokat és állapítsuk meg, hogy honnan származnak. Ezek az információforrások és -nyelők lesznek a definíciónak megfelelően a külső entitások. Különös figyelmet érdemes szentelni annak, amikor a szervezetben csak egy bizonyos papír



mozog valamilyen szervezeti cél érdekében.

Azonosítsuk a folyamatokat

Minden adatfolyamra hozzuk létre a rendszeren belül a létrehozó vagy fogadó folyamatot; a folyamatokat egészítsük a hozzájuk tartozó adattárolókkal.

További folyamatok felismerése

Az eddigi alapján az adatfolyam diagram első változatát már fel lehet vázolni. Keressük azokat a folyamatokat, amelyek csak kizárólag a rendszeren belül működnek (a külső entitások számára ezek nem láthatóak). Ide tartozhatnak az adatok karbantartása a szervezet céljaival összhangban, esetleg adat konvertálások.

További adatfolyamok felismerése

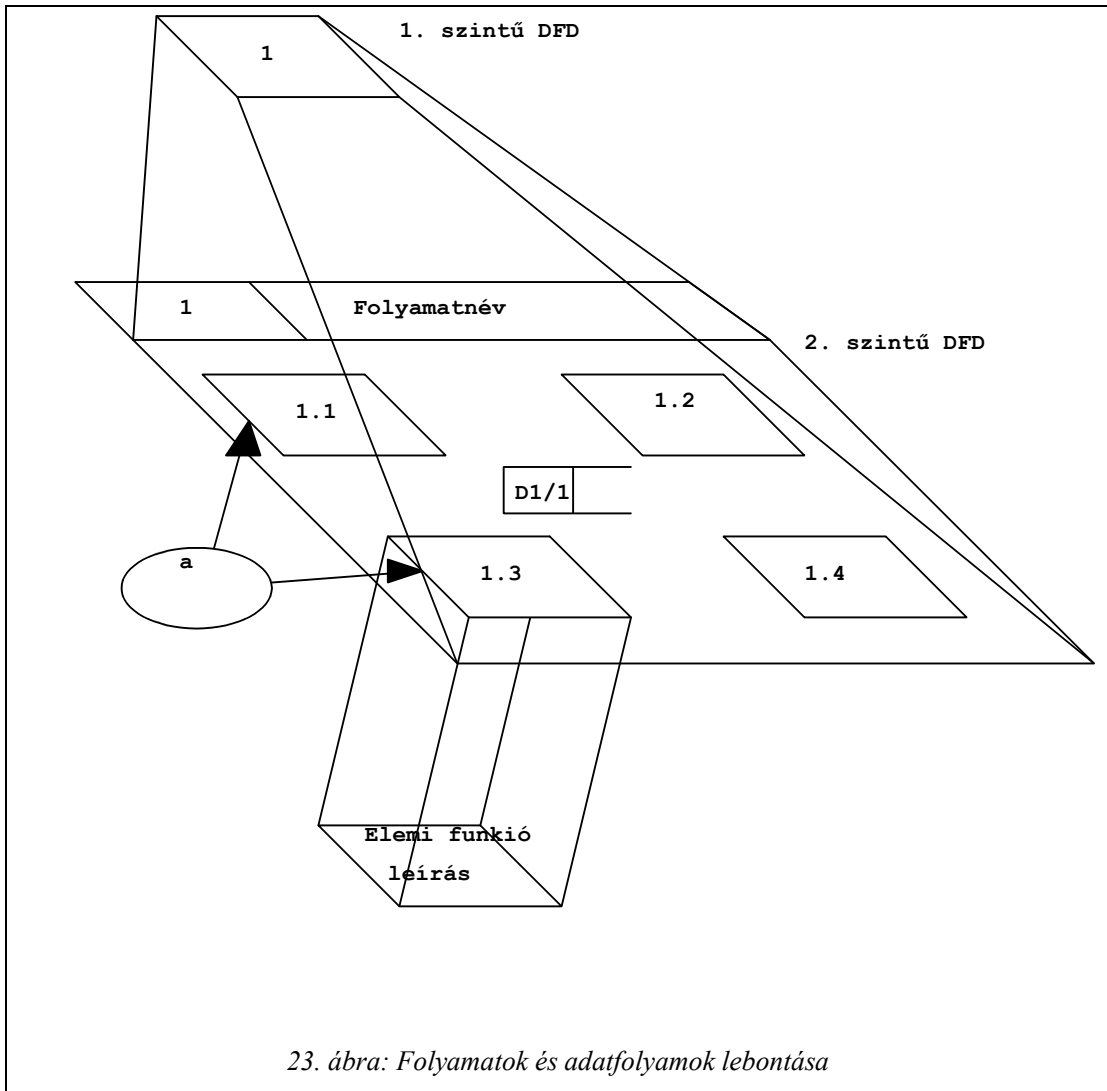
Az előbbi lépés (0) miatt fellépő újabb adatfolyamok meghatározása következik.

Konzisztencia és teljesség ellenőrzés

Miután elkészült a diagram hajtsunk végre egy informális ellenőrzést, lehetőleg a felhasználóval együtt:

Minden folyamat megnevezése egy igei és egy főnévi kifejezésből álljon. Ha ez nem így van, akkor esetleg további szétbontásra lehet szükség.

- A folyamatok nem adat források vagy nyelők, hanem *átalakítók*, (*adat-transzformátorok*), azaz egy folyamat a bemenő adatot változtatja át egy kimenő adatfolyammá, amely egy másik folyamathoz, adattárolóhoz vagy külső entitáshoz vezet. Egy folyamat nem generálhat adatot a semmiből és ezt nem is ábrázolhatja a diagram.
- Egy adattárolóhoz tartoznia kell mind belépő mind kimenő adatfolyamoknak. Ha csak belépő van, akkor az adattároló 'felrobban', vagyis végtelenségig csak töltődnek adatokkal, vagy másképp fogalmazva csak az a folyamat(ok) használná(k) adattárolásra, amelyből a bemenő adatfolyamok érkeznek, és ekkor szükséges a diagrammon megjeleníteni, mert ez azt jelenti, hogy a megfelelő folyamat 'hasában' kell ennek az adattárolónak feküdnie.
- A diagram készülttségének ezen a fokán lehetséges, hogy minden dokumentum, adat teljes életét és felhasználását végigkövessük a rendszerbe történő belépésétől kezdve addig, amíg el nem hagyja azt. Ellenőrizzük nem hagytunk-e ki valamit, nincsenek-e szabálytalanságok, eltérések.



23. ábra: Folyamatok és adatfolyamok lebontása

- Az adatfolyam diagramnak pontosan tükröznie a jelenlegi rendszert.
- A felhasználónak elégedettnek kell lennie a rendszer kiterjedésével, megállapított hatáiraival.
- A rendszer funkcionális felbontása - az adott terület funkciói szerinti szétbontás - egyszerre praktikus és hasznos a munkafolyamatok átszervezése és felülről lefelé haladó felbontás előkészítése végett.

1.8.1.6 Hierarchikus adatfolyam diagram (DFD) készítése

Az adatfolyam diagram erőssége a kifejezőerejében áll, azaz abban, hogy voltaképpen mindegyik ábra felér vagy ezer szóval. Azonban csak a legegyszerűbb rendszerek azok, amelyek egy A4-es lapon ábrázolhatók (pl. statisztikai, csak lekérdezésre szolgáló rendszerek), egy bonyolultabb rendszer részletes leírása nem fér el egy lapon. Ennek a

nehézségnek a legyőzésére, több adatfolyam diagrammot kell használni, amelyeket hierarchikusan rendezünk el, a hierarchia legfelső elemét első szintű adatfolyam diagramnak (1 szintű DFD) hívják. Az ezen ábrázolt dolgok határozzák meg az alacsonyabb szintű diagrammok alapvető jellegzetességét; a felső szintű egy átfogó, magas szintű, nem nagyon részletes képet nyújt a rendszerről, az alacsonyabb szintűek a rendszer egy korlátozott részéről adnak egy sokkal részletesebb ábrázolást.

Az irodalomban a szakértők vitát folytatnak arról, hogy mennyire van szükség az adatfolyam diagram jellegű ábrázolásra, mivel lehetne helyettesíteni a szervezeti működési modellezés leírására. Van néhány javaslat ([Vecsenyi88], [Checkland81], [Checkland90]) de a gyakorlatban úgy tűnik, hogy nehéz helyettesíteni, mert bár nem tökéletes technika, mégis egy jó kompromisszum az informatikai igények, a felhasználók számára a megérthetőség, olvashatóság szempontjából, valamint a fokozatos finomítás, felülről-lefelé haladó, elemzés, a szükséges részletezettség szempontjából.

Az adatfolyam diagram lebontásának szabályai

A legfontosabb rendszer funkciókat a az 1. szintű adatfolyam diagram folyamatai ábrázolják, mindegyik ilyen folyamatot szükség szerint ki lehet terjeszteni egy alacsonyabb szintű folyamattá. Úgy gondolhatunk minden egyes folyamatot ábrázoló téglalapra mint egy ablakra, amelyen keresztül az alacsonyabb szintű diagramra tekinthetünk ki. Az alacsonyabb szintű diagram tartalmazza a magasabb szintűn megjelenő elemek kiterjesztését (23. ábra), nevezetesen:

folyamatok,

- adattároló,
- adatfolyamok (az ábrán az a-1 adatfolyam az a-1.1 és a-1.3 adatfolyamra bomlik le),
- alsó szintű külső entitások.

A lebontás legfontosabb szabályai a következők:

A magasabb szintű folyamatot ábrázoló téglalap határa lesz az alacsonyabb szintű adatfolyam diagram határa. Ezt a határt a világosan meg kell mutatni, az új diagrammon pedig meg kell jeleníteni annak a folyamatnak a számát, amiből származtattuk, valamint a nevét.

- Az alsóbb szintű folyamatok azonosítójának egyértelműen mutatnia kell az eredetét, erre a "tizedes pontos" jelölést alkalmazzuk, tehát az alsóbb szintű folyamatok származtatott azonosítója 1.1, 1.2, stb. lesz.
- Ha egy adattárolót csak az adott folyamat határain belül használunk, akkor azt a folyamat határain belül használjuk és D1/1, D1/2, stb. azonosítóval látjuk el. Azokat az adattárolókat, amelyeket más folyamatok is használnak, az adatfolyam diagram határán kívül maradnak.
- Külső entitások az alacsonyabb szintű adatfolyam diagrammok határain is kívül maradnak, még akkor is adatot cserélnek ezzel a folyamattal.
- **Szintek közötti egyensúly** fenntartása. Egy adott folyamatból kimenő és bejövő adatfolyamoknak az alacsonyabb szintű diagrammon is meg kell jelennie. De megfordítva is, ha az alacsonyabb szinten fedezünk fel újabb adatfolyamot és az nem származtatható egy magasabb szintű adatfolyamból, annak a lebontásából, akkor azt vissza kell vezetnünk a magasabb szintre. Vagyis a bemenő adatelemek halmazelméleti uniójának meg kell egyeznie a kimenő adatelemek halmazelméleti uniójával.

Ezek a szabályok azt hangsúlyozzák, hogy egy folyamat részletes lebontása azt jelenti, hogy a folyamatot úgy vizsgáljuk meg részleteiben, hogy ne veszítsünk információt a

folyamatot körülvevő környezetből; a többi folyamattal, adattárolóval, külső entitással fennálló kapcsolatokból.

Az SSADM 2 illetve 3 szintű diagram elkészítését javasolja, Yourdon módszer nem szab ilyen határokat. Ennek a határnak is főleg az az értelme, hogy ne töltsünk feleslegesen sok időt egy esetleg rosszul működő rendszer pillanatnyi helyzetének elemzésével és ennek a leírásával.

Azokat a folyamatokat, amelyeket nem lehet tovább bontani, illetve a legalacsonyabb szintű diagrammon jelennek meg elemi folyamatoknak vagy funkcióknak hívjuk. Vannak elemzők, akik akkor tekintenek egy folyamatot eleminek, ha csak egy bemenő és csak egy kimenő adatfolyama van. Másik megközelítés szerint, akkor 'jó' egy elemi folyamat, ha a funkcionális '*kohéziója*' nagy, vagyis kifejezhető a tevékenysége egy igével és egy tárggyal, az elemi tevékenységei közötti összetartó erő, összefüggés nagyon erős.

A lebontás olyan részletezettségű diagramhoz vezethet, amelyen előfordulhatnak olyan, folyamatok, amelyeket nem kötnek össze más folyamatokkal adatfolyamok, vagy nagyon sok adatfolyam lépi át egyes folyamatok határát. Ez arra figyelmeztet, hogy ezeket a folyamatokat, illetve a magasabb szintű szülőiket felül kell vizsgálni.

Módszertanok

A rendszerelemzés legfontosabb technikáinak, valamint néhány tervezési eljárás rövid ismertetése után a *legelterjedtebb* módszertanokról adunk egy összefoglaló képet, tekintettel az osztályozási szempontjainkra (1.3). A világon több száz különböző módszertant használnak, egy felmérés 300-ra teszi a számukat. De valószínűleg még ennél is nagyobb a többé-kevésbé hasonló módszertanok száma, amelyeket valahol leírtak, publikáltak, vagy egyes cégek egy bizonyos változatot kötelezővé tettek belső használatra. Vannak olyan vélemények is, hogy a különbség az egyes módszertanok között valójában triviális, a hangsúlyozott különbségek csak a piaci versenyben játszanak szerepet.

1.9 Strukturált módszertanok

A strukturált módszertanok fejlődése több mint két évtizedre tekint vissza. Ennek köszönhetően a legjobban kidolgozott, érett módszertanok, alkalmas eljárás és technika halmazzal tartoznak ide. Ezek a módszertanok azért jöttek létre, hogy az információrendszer-tervezés minőségi problémáit kezeljék illetve enyhítsék (1.6). Ezt abban próbálták megfogalmazni, hogy a leendő információrendszer "használhatósága" jó legyen; ami persze ebben formában meglehetősen homályosan hangzik.

Jobb végeredmék

A következőkben felsorolunk egy kritérium rendszert, amely egy kicsit közelebről körül járja az információrendszer "használhatóságának" fogalmát:

Elfogadhatóság: azok számára, akik a rendszert használják, vajon a rendszer megfelelő-e, kielégíti-e az általuk támasztott követelményeket.

- *Rendelkezésre állás:* vajon a rendszer rendelkezésre áll-e ott és akkor, amikor arra szükségre van.
- *Kohézió:* a rendszer alkotórészei (a részrendszerek) közötti információ csere vajon olyan-e, hogy a végeredmény egy összehangolt működésű, egységes (integrált) információrendszer.
- *Kompatibilitás:* vajon bármelyik adott részrendszer zavartalanul illeszkedik-e be a az egész integrált rendszerbe.
- *Dokumentáció:* vajon létezik-e olyan jó minőségű dokumentáció, amely segíti a felhasználókat, a fejlesztőket, a vezetőket és az üzemeltetőket közötti kommunikációt.
- *Megtanulható:* új felhasználók esetében a betanulási görbe vajon elég rövid-e.
- *Gazdaságosság:* vajon maga az információrendszer eredményez-e költségtakarékosságot, és a tervezett erőforrás felhasználáson és egyéb korlátokon belül marad-e.
- *Hatékony:* vajon a rendszer hatékonyan használja fel és ki a rendelkezésre álló erőforrásokat.
- *A fejlesztés gyorsasága:* vajon az információrendszer készítésére fordított idő viszonylag rövid volt-e a rendszer méretéhez és bonyolultságához képest.
- *Rugalmasság:* vajon könnyű-e a rendszert módosítani, új alkotórészt beépíteni vagy egy régit eltávolítani.
- *Funkcionalitás:* vajon a rendszer gondoskodik-e a rendszer (funkcionális) követelményeinek megvalósításáról.

- *Alkalmazásba-vehetőség*: vajon a régi rendszerről az új rendszerre az áttérés megvalósítható-e.
- *Gyenge csatolás*: a részrendszerek közötti kommunikáció, információcsere vajon olyan-e, hogy a részrendszerek úgy módosíthatók, hogy annak nincs hatása a rendszer többi részére.
- *Karbantarthatóság*: vajon mekkora erőfeszítésre van ahhoz szükség, hogy a rendszer üzemeltetését folyamatosan fenntartsák.
- *Hordozhatóság*: vajon az információrendszer tud-e más környezetben, hardver konfiguráción, platformon, más osztályon futni.
- *Megbízhatóság*: vajon a hibák előfordulásának gyakorisága alacsony-e, a kimenetek helyesek-e, és ellentmondásmentesek-e.
- *Robosztusság*: vajon a rendszer hibatűrő-e és hiba esetén visszaállítható-e valamilyen alapállapotból, vagy esetleg automatikusan saját maga is megteszi ezt.
- *Biztonság*: vajon az információrendszer elég robusztus-e ahhoz, hogy szándékos vagy véletlen helytelen használat nem vezessen a rendszer teljes összeomlásához
- *Egyszerűség*: vajon a kétértelműségeket, redundanciákat, bonyolult megoldásokat minimalizálták-e.
- *Teszteltség*: vajon a rendszert alaposan tesztelték-e azért, hogy elkerüljék az üzemelési hibákat és a felhasználók elégedetlenségét.
- *Reszponzivitás*: vajon a rendszer normálisan működik-e a csúcs, normális és a minden egyéb feltételű terhelések időszakában, és akkor és ott szolgáltatja az információt, amikor és ahol az szükséges.
- *Használhatóság*: a használat egyszerűsége, könnyűsége, funkcionalitása, stb., amely megfelel a különböző típusú felhasználók igényeinek (alkalmi, rendszeres, tapasztalt).
- *Átláthatóság*: vajon a felhasználó számára nyomon követhető-e, hogy bizonyos tevékenységek miért bukkantak fel, és miért úgy hajtották végre.

Természetesen optimális egyensúly a fentebb felsorolt kritériumok között nem hozható létre, hiszen vannak olyanok, amelyek csak egymás rovására elégíthetők csak ki. A módszertan illetve a projekt maga hangolható úgy, hogy azok a szempontok kapjanak hangsúlyt, amelyek különösen fontosak az adott probléma területen.

SSADM

Az **SSADM** (Structured Systems Analysis and Design Method) tulajdonosa a **CCTA** (Central Computer and Telecommunications Agency), amely Nagy-Britannia pénzügyminisztériumához tartozik, és a kormányzati információs rendszerek beszerzése és készítése felett lát el felügyeletet, valamint az információs rendszerek és az informatika területén a kormányzati politikát alakítja ki. A továbbfejlesztését a Nemzetközi SSADM Felhasználók Csoportja (International SSADM User's Group, ISUG) illetve egy arra illetékes testülete felügyeli. A Brit Számítógéptudományi Társaságon belül (British Computer Society, BCS) létezik egy olyan testület, amely a szakmai előírások megvalósulását, azok teljesítésének ellenőrzését egy vizsgáztatási rendszer kialakításával

Az **SSADM** tulajdonképpen eljárási, műszaki és dokumentációs szabványok gyűjteménye, amelyet úgy terveztek meg, hogy kifejezetten a rendszerelemzést és a szoftverfejlesztést támogassa. Két főrészből áll, az egyik a felhasználói követelmények elemzése, a másik a rendszer tervezése. Ezeket a részeket szakaszokra és lépésekre tagolja. A szakaszok összessége lefedi az adatmodellezés technikáit, a követelményelemzést és a

szoftver tervezést. Az **SSADM** egyik legfontosabb tulajdonsága, hogy rugalmas, azaz az adott (fejlesztési) körülményekhez igazítható, továbbá az egyik leghatékonyabb módszer, amely olyan szervezetek rendelkezésére áll, amelyeknek egy szabványos rendszerfejlesztési filozófiára és megközelítésre van szükségük.

Az **SSADM** nyílt rendszer. Ez azt jelenti, hogy nyilvános, bárki számára hozzáférhető, bárki használhatja licenc díj fizetése nélkül, engedélyt sem kell kérni a **CCTA**-tól.

Ez a nyíltrendszer-stratégia egybeesik más egyéb kormányzati, nyílt szabványnál követett eljárással Nagy-Britanniában (pl. OSI, POSIX). Kifejezetten úgy tervezték, hogy a megjelenése a piacot újra szabályozza és a versenyt a termékek és a szolgáltatások (pl. konzultáció) között fokozza, valamint felszabadítsa a piacot azokról a korlátokról, amelyet a tulajdonosnak fizetendő licencdíjak jelentenek. Az **SSADM** stratégia egyik legfontosabb célja, hogy biztosítsa a szolgáltatási piac hatékony működését, a felhasználói igényeket a piaci lehetőségek maximumáig kielégítse. Ily módon a fejlesztésért felelős vezető nem kerül kiszolgáltatott, függő helyzetbe a konzultációt, oktatást és a megvalósítást végző személyektől, ha azokat egy idő után nem találja már a legalkalmasabbnak a feladat ellátására; ilyen esetben a szerződéses partnereket másikkal helyettesítheti anélkül, hogy a befektetések (pénz, idő, stb.) elvesznének.

Az **SSADM** 4.0 és 4.2 verziójában pontos útmutatások találhatók, hogy hol és hogyan alkalmazzák a minőségbiztosítási szabványokat ill. a kapcsolódó eljárásokat, nevezetesen az ISO 9001-t. Ezek az útmutatások nagymértékben rögzítik az ISO 9001 minőségellenőrzési eljárásai bevezetésének a módját azok számára, akik ezt alkalmazni kívánják.

A projekt vezetést/irányítást a **PRINCE** módszertan adja, ami jól összeillik az **SSADM**-mel.

Filozófia:

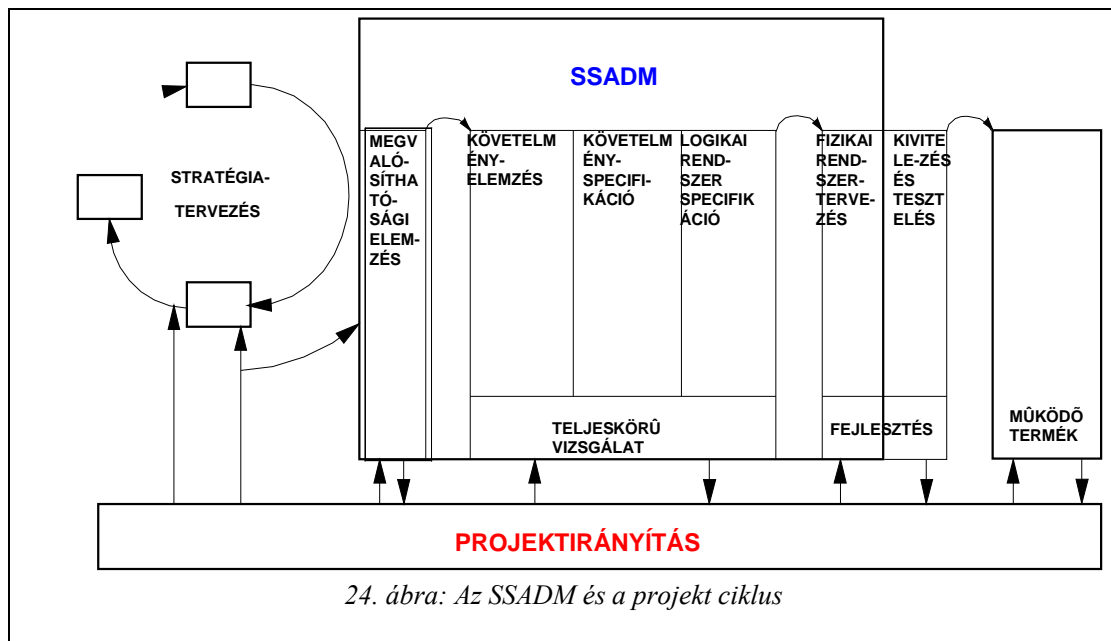
Az **SSADM** alapfilozófiája a különböző nézetek tudatos ütköztetésére, az adatvezérelt, folyamatközpontú és eseményirányított megközelítések tudatos kompromisszumának kialakítására törekszik. Alapvetően felülről-lefelé haladó és adatközpontú elemzési és tervezési módszer, valamint nagy hangsúlyt helyez a felhasználók bevonására. Stuktúrált módszertan és ezért a tudományos alapúak közé soroljuk.

Modellek:

Az entitás-kapcsolat, adatfolyam, entitás-élettörténet és esemény-hatás diagrammok (Jackson-szerű diagrammok) valamint a relációs technika azok legfontosabb eszközei, amelyekkel modellezi, leírja a rendszert.

Életciklus lefedése:

Az **SSADM** nem foglalkozik informatikai stratégiai tervezéssel - (de feltételezi a létét, pontosabban a rövid projekt specifikációk / meghatározások létét) - az opcionális Megvalósíthatósági Tanulmány készítésétől a Fizikai Rendszertervezéséig terjedően fedi le a rendszerelemzés és a - tervezés szakaszait. Vagyis csak részleges, nem teljes az életciklus lefedése. Az **SSADM** nem foglalkozik a rendszerkészítés, karbantartás, üzembe helyezés, és egyéb kiegészítő területek módszereivel ide értve a projektirányítást is.



Leszállítandó termékek:

Alapértelmezésben mindegyik szakasz végén egy pontosan meghatározott dokumentáció készletet kell átadni, amelyeket az adott szakaszban alkalmazott modellezési eljárások és technikák eredményeit tartalmazzák. Például az adatfolyam modell és a logikai adatszerkezet dokumentumait.

Előfeltételezések:

Az SSADM feltételezi, hogy a rendszerfejlesztés célja egy információrendszer létrehozása, azaz egy adatbázis-központú, tranzakció-orientált rendszer elkészítése. Feltételezi, hogy létezik a szervezet meghatározott, kezelhető méretű részére vonatkozó projekt alapító okirat, amire alapozva indulhat a munka. Továbbá a szervezetben van elfogadott projektirányítási módszertan és gyakorlat, valamint több területre kiterjedő szabályok, helyi szabványok és előírások (szervezeti és alkalmazási szintű felhasználói felület tervezési útmutatók, programozási, kódolási, biztonsági szabványok, stb.).

1.9.1.1 Döntési pontok az SSADM-ben

A hagyományos rendszerfejlesztési eljárásokban a végfelhasználók meglehetősen passzív szerepet játszottak, ők látták el a rendszerelemzőt információkkal és a specifikáció ellenőrzésében valamint a rendszer tesztelésében vettek részt. Azonban semmi esetre sem jöhetett az szóba, hogy befolyásolják vagy megpróbálják befolyásolni a rendszer tervét. Ilyen körülmények között a felhasználó hajlott arra, hogy elfogadja azt a tervet, amit megoldásként adtak neki anélkül, hogy a végfelhasználók kellő időben megkérdőjelezhették volna a terv alkalmasságát. Ennek az eljárásnak aztán számos súlyos következménye támadt.

Az **SSADM** ezzel szemben teljesen eltérő szerepet szán a végfelhasználóknak, ugyanis nekik kell mindazon kritikus döntéseket meghozni, melyek lényegesen befolyásolják a fejlesztés további menetét. Konkrétan három ilyen fontos döntési pont van:

- **A megvalósíthatósági tanulmány:** A rendszer terjedelme, határa, legfontosabb paraméterei, a rendszerfejlesztés stratégiája a végfelhasználók igényének megfelelően az ő egyetértésükkel kerül meghatározásra.
- **Rendszerszervezési alternatívák:** Lényegében azt határozzák meg, hogy a rendszernek tulajdonképpen MIT is kell csinálnia.
- **Műszaki megvalósítás alternatívái:** Ekkor a kiválasztott rendszerszervezési alternatíva lehetséges technikai/műszaki megoldásai közül választanak a végfelhasználók, ezek a megoldások többnyire széles skálán demonstrálják a szóba jövő műszaki opciókat. A kiválasztás megtörténte után világossá válik, hogy a rendszer HOGYAN fogja megvalósítani azt, amit szolgáltatnia kell.

1.9.1.2 SSADM egyéb tulajdonságai

A jelenlegi változatot SSADM 4.2-nek illetve SSADM4+-nak nevezik a következő értelemben, amelyik magában foglalja:

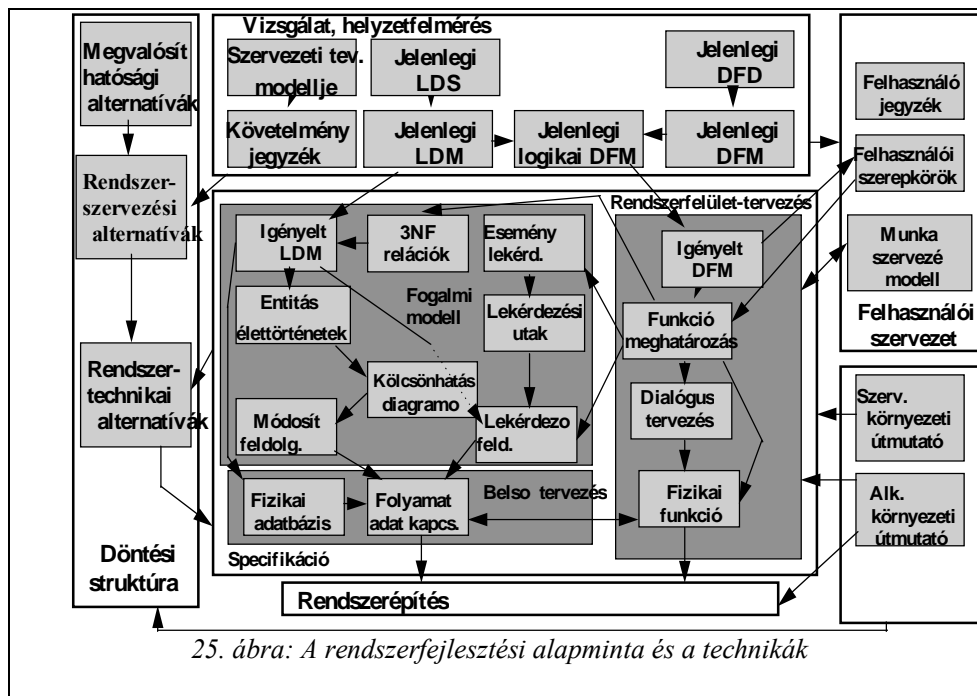
a módszertan magját, az SSADM 4.2-t;

- Az SSADM különböző körülményekre történő testreszabását segítő útmutatások;
- Egy Információrendszer Tervezési Könyvtár (Information System Engineering Library, ISE).

Az SSADM voltaképpen egy termék-orientált specifikációja egy minőségi információrendszer előállításához szükséges tevékenység sorozatnak, technológiának. A megvalósíthatóságra, az elemzésre, a rendszer / követelmény specifikációra és a tervezésre koncentrálnak. Az SSADM szakaszolása komoly segítséget nyújt a projekt tervezéshez és ellenőrzéshez, komoly hangsúlyt helyezve a termékek minőségére. A módszertanban részletesen leírt és előírt termékek vannak, amelyek elősegítik és lehetővé teszik formális projektirányítási módszerek alkalmazásával, mint például a PRINCE.

Az SSADM maga tartalmazza azokat a technikákat, amelyek egy teljes rendszerelemzés végrehajtásához, valamint egy információrendszer informatikai komponenseinek specifikálásához és megtervezéséhez szükségesek. Továbbá a felhasználói felület, az ember-gép kapcsolat felvázolására alkalmas technikákat, amelyek a manuálisan végrehajtható tevékenységeket megfogalmazását támogatják azért, hogy a szervezet teljes mértékben ki tudja használni az informatikai rendszerben rejlő lehetőségeket.

Ezek között a technikák között természetesen van egy bizonyos belső összefüggés (adatfolyam modellezés, logikai adatszerkezet, stb.), de nem mindegyikre van szükség minden egyes projektnél. Az SSADM4+ ad egy alap módszertani keretet, amelybe ezek a technikák elhelyezhetők (ld. 35. ábra) és bizonyos előfeltételek fennállnak. Ez a keret vagy alampinta adja a testreszabás mozgásterét módszertani értelemben akkor, amikor egy adott projekt környezethez kívánjuk illeszteni. Az SSADM korábbi verziói nagy hangsúlyt fektettek a szakaszokra és lépésekre, ezek pontos megfogalmazására, szabványos módszertani keret kialakítására. Noha ez nagy mértékben segítette a projekt irányítást, viszont egy fajta merevséghez, rugalmatlansághoz vezetett, környezethez való adaptációs képessége alacsony volt, nehezebben volt illeszthető az inkrementális és evolúciós rendszerfejlesztési megközelítésekhez.



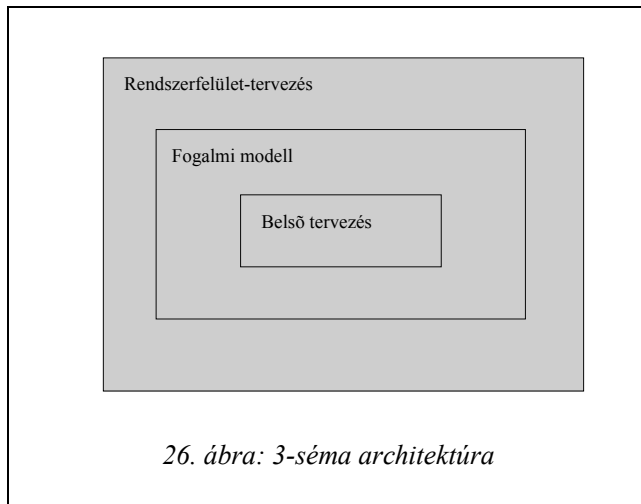
25. ábra: A rendszerfejlesztési alapminta és a technikák

Az SSADM4+ továbbra is tartalmaz egy strukturális modellt, amelyet a testreszabás kiinduló pontjaként lehet használni, de az ábrán látható alapminta egy sokkal rugalmasabb módszertani keretet sugall, amelyben bizonyos módszertani szempontok és az adott projekt céljai szem előtt tartásával egy megfelelő, testreszabott környezetet lehet kialakítani.

1.9.1.3 Az SSADM kulcsfogalmainak háttere

Az SSADM4+ két kulcsfogalma: a fejlesztési alapminta és a 3-séma architektúra (ld. 25. ábra, 36. ábra). Az SSADM kezdetben a szervezet működésének (üzleti környezetének) a vizsgálatára koncentrált azért, hogy minél jobban meg tudja határozni a leendő rendszerrel szemben támasztott követelményeket. Majd a(z informatikai) rendszerrel leírását, specifikációját és a kapcsoló-felületeket a valóságban működő szervezeti folyamatokhoz határozza meg. Az elemzés és a tervezés termékeit erre a fejlesztési alapmintára lehet leképezni, ezen megtalálhatóak a rendszerfejlesztés legfontosabb területei, nevezetesen:

- helyzetfelmérés;
- specifikáció;
- rendszerkészítés;
- felhasználói környezet;
- döntési pontok;
- szervezeti célok, politikák és eljárások.



26. ábra: 3-séma architektúra

Az információrendszerek készítésekor különbséget teszünk a(z informatikai) *rendszer* és a *külvilág* között. A *rendszer* specifikáció három fő területét jeleníti meg a 3-séma architektúra, ezen keresztül lehet látni azt, hogy az egyes termékeknek és technikáknak mi a feladata voltaképpen és a módszertan testreszabott verziója készítésekor világossá válik, hogy a séma egyes elemei közül mit és milyen

mértékben kíván a testreszabott változat megcélózni és teljesíteni.

Fogalmi modell:

a szervezeti, működési szabályok;

Logikai Adatmodell;

Entitás Viselkedés Modell;

Fogalmi szintű Adatfeldolgozó Folyamatok Modellje.

Ez a rendszer modell független a felhasználói felülettől, és különböző hardver és szoftver környezetben megvalósítható. A megvalósítás egyik lehetséges módja az, hogy a logikai adatfeldolgozó folyamatokat úgy készítik el, hogy azok a logikai adatmodell entitásain végezzenek olvasási és írási műveleteket.

Rendszerfelület-tervezés (Külső terv):

felhasználói felület, ember-gép párbeszéd;

be- és kimeneti adatok, állományok;

képernyők, jelentések;

dialógus tervek, programok, kötegelt adatfeldolgozás be- és kimeneti programjai.

A rendszerfelület terve egy kompromisszum:

a szervezet felépítése;

a rendszer hatékonysága, teljesítménye;

a végfelhasználói felület megvalósításának technológiája;

az egyes felhasználók egyedi kívánságai;

a biztonsági, auditálási előírások, stb.

között.

A Belsőterv:

fizikai adatterv (esetleg optimalizált a teljesítmény igényekre);

adatfeldolgozó folyamatok és fizikai adatok közötti kapcsoló felület (folyamat-adat kapcsolat);

A fizikai terv is egy kompromisszum:

a válaszidők, időzítési és idő korlátok;

háttértár;

karbantarthatóság;

között.

1.10 Objektum-orientált módszertanok

UML / OMT – Unified modeling Language / Object Modelling Technique (Objektum Modellezés Technikája)

Ezt a módszertan családot is érdemes a teljesség kedvéért megemlíteni. Az UML az OMG (Object Management Group) felkérésére készült el. A cél az volt, hogy egy egységes, szabványos jelöléstechnikát alakítsanak ki az objektum-orientált alkalmazások leírására. Az UML a technológia jelenlegi állásának megfelelően, a legkorszerűbb jelöléstechnikát tartalmazza. A folyamatos fejlődés eredményeként több módszer és jelöléstechnika eredményeit hasznosítja (Booch félé, Object Modeling Technique, OOSE, Object Oriented Software Engineering, OMT).

Vannak akik azt állítják, hogy ez a módszertan lesz a legelterjedtebb típus ([Rumbaugh91], [Shlaer88], [Meyer88], [Coad91]). Minthogy eddig nem mutattuk be ennek az elemzési típusnak az alapfogalmait, ezért azzal kell kezdenünk és az egyik legnépszerűbb módszertan az OMT rövid ismertetésével zárjuk. Ez a megközelítési mód bizonyos területeken nagy sikereket ért el; nevezetesen a termékként forgalmazott szoftverek körében, a felhasználói és grafikus felületek, ember-gép kapcsolat területén (pl. ablakos rendszerek). Ez a megközelítési mód terjedőben van az információrendszerek területén is, de a terjedés sebességét korlátozza az alaptechnológia nevezetesen az információrendszerek készítésére alkalmas objektum-orientált fejlesztő rendszerek — objektum-orientált adatbáziskezelő rendszerek — elterjedtségének hiánya.

1.10.1.1 Az objektum-orientált megközelítés alapfogalmai

Egy objektum-orientált rendszertől megkövetelt tulajdonságok - amiben a szakirodalom nagyjából egyetért - a következők:

identitás;

- osztályba sorolás;
- polimorfizmus;
- öröklődés;
- objektum, objektum példány és objektumok osztálya;
- a metódusok (objektumoknál kezdeményezhető eljárások, tevékenységek, program rutinok);
- beágyazás.

Definíció 1-1 Objektum

az adatok és az adatok viselkedésének szintézise; az adatok és azokat kezelő programok, program rutinok egyesítése.

A strukturált módszertanokat ismerőknek erről a definícióról az entitás fogalma juthat eszükbe, ami majdnem fedi is az objektum fogalmát egy kis pontosítás azonban szükséges.

Definíció 1-2 Objektum példány

Egyedileg azonosítható valami (identitás), ami a vizsgált probléma területen belül fontos, a szervezet számára jelentősége van. (Az entitástól annyiban különbözik, hogy magában foglalja az adatfeldolgozással kapcsolatos feldolgozó eljárásokat is.)

Itt az adott dolog egyedi, individuálisan felbukkanó példányairól beszélünk, amelyek a szervezet számára valóságosan létező dolgok, ezért a felhasználóval folytatott konzultációk során felismerhetők.

Definíció 1-3 Osztály (Class)

Objektum példányok csoportja, amelyek sajátosságai hasonlóak (attribútumok), közös viselkedést mutatnak (azonosak a hozzájuk kapcsolt metódusok, eljárások), más objektumokkal azonos módon állnak kapcsolatban, és szemantikájuk is megegyezik (ugyanazt jelentik, ugyanaz a jelentésük a szervezet számára).

Definíció 1-4 Metódus

Ügyfél	Objektum neve
Név Cím	Attribútum, látható változó
létrehoz megjelenít módosít törölés	Metódus

27. ábra: Egy objektum specifikációja

A metódus olyan tevékenység vagy transzformáció, amelyet egy objektum végrehajt vagy elszenved mint az alanya az adott eljárásnak.

Definíció 1-5 Beágyazás

A metódusok hozzáláncolását jelenti a megfelelő objektum osztályokhoz.

Ennek nagyon nagy jelentősége van. Ide tartozik az, hogy ezzel az objektumok 'logikai' sajátosságait elválasztjuk a leendő fizikai megvalósításától. Ennek az a célja, hogy az adatfeldolgozás függetlenségét megőrizzük az adatokkal szemben, azaz egy objektum leendő megvalósítását anélkül lehet megváltoztatni, hogy az az objektumot használó alkalmazásra hatást gyakorolna. A beágyazás rokon a *tűzfal* fogalmával, amely megakadályozza a jogosulatlan adat-hozzáféréseket és adatmanipulációkat. (Ezt a fogalmat a hálózatoknál, nyilvános adatbázisoknál is használják.)

Definíció 1-6 Identitás

Annak a kifejezése, hogy az adat diszkrét, megkülönböztethető entitás, objektum példány. Ezek az objektumok egyedileg azonosíthatók, még akkor is, ha az attribútumaik ugyanazok is.

Definíció 1-7 Osztályozás

Az objektum példányok olyan csoportosítása, amikor az azonos attribútummal, adatszerkezettel és viselkedéssel rendelkező objektumokat egy objektum osztályba soroljuk be.

Definíció 1-8 Polimorfizmus

Ez a fogalom azt írja le, hogy ugyanazt a műveletet (az objektumhoz szorosan kötődő metódust) különböző objektum osztályokon lehet használni, de az objektum osztálytól függően különböző módon viselkedhet. Például a *'rendelés elfogadás'* a *rendelés* objektum egy konkrét példányát hozza létre, a *termék* objektum állapotát illetve egy attribútumát, pl. a *'rendelkezésre álló mennyiség'* értékét módosítja.

Definíció 1-9 Öröklődés

Ez a fogalom akkor jelenik meg, amikor az objektum osztályok között közösen használt metódusok és attribútumok jelennek meg. Ugyanaz az attribútum több objektum osztályban jelenhet meg, az osztályok között fennálló hierarchikus kapcsolatok révén. A hierarchia alsóbb szintjein megjelenő alosztályok a főosztályaik összes attribútumát öröklik, ezt az alap attribútum készletet egészítik ki a saját egyéni adataikkal, attribútumokkal. Továbbá ugyanaz a művelet, metódus érvényes lehet több objektum osztályra is, az öröklődés révén.

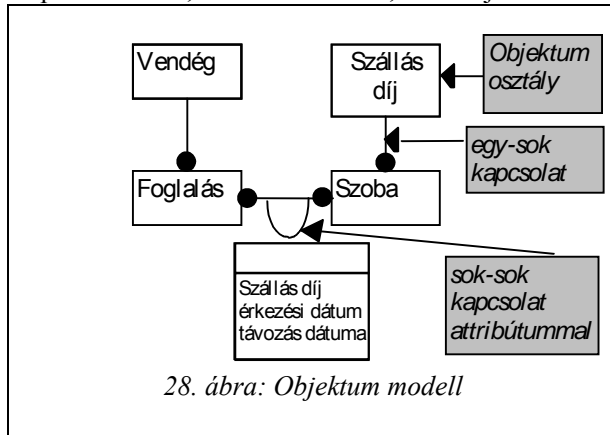
1.10.1.2 Az OMT három modellje

Az OMT is elfogadja a modellezés egyik alapelvét az **absztrakciót**. Vagyis az objektumok lényegesnek tartott oldalaira koncentrál, vagyis arra, hogy mi az objektum tipikus viselkedése, és nem a megvalósítás kérdéseire. Ez a megközelítés megegyezik azzal, amit a különböző strukturált módszertanoknál láttunk az adatszerkezetek, folyamatok, az adatok viselkedésének elemzésénél. Az OMT természetesen gondoskodik grafikus diagram technikáról is a különböző modellezési oldalak leírására. Az OMT három fajta modellezési eljárást alkalmaz:

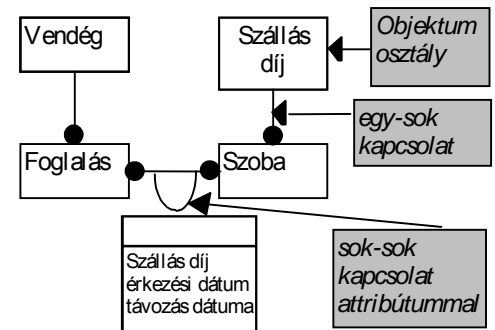
- objektum modell;
- dinamikus modell;
- funkcionális modell.

Az objektum modell

Az objektum modell az objektumok statikus, időben állandó szerkezetének, közöttük fennálló kapcsolatoknak, attribútumaiknak, a hozzájuk kötődő műveleteknek leírását jelenti.



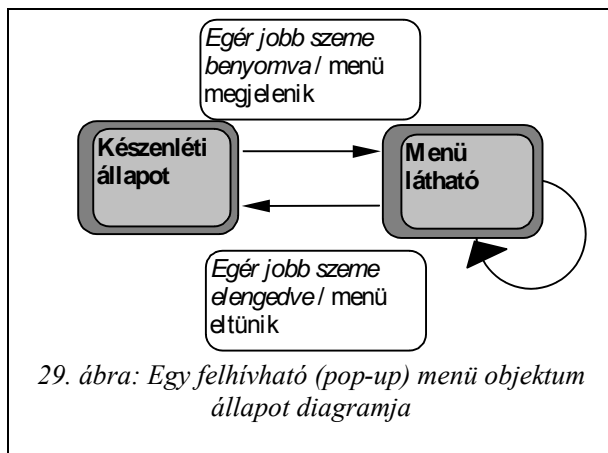
A grafikus megjelenített objektum modell (Isd. 28. ábra)



28. ábra) mutatja az objektumok között fennálló kapcsolatokat is, a már jól ismert egy-egy, egy-sok, sok-sok, opcionális esetek kombinációinak.

A dinamikus modell

A dinamikus modell a rendszer időben változónak tekinthető oldalát írja le. A dinamikus oldalt lehet arra használni, hogy az adatok összhangjának a fennállását (integritását) biztosítsák, valamint az objektumok érvényesnek tekintett állapotait írja le, továbbá az események által okozott állapotok közötti átmeneteket. Az objektum állapota az attribútumainak értékéből látható, valamint az általuk megvalósított kapcsolatok is ezekből ismerhetők fel. Az állapotváltozások az események hatására következnek be, amelyeket az egyik objektum által a másikkra gyakorolt hatás, (stimulus) formájában foghatók fel. Az eseményeknek szintjén vannak osztályaik és ezeknek pedig példányaik.



Az állapot diagram összekapcsolja az esemény osztályokat és az állapotokat.

A funkcionális modell

A funkcionális modell azokat a transzformációkat írja le, amelyeket az adatokon hajtanak végre a műveletek (metódusok).

Az alkalmazott diagram technika az adatfolyam modellezés; ez megmutatja, hogy a rendszernek milyen adat transzformációkat kell végrehajtani anélkül, hogy tekintettel

kellene lenni arra, hogy hogyan, mikor, hol és ki hajtja végre az adatfeldolgozást.

