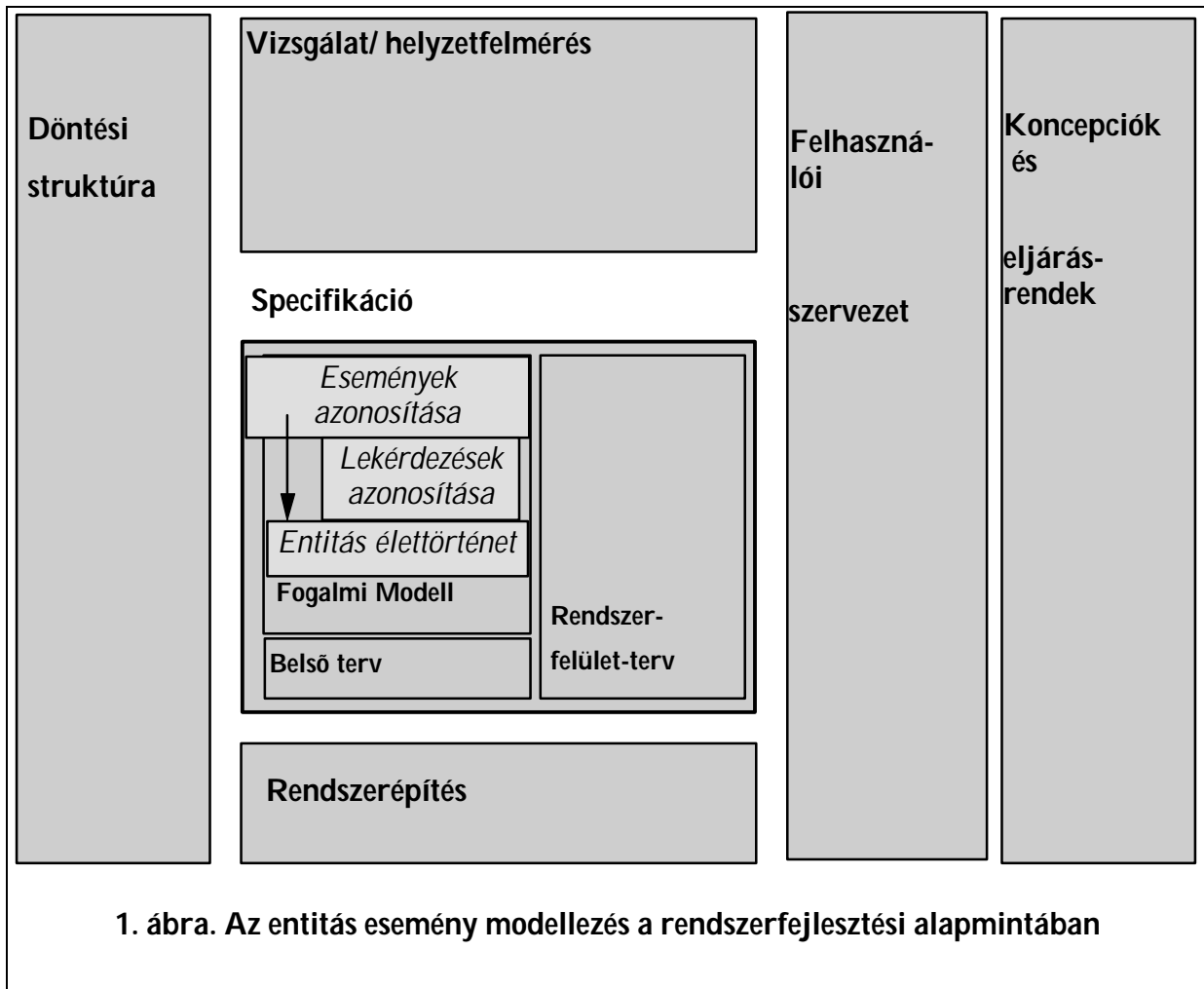


1. Áttekintés az entitás-esemény modellezésről az SSADM4+-ban¹

Az entitás-esemény modellezésbe azok a technikák tartoznak, amelyek a 'Fogalmi modell' adatai és folyamatai közötti kapcsolatokat írják le. Az ábra (1. ábra.) mutatja ezeknek a technikáknak a helyét a rendszerfejlesztési alapmintában.



Ebbe a technika halmazba tartoznak:

- az esemény azonosítás;
- a lekérdezések azonosítása;
- az entitás életének a modellezés.

Az entitás élettörténet készítése az egyik legjelentősebb technikája az SSADM-nek. Ez a technika a magas szintű² adatfeldolgozási folyamatok helyességét és érvényességét ellenőr-

¹ [CCTA95], [CCTA95A], Reference Manual Part 6: Modelling from the System's Perspective, Entity Behaviour Modelling, 6-9—6-57, Users Guide Part 3: Specification (Conceptual Model), Entity Behaviour Modelling, 3-59—3-90. Továbbá [CCAT90].

zi³, továbbá tisztázza és azonosítja a sokkal részletesebb adatfeldolgozási és adatszerkezeti követelményeket.

1.1 Cél

Az események és a lekérdezések alkotják azokat az építő elemeket, amelyekből az igényelt rendszer adatfeldolgozási specifikációja felépül. Az események és lekérdezések indítják a fogalmi modell azon folyamatait, amelyek az igényelt rendszer logikai adatmodelljének adatszerkezetében a keresést, navigálást és az adatok manipulálását végzik el. Az illetékes funkciókhoz kapcsolják az eseményeket és a lekérdezéseket, mivel a funkciók egy fajta szűrőként működnek a felhasználó és az informatikai események és lekérdezések között (2. ábra.):

- a funkció közvetíti a fogalmi modell felé az eseményeket és a lekérdezéseket, a felhasználók által előállított bemeneti adatokból az esemény- és lekérdezés-kiváltó adatok kiemelésével;
- a funkció fogadja az események és lekérdezések kimeneteit, a kimeneti adatokat pedig a felhasználó számára értelmezhető formában állítja elő.

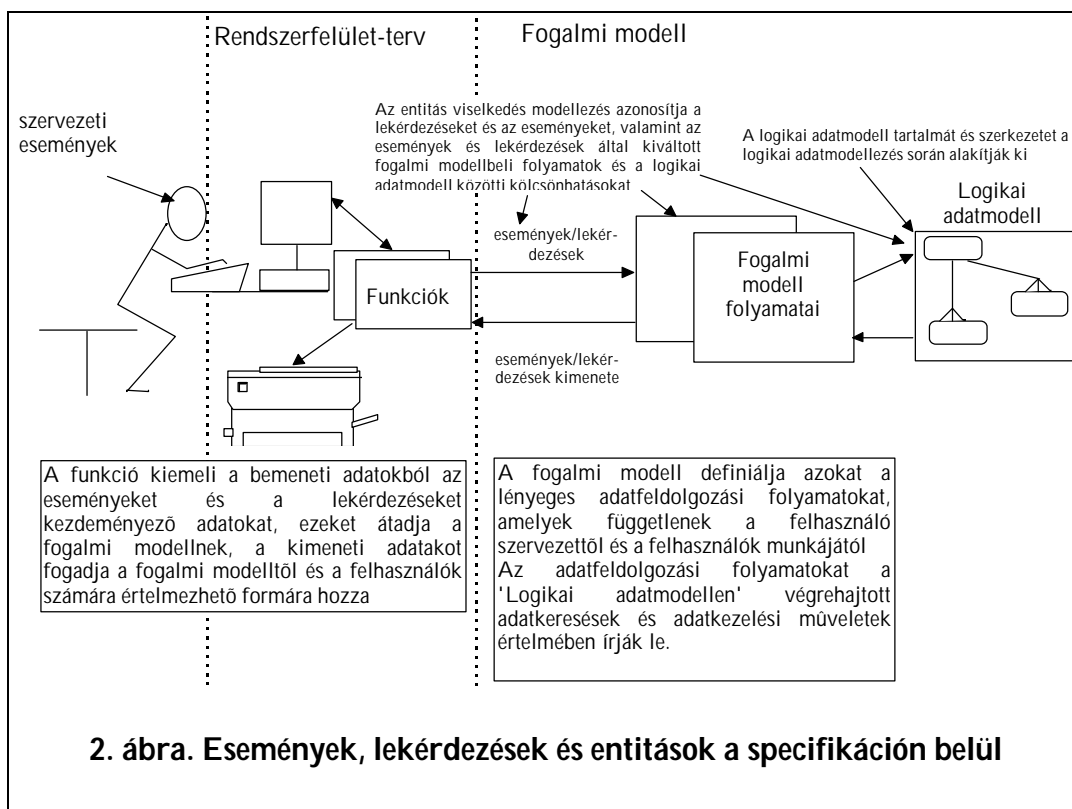
Az esemény azonosítás az a technika, amely az igényelt rendszer adatfolyam és adatmodelljének vizsgálatából meghatározza az összes eseményt.

A lekérdezések azonosítása az a technika, amellyel a felhasználók és a szervezeti tevékenységek információtámogatási igényeiből vezeti le a lekérdezéseket. A lekérdezések leírása és meghatározása segíti a logikai adatmodell elkészítését és érvényességének ellenőrzését.

Az entitás élettörténet technikát arra használják, hogy a logikai adatmodell egy adott entitását érő esemény hatások sorrendjét leírják. Ezt azért kell csinálni, hogy az entitások aktualizálására vonatkozó peremfeltételeket, korlátokat rögzítsék és feltárják, hogy a szervezeti események és a működési szabályok milyen mértékben jelenjenek meg a rendszer adatainak aktualizálására vonatkozó követelményekben. Az entitás élettörténet elemzés végeredménye beépül a fogalmi modellbe.

² Magas szintű adatfeldolgozási folyamat alatt azokat a folyamatokat értjük, amelyeket azok az informatikai események váltanak ki, amik közvetlenül egy szervezeti szintű eseményből származnak.

³ Azaz validálja és verifikálja az elképzelt adatfeldolgozási folyamatokat.



Az ide tartozó technikák, az esemény és lekérdezés azonosítás, entitás élettörténet elemzés segíti a funkció-meghatározás és a logikai adatmodell helyességének és érvényességének ellenőrzését, valamint a tovább fejlesztésüket.

1.2 A fogalmi modell kifejlesztése

A rendszerrel szemben támasztott követelmények meghatározásának lényeges elemei az események, lekérdezések, az entítások és a kölcsönhatásaik. Ezeket az elemeket modellező technikák a fogalmi modell készítésének központi részét alkotják. Az események, lekérdezések, entítások helyét a specifikáción belül az ábra érzékelteti (2. ábra.). Van néhány lényeges összefüggés a fogalmi modell termékeinek kifejlesztése során:

- a szervezet működése által igényelt információ támogatási követelményeket a 'Követelményjegyzék' tartalmazza, amelyet a helyzetfelmérés során fejlesztettek ki, ezért a követelményjegyzék a lekérdezések azonosítása szempontjából kulcsszerepet játszik. A követelményjegyzékben rögzített információ-támogatási igények a következő típusúak lehetnek:
- lekérdezések;
- jelentések / beszámolók (riportok);
- ad-hoc és sürgős kérések;
- adatok átadása / exportja külső eszközöknek, pl. kalkulációs lapok;

- a követelményjegyzék bejegyzései a logikai adatmodell tartalmát segítik meghatározni, és a modell helyességét / érvényességét is ellenőrzik egyidejűleg. Ha van olyan követelmény, amelyet a logikai adatmodell nem tud kielégíteni, akkor a modellt a kívánalmaknak megfelelően át kell alakítani. Ha pedig a logikai adatmodell egyes részei nem találkoznak a követelményekben rögzített igényekkel, akkor ennek a résznek a helye megkérdőjelezhető a logikai adatmodellben;
- az igényelt rendszer logikai adatmodelljének a kialakítása és a szervezet információ iránti követelményeivel történő összekapcsolása után, lehetőség nyílik az adat-aktualizálással szemben támasztott igények elemzésére. Mivel a logikai adatmodell közvetlenül a szervezet információ igényét tükrözi vissza ezért a logikai adatmodell entitás példányain végrehajtandó változtatási szándékoknak egybe kell esniük a szervezet működésében változást okozó bizonyos eseményekkel;
- az (informatikai) eseményeket azokból szervezeti eseményekből ismerhetik fel, amelyek változásokat okoznak a fogalmi modell adataiban. A rendszer úgy értesül az egyes események bekövetkezéséről, hogy egy vagy több funkcióhoz bemenő adatok érkeznek, amelyek a funkciók indítását váltják ki;

A fogalmi modell a szervezeti tevékenységek megértése és az információ támogatási igények felismerésén keresztül egy feltáró, felfedező jellegű folyamatban jön létre. Amikor, már tudják, hogy milyen információ támogatásra van szükség, akkor kialakíthatják a fogalmi modellt, amely összhangban áll az információ-támogatással szemben támasztott igényekkel és a szervezet környezetében bekövetkező események által kiváltott hatások adat-aktualizálási következményeivel.

1.3 Összefoglaló áttekintés

Az esemény azonosítás vezeti le az igényelt rendszer adatfolyam modelljéből és az igényelt rendszer logikai adatmodelljéből az eseményeket. Az *események* és a *funkciók* azonosítása történhet **párhuzamosan** is. A funkciók felismerése segítheti az események azonosítását. A szervezeti tevékenység modell (BAM) szintén segítheti a azoknak a szervezeti eseményeknek feltárását, amelyek megfeleltethetők olyan (informatikai) eseményeknek, amik a fogalmi modell adatain hajtanak végre módosításokat.

A lekérdezéseket a szervezeti tevékenységek információ-támogatási igényeiből lehet felismerni.

Az eseményeket és a lekérdezéseket az 'Entitás-használati mátrix' segítségével össze kell rendelni az igényelt rendszer logikai adatmodelljével.

Az eseményeket és a lekérdezéseket az 'Esemény és lekérdezés jegyzékben' kell dokumentálni.

Az entitás élettörténet elemzést az entitásokkal való történések felismerésére, az entitások életét befolyásoló események leírására használják, továbbá az események hatásának módját és sorrendjét dokumentálják ezzel a technikával. A legjelentősebb logikai adatbázis műveleteket is azonosítják ebben az eljárásban.

2. Entitás viselkedés modellezés

2.1 A technika célja

Az entitás-esemény modellezés két technikát jelent, az entitástörténet elemzését (ELH) és az eseményhatás-elemzést (ECD).

Az entitástörténet-elemzés egy nagyobb technika az SSADM-n belül. Ellenőrzi a magas szintű feldolgozási folyamatok és a logikai adatmodell érvényességét, valamint további részletes feldolgozási és adatokra vonatkozó követelményeket tár fel.

Az eseményhatások elemzése a rendszer követelményeinek egy eseményközpontú nézőpontját adja, aminek az eredményét a logikai rendszertervezés során a módosító feldolgozási modellek kialakításakor kell felhasználni.

A 360. lépés során ("Feldolgozási folyamatok meghatározása") az entitástörténeti technikát a funkcióleírások érvényesítésére használják.

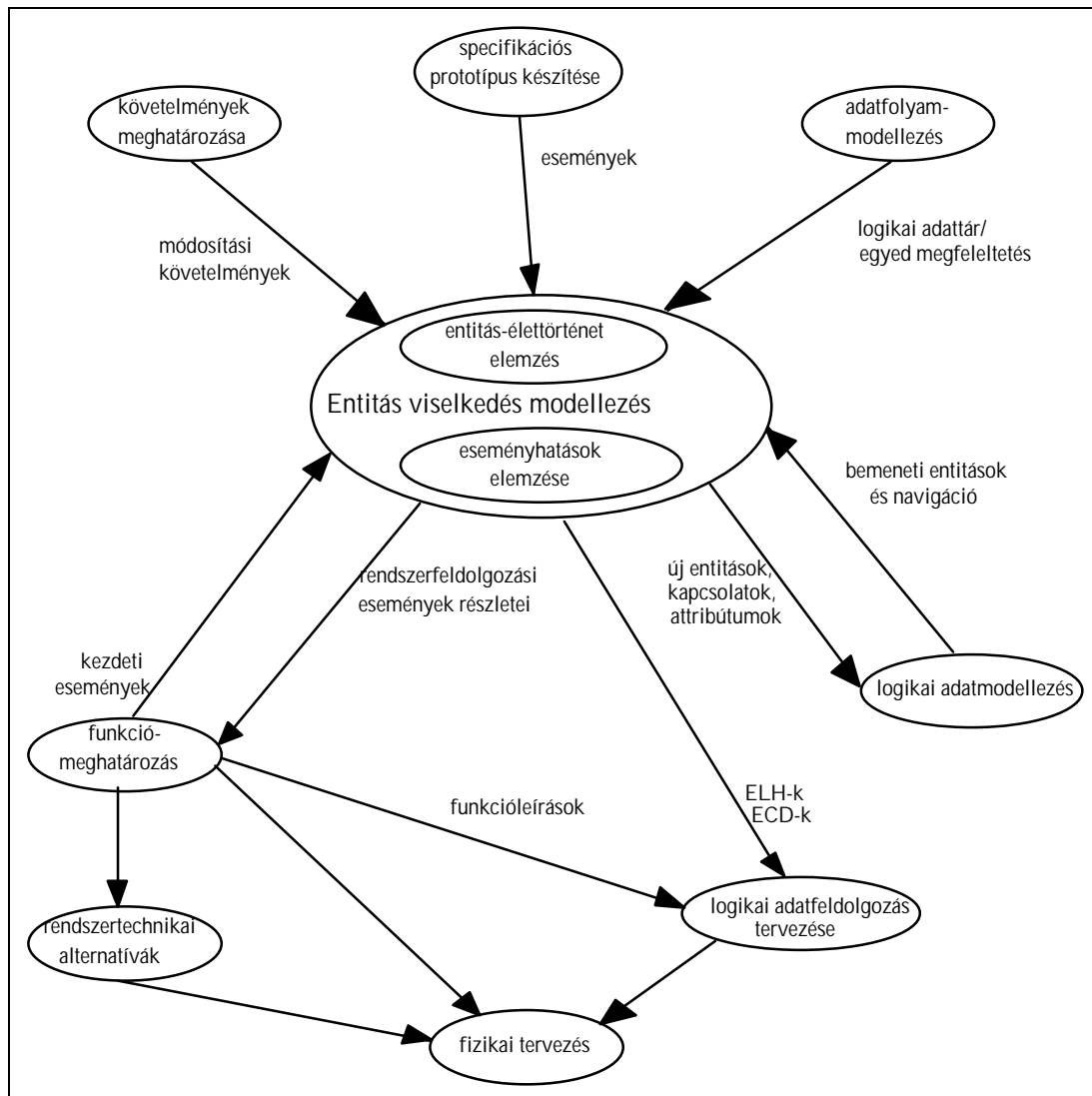
Főbb tevékenységek:

- I. Az adatfeldolgozás további részleteire vonatkozó követelményeket azonosítják, ami a funkcióleírások módosítását vonhatja maga után.
- II. A logikai adatmodell (LDM) helyességének ellenőrzése:
 - A. Az LDM hibáira és hiányosságaira az entitás viselkedés elemzése során derülhet fény. A gyakorlat azt mutatja, hogy az LDM jelentősen módosul ennek eredményeképp.
- III. További szervezeti / működési szabályok felismerése és rögzítése:
 - A. A logikai adatmodellben modellezett működési szabályokat az entitástörténeti elemzés felhasználja és továbbfinomítja adatfeldolgozási szempontból, egészen a fizikai tervezésig, ahol az LDM-mel való átfedéseit feloldják.
- IV. A perem - és kényszerfeltételek meghatározása:
 - A. Az igényelt rendszer belső megszorításait azonosítják és dokumentálják, meghatározva az események prioritását és sorrendjét. Továbbá meg határozzák az informatikai, logikai adatbázis műveleteket, amelyek az attribútumokkal és a kapcsolatokkal foglalkoznak.

A felhasználóval meg kell vitatni az események sorrendjét és a megszorításokat, feltételeket, mivel ezeket továbbkerülnek a logikai és fizikai tervezésbe és a végső rendszerben is szükség lesz rájuk. Az entitástörténeti ábrák (ELH-k) a működési szabályokat írják le, az eseményhatás-ábrák (ECD-k) a rendszer adatfeldolgozási szerkezetét tükrözik. Emiatt az ELH-kat és ECD-ket a felhasználóval szorosan együttműködve kell kialakítani. A felhasználónak képesnek kell lennie még arra is, hogy a "normálistól" eltérő szervezeti folyamatokat, a kivételes

helyzetek kezelését, a szervezeti szintű hibakezelést leírja. Ezt a felhasználói szerepkört a gyakorlatban több ember is betöltheti.

A logikai specifikáció szakaszában, az ELH-k és ECD-k a fogalmi folyamatok tervezéséhez adnak kiindulópontot. Minden eseményhez egy módosító feldolgozási modellt határoznak meg, amely a módosítási folyamat specifikációját és abban az integritási hibák felismerését foglalja magában.



3. ábra. Az entitás viselkedés modellezés kapcsolata más SSADM technikákkal

2.2 A technika rövid leírása

Az entitás-élettörténet technikáját az entitások életének vizsgálatára lehet használni, meghatározva az életüket befolyásoló eseményeket, dokumentálva a befolyásolás módját és megmutatva azt a sorrendet, amelyben a hatások bekövetkeznek. A hatásokhoz tartozó fontosabb logikai adatbázis műveleteket is meg lehet határozni.

Az eseményhatás elemzéssel a rendszer részletes adatfeldolgozási folyamatait lehet leírni, meghatározva egy adott esemény-előfordulás hatását az entitásokra.

Alapfogalmak

Definíció 2-1 Szervezeti és informatikai esemény

Az elemzés, a termékek készítése és azok készítési módjának leírása során szigorúan meg kell különböztetni a 'szervezeti eseményeket' és az '(informatikai) eseményeket'. A keveredés elkerülése végett a következő informális útmutatásra lehet támaszkodni:

- szervezeti eseménynek valami olyan eseményt tekinthetünk, amely a szervezet környezetében következik be és a szervezetnek reagálnia kell rá. A szervezeti események hatása nem mindig érinti az automatizált rendszerben tárolt adatokat. Példák szervezeti eseményekre 'ügyfél érkezik', vagy 'balesetes gépkocsi kerül vissza'.

informatikai esemény az, amely a(z informatikai) rendszer környezetében következik be és amelyre az automatizált rendszernek az adatok aktualizálásával kell reagálnia. Egy szervezeti esemény kiválthat informatikai eseményt, de természetesen nincs szükségszerűen egy-egy értelmű megfeleltetés közöttük. A gépkocsi kölcsönzési példában, amikor egy ügyfél betér a fiókhoz, akkor több olyan szervezeti eseményt indíthat el, amiknek semmi kapcsolata sincs az automatizált rendszerrel, egészen addig, amíg bizonyos előidézett szervezeti események le nem "fordítódnak" 'Alkalmi gépkocsi kölcsönzés' vagy 'Átírányított gépkocsi átvétele' nevű (informatikai) eseményre. Egy másik esetben a 'közúti balesetben megsérült gépkocsi' és 'a parkolóban megsérült gépkocsi' egy informatikai eseményre fordítódik 'sérült gépkocsi'.

Egy (informatikai) esemény az olyan valami, ami egy a rendszeradatokat megváltoztató feldolgozási folyamatot indít el, vagyis a fogalmi folyamat modell egyik folyamatának elindítását kezdeményezi.

Definíció 2-2 Lekérdezés

Lekérdezés az, ami elindít egy fogalmi modellbeli folyamatot azért, hogy a rendszer adataiból információt gyűjtsön össze, de semmiféle aktualizálást nem hajt végre a logikai adatbázison.

Definíció 2-3 Entitástípusok és entitás-előfordulások

Ebben a részben megkülönböztetjük az entitások típusát és előfordulását, az entitások egyedi példányát.

Az entitás objektumoknak, tárgyaknak, dolgoknak, fogalmaknak egy osztálya, nem egyedi valami. A "Személy" nevű entitástípus nem jelöl egyetlen konkrét személyt sem, hanem az összes olyan embert jelzi, akiről információt akarunk tárolni. Nyelvtanilag a gyűjtőfogalomnak felel meg.

Az entitás-előfordulás vagy az entitás példány azt az egyedi dolgot, objektumot jelöli, amelyről információt akarunk tárolni. Az entitás példány pedig az egyedi fogalomnak felel meg.

Minden entitástípusnak lesz egy attribútum-halmaza, amely leírja azt az entitástípust, pl. "Név", "Cím", "Születési hely", stb. Minden egyes entitás-előforduláshoz ezeknek az attribútumoknak valamilyen konkrét értéke fog tartozni. Ha az entitás-előfordulás Kovács Jánost jelöli, akkor a név "Kovács János" lesz és a további attribútumok a neki megfelelő adatokat tartalmazzák.

Az elemzőnek tudatában kell lennie annak, hogy az entitás típus és a példány is csak a valóság egyfajta reprezentációja, leképezése, ennek a leképezésnek a megjelölésére használjuk az entitás típus illetve a példány nevét (azonosítóját).

Definíció 2-4 Entitás-élettörténet

Az entitás-élettörténet ábrázolja az összes eseményt, amely bármelyik entitás-előfordulás valamilyen megváltozását okozhatja.

Az entitás-élettörténet egyrészt az entitás bármelyik példányának az összes lehetséges életét, az összes elképzelhető az entitást érő eseményhatás leírását jelenti. Bármelyik entitás-előfordulásnak úgy kell viselkednie, ahogy azt az adott entitás ELH-ja meghatározza⁴.

Definíció 2-5 Hatás

Egy esemény által okozott egyetlen entitás-előfordulás megváltozását **hatásnak** hívjuk, amely valójában logikai adatfeldolgozási / adatbázis műveletek együttese. Minden hatásnak meg kell jelennie az entitástörténet ábrán, a hatás jele egy téglalap, amibe a hatást kiváltó esemény neve kerül. A hatások a faszerű ábra levelein jelennek meg.

Definíció 2-6 Hatás-minősítő⁵

Ha egy adott esemény bekövetkezése nem minden esetben váltja ki ugyanazt a hatást, hanem egy kissé különböző adatfeldolgozási folyamatot idéz elő⁶ akkor ezt ezzel a minősítővel jelöljük. Egy esemény számos, egymást kölcsönösen kizáró módok egyike révén érintheti az egyedi entitás példányt. Minden egyes lehetséges hatást külön jelölnek az entitás-élettörténet ábrán, felhasználva az esemény elnevezését, "gömbölyű" zárójelbe helyezve a hatás-leírást, hatás-minősítőt.

⁴ Tehát az entitástörténet ábrán mindig egy, konkrét, egyedi entitás példány életét követhetjük végig (egyedi azonosító, az attribútumokhoz rendelt megengedett értékekkel az értelmezési tartományukból). Az ábra egészének pedig úgy kell kinéznie mint az állatorvosi beteg lónak, amelyen az összes lehetséges "betegséget", vagyis a kivételes és nem-kivételes (normális) események hatását is érzékeltetni lehet.

⁵ **esemény altípusnak** is nevezhetjük ezt a jelenséget.

⁶ Ezt úgyis fogalmazhatjuk, hogy a logikai adatbázis műveletek, amelyek az esemény által kiváltott hatást valósítják meg a logikai adatbázisban, kissé különböznek, pl. a sorrendjükben, esetleg nem ugyanazok a műveletek

Definíció 2-7 Entitás-szerep (egyed-szerep)

Egy esemény egy konkrét előfordulása okozhatja egynél több entitás példány megváltozását. Ha egy esemény ugyanazon entitástípus egynél több előfordulását különböző módon befolyásolja, a különböző hatásokat **entitás-szerepnek** hívjuk. Az entitás-szerepeket úgy jelölik, hogy az entitás-élettörténet hatás-dobozaihoz az esemény neve alá szögletes zárójelek közé zárva hozzáírják.

Megjegyzés: Valamely hatásnak lehet entitás-szerepe és hatás-minősítője is, de valószínűbb, hogy vagy csupán egyikkel, vagy a másikkal rendelkezik.

Definíció 2-8 Felettes-esemény (szuper esemény)

Ha egy entításra életének ugyanazon pontján több különböző eseménynek pontosan ugyanaz a hatása, a hatások olyan névvel láthatók el, amely valamennyit kifejezi összefoglalóan - ez a felettes-esemény neve. Ha az eseményeknek ugyanez az együttese egy másik entitás élettörténetében megjelenik, ismét ugyanazzal a hatással, az egyes hatásokat nem szükséges külön elnevezni (a hozzátartozó eseményről), hanem helyette a felettes-esemény elnevezése használható. Az eseményhatás ábrákon a felettes-események közhasznú folyamatokká változnak, melyeket a felettes-eseményt alkotó minden esemény meghív, azaz mindegyik esemény kiváltja ugyanazt a hatást az adott entitás és esemény kontextusában⁷. A felettes-eseményeket az entitás-elérési táblázatban dokumentálják, az esemény- és lekérdezés-jegyzékben írják le és az eseményhatás ábrákon modellezzik.

Definíció 2-9 Közhasznú lekérdezés

A lekérdezés egyik típusa, mely azáltal ismerhető fel, hogy lényeges közös részt jelent több esemény nem-módosító hatása és/vagy több lekérdezés között. Ahol ezt a közhasznúságot a rendszeren belül felismerik, ott a ezeket közös adatfeldolgozási területeket, mint önálló lekérdezéseket külön megjelölik és definiálják. Ezek a lekérdezések csak azoknak az eseményeknek vagy lekérdezéseknek hatására indulhat be, amelyekből ezeket a közhasznú lekérdezéseket felismerték, azonosították, absztrahálták.

Minden közhasznú lekérdezést ugyanúgy kell azonosítani és dokumentálni, mint más lekérdezéseket. A közhasznú lekérdezések modellezése a lekérdezési utak segítségével történik.

Definíció 2-10 Halál vagy törlés

Valamely entitás példánynak az az állapota, amelyben nincs többé aktív szerepe a rendszeren belül. Az entitás példányhoz esetleg a lekérdezések hozzáférhetnek. A ha-

kerülnek alkalmazásra, vagy más attribútum értékeket kapnak ugyanazok az attribútumok, esetleg más attribútumokat fog módosítani az esemény hatása az esemény különböző előfordulásakor.

lált követő állapot rendszerint a törlés. Ez általában a konkrét információrendszerben követett archiválási stratégiától függ, amely gyakran olyan egyszerű korlátoktól függ mint a rendelkezésre álló merevlemez tároló kapacitás. Az entitástörténet diagrammon a törlést okozó események feltüntetésére nincs szükség, lehet, hogy az elemzés idején nincs is információ erről, azonban ha van, akkor a teljesség kedvéért érdemes feltüntetni.

2.3 Az entitás viselkedés modell termékei

A következő termékeket kell elkészíteni az entitás viselkedés modellezés során:

- entitás-elérési táblázat (egyed-elérési táblázat);
- esemény és lekérdezés jegyzék / katalógus;
- entitástörténet (entitás-élettörténet).

2.4 Jelölésmód és fogalmak

2.4.1 Fogalmak

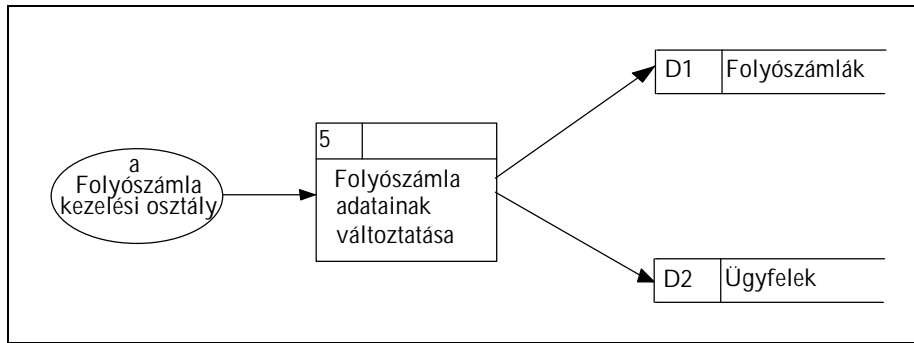
2.4.1.1 Esemény

Egy esemény szolgáltatja az okot egy módosító feldolgozási folyamat indításához. Az esemény nevének azt kell kifejeznie, ami a folyamatot okozza, és nem a folyamatot magát. Tipikus eseménynévek olyan fogalmakat tartalmaznak, mint "Befogadás", "Visszajelzés", "Döntés", "Beérkezés", "Új", "Változás", ami mind a folyamatot okozó eseményre utal és nem a feldolgozásra magára. Ha egy esemény neve az adatfolyam-ábrán szereplő elemi folyamat nevét tükrözi, akkor meg van az a veszély, hogy ugyanazt a folyamatot indító más események elfelejtődnek.

A következő egyszerű ábrán szereplő egyetlen bemenő adatfolyam megtévesztheti az elemzőt, mivel esetleg feltételezheti, hogy az egyetlen adatfolyam egyetlen eseményt tartalmaz, tehát nyugodtan lehetne hívni az eseményt úgy, ahogy a folyamatot. Egy részletesebb elemzés ennek ellenére felderítené, hogy itt több eseményt kell azonosítani, mégpedig a következőket:

- folyószámla nyitás
- folyószámla megszüntetés
- ügyfél nyilvántartásba vétele
- ügyfél adatainak változása

⁷ Minthogy a hatás ugyanaz, ez azt jelenti, hogy lényegében a logikai adatbázis műveletek is ugyanazok, azaz egy többször alkalmazható művelet együttest, rutint, közhasznú folyamatot találtunk. Objektum orientált környezetben ez egy magasabb osztály szintjén meghatározható közös metódust jelent.



4. ábra. Bemenő adatokat ábrázoló adatfolyam-ábra

Ha a folyamat nevét használta volna az esemény megnevezésére, akkor a fenti események nem bukkantak volna fel.

Az eseményt a neve egyértelműen azonosítja a funkcióleírásokban, entitástörténeti elemzésben, az eseményhatás-elemzésnél és a fizikai folyamatok meghatározásánál egyaránt.

2.4.1.2 Hatások

Egy esemény egy entitás egy előfordulását négyféleképpen befolyásolhatja. Az entitás előfordulását:

- létrehozhatja
- módosíthatja
- törölheti

illetve az állapotjelzőjének az értékét változtathatja.

Egy esemény legalább egy entitás megváltoztatásának okozója, ezt hívják hatásnak.

Minden hatásnak szerepelnie kell a megfelelő entitás-élettörténeti ábrán. A hatásokat az entitástörténeti ábrán egy doboz jelöli, amiben az esemény neve szerepel. Ez lehetővé teszi, hogy egy adott esemény hatásait, az élettörténeti ábrákról kiindulva, összegyűjtsük és átvigyük az esemény kölcsönhatásait leíró ábrára.

Lehetnek olyan esetek, amikor egy entitás egy előfordulására egy adott esemény több egymást kölcsönösen kizáró módon hat. Ilyenkor minden egyes lehetséges hatást külön-külön azonosítani kell az élettörténeti ábrán az esemény nevével, kiegészítve azt egy zárójelbe zárt leírással a különbségről (**hatás-minősítő**). Az itt használt zárójelnek különböznie kell az **entitás-szerepkörök** jelölésére használt zárójelaktól. Általában az elsőt gömbölyű, a másodikat szögletes zárójel jelöli.

Például egy "Átutalás feladása" nevű esemény két különböző módon hat egy adott folyószámlára, attól függően, hogy a folyószámlán van-e elegendő fedezet vagy nin-

csen. Ilyenkor az élettörténeti ábrán az esemény kétszer fog szerepelni, a következő módon: "Átutalás feladása (elegendő fedezet)" és "Átutalás feladása (fedezethiány)".

2.4.1.3 Entitás-szerepkörök

Ha egyetlen esemény egy adott entitás egynél több előfordulására hat, és a hatás minden érintett előfordulásra különböző, akkor az entitás valószínűleg különböző "szerepeket" tölt be. Minden ilyen különböző "szerepet" meg kell különböztetni az adott entitás élettörténeti ábráján, mivel különböző feldolgozást igényel minden szerepkör. A különböző entitás-szerepkörök azonosítására az ábrán az esemény nevét ki kell egészíteni az adott entitás által betöltött szerepkör leírásával.

Például, ha egy "Belső átutalás" nevű esemény egy banki rendszeren belül két nyilvántartott folyószámla közötti átutalást jelöl, akkor ez az esemény a "Folyószámla" nevű entitás két előfordulására is hat. Azt az előfordulást, amely az átutalás kiinduló folyószámláját képviseli módosítani kell, levonva az átutalt összeget a folyószámla egyenlegéből. A másik előfordulást, amely az átutalást befogadó folyószámlát jelöli szintén módosítani kell, hozzáadva az átutalt összeget az adott előfordulás egyenlegéhez. A két hatásnak a rendszer számára egyidőben kell bekövetkeznie. Mivel egy adott entitás élettörténeti ábrája az összes előfordulás összes létező életét leírja, ezért a fenti eseményt kétszer kell felvenni az ábrára, megkülönböztetve az előfordulások szerepét, például a következő módon: "Belső átutalás [terhelési folyószámla]" és "Belső átutalás [jövőírású folyószámla]". Mindegyik folyószámla előfordulhat mindkét szerepben az élete során, de az esemény mindig folyószámla-párokra hat.

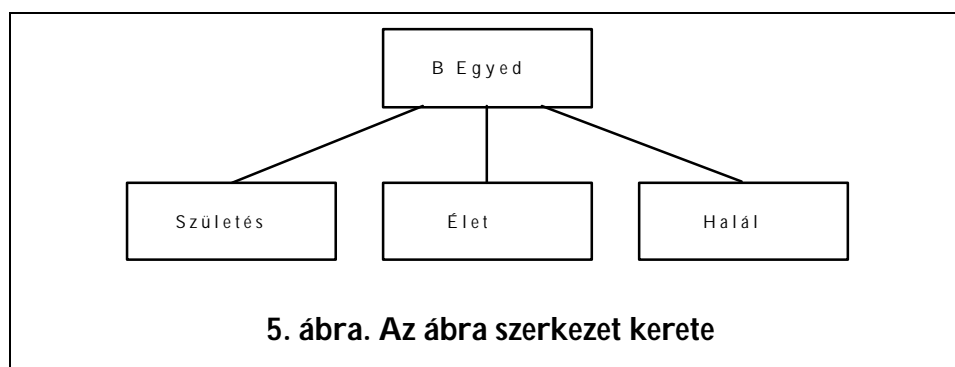
A hatások nevének és a szerepkörök nevének megkülönböztetése fontos, ezért a két különböző zárójelzést nem szabad összekeverni.

2.4.2 Jelölésmód

2.4.2.1 Entitás-élettörténet

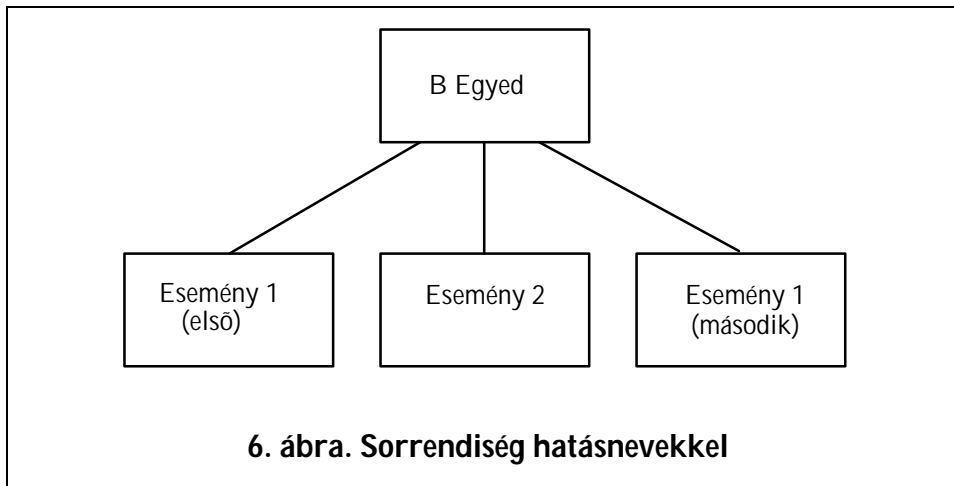
Az entitás-történeti ábrák a Jackson módszer jelöléseit követik. Van egy kiegészítő jelölésmód a kilépések és folytatások jelölésére. Az ábra elemei szögletes sarkú téglalapok, dobozok, az ábraszerkezet tetején lévő doboz az entitástípust jelöli és az entitás nevét viseli.

Sorrendiség (Szekvencia)



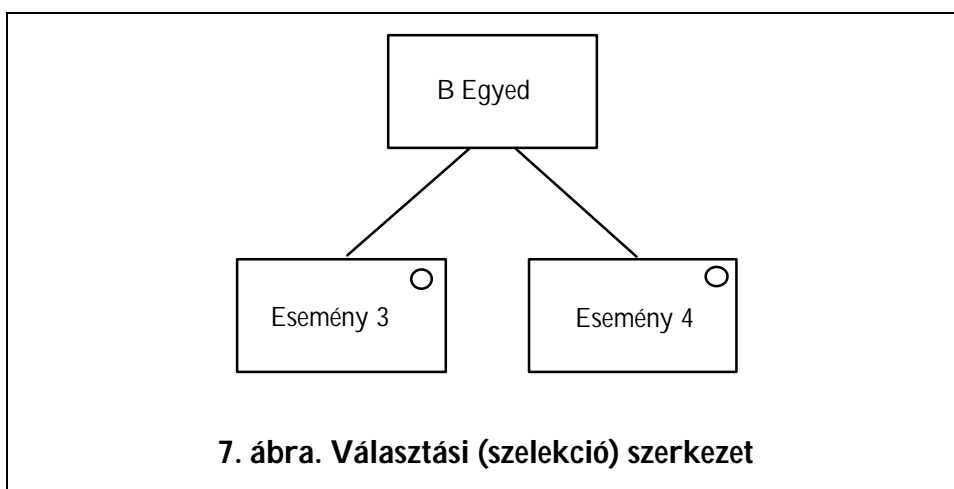
A sorrendiségi szerkezet az ELH ábra alapja. A "Születés", "Élet", "Halál" általában jó keretet jelent minden ábrához.

Egy esemény egy entitás életében többször is előfordulhat, különböző hatásokat kiváltva. Egy adott esemény okozhatja egy entitás születését illetve halálát. Például ha az esemény neve "Folyószámla változtatás", akkor ez jelenthet egyszer folyószámla nyitást, máskor folyószámla megszüntetést.



Választás (szelekció)

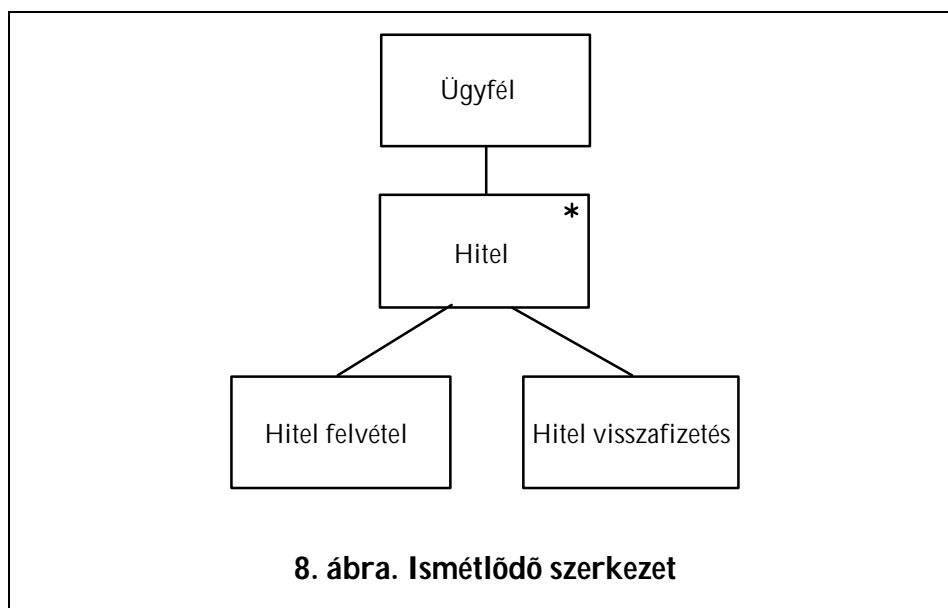
A választás jelölésénél nincsenek hozzárendelt feltételek. A választás jelzésével azt kell kifejezni, hogy az entitás-előfordulásokra különböző események hatnak egy adott ponton. A következő ábra azt mutatja, hogy vagy a 3. vagy a 4. eseménynek kell bekövetkeznie, de soha nem következhet be mindkettő. Nem szabad elfelejteni az entitástípus és az entitás-előfordulás közötti különbséget. A "B entitás" néhány előfordulására a 3., a többire a 4. esemény fog hatni. Ez a kizáró vagy logikai műveletének felel meg.



Ismétlődés (iteráció)

Az ismétlődés jelölésénél nincs hozzárendelt feltétel. A jelöléssel azt kell kifejezni, hogy egy esemény egy entitás-előfordulásra többször hathat. Egy ismétlődő esemény minden egyes előfordulásának be kell fejeződnie mielőtt a következő elkezdődhetne.

Egy banki rendszerben az ügyfelek egyszerre csak egy hitelt kaphatnak, de az életük során felvehetnek több hitelt is. Itt egy ismétlődő szerkezettel lehet jelezni azt, hogy a "Hitel felvétel" és "Hitel visszafizetés" események sorozatban többször is követhetik egymást, de egy újabb ciklus csak a visszafizetés után kezdődhet. Természetesen a fent leírt ismétlődés jelölését események csoportjaira is lehet használni, ahogy a példa mutatja.

*Párhuzamos szerkezetek*

Nem minden esemény következik be szigorú sorrendben egy entitás életében. A valós életben sokszor tudjuk, hogy bizonyos események feltétlenül bekövetkeznek, de nem tudjuk milyen sorrendben. Ezeket lehet kifejezni a párhuzamosság jelölésével.

Nagyon nem kívánatos, hogy két esemény egy adott entitás-előfordulást egyidőben érítsen. Még ha ilyen dolgot ki is fejeznénk, gyakorlatilag lehetetlen megvalósítani hagyományos rendszerekben. Ezért a párhuzamos szerkezetet csak az előre nem látható esemény-sorrendek kifejezésére lehet használni, és azzal a feltétellel, hogy a párhuzamos ágba levő események nem 'szignifikánsak', nem lényegesek az entitás attribútumai szempontjából. Az entitástörténetben a lényeges attribútumok változását követjük végig, ha vannak olyan események, amelyek az entitás elemzés pontjából nem módosítják az entitás állapotát, de nehezen helyezhetők el a szerkezetben, akkor alkalmazható ez a szerkezet. Ha az ilyen esemény halmaz mégis módosítja az entitás

állapotait, akkor szelekciók és iterációk alkalmas kombinációjával kell feloldani a konfliktust.

Bonyolultabb esetekben szükség lehet az adatmodell megváltoztatására az entitás nézetek bevezetésére, az egyik nézet vagy aspektus az egyik párhuzamos ágnak felel meg a másik pedig a másiknak. Ezért csak olyan esetekben szükséges a párhuzamos ágak megtartása, amikor az entitás nézetekre bontáshoz szükséges erőfeszítések nem igazolhatók.

Kilépés és folytatás jelölése

Az SSADM4+ változatában ennek a szerkezetnek a kezelése lényegesen különbözik a korábbi változatoktól.

Ennek a konstrukciónak az a célja, hogy az entitások élettörténetében feltárja az alternatív életutakat. Ez megfelel annak a filozófiának, hogy először az entitások 'normális' életét kell leírni, majd az alternatív esemény sorozat mintákat próbálják meg felismerni. Vagyis a normális élet egy részét mint feltételezeten bekövetkező esemény és hatás sorozatot kezeli, azaz a 'normális' esemény lefutások és az alternatív utak ugyanazzal a sorozattal kezdődik. Pl. egy hitel kérelem benyújtásakor nem lehet tudni, hogy azt elfogadják-e vagy sem.

Tehát a kilépés / folytatás egy 'feltételezett esetből' megy egy 'alternatív ágba'. Mindig először a feltételezett esettel kezdődik az entitástörténet. Amikor egy olyan esemény következik be, ami azt mutatja, hogy az alternatív eset következik be, akkor egy kilépés következik be, és az élettörténet a megfelelő alternatív ágon folytatódik.

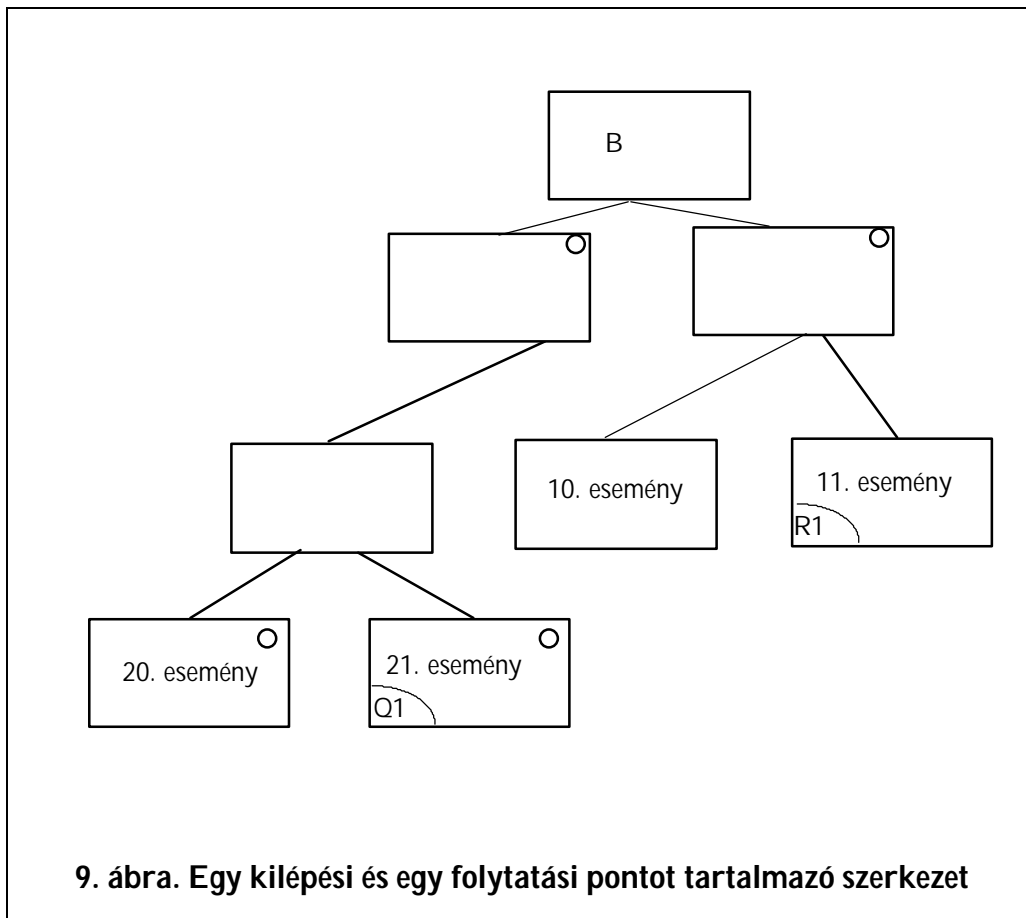
A kilépést egy "Q" (az angol "Quit" rövidítése) és mögötte egy egész szám jelzi, egy vagy több doboz baloldalán a sarokban, vagy annak közelében. A folytatást egy "R" (az angol "Resume" rövidítése) és egy egész szám jelzi egyetlen doboz baloldalán, vagy annak közelében. Így több kilépés is vezethet ugyanahhoz a folytatási ponthoz. Az összetartozókat ugyanaz a szám azonosítja. A Q-val megjelölt esemény hatás helyett az R-el megjelölt következik be.

A kilépés / folytatás csak bizonyos helyzetekben alkalmazható:

- a szelekció egyik ágából a másikba;
- egy iteráció belsejéből az entitástörténet törzsrészébe;
- az entitástörténet bármely pontjáról egy az ábrán kívüli struktúra dobozba, amely egy vagy több esemény hatását reprezentálja, és az entitás életének tetszőleges pontján bekövetkezhethet⁸.

⁸ Ez az "élet" esetlegességét kifejező konstrukció.

Általában arra kell törekedni, hogy a kilépés és folytatás alkalmazását, ha az csak lehetséges, kerüljük el.



Műveletek

Az entitások élettörténeti ábráin a műveletek a feldolgozási folyamat elemi egységeit jelölik, amelyek kombinációi alkotják a hatásokat.

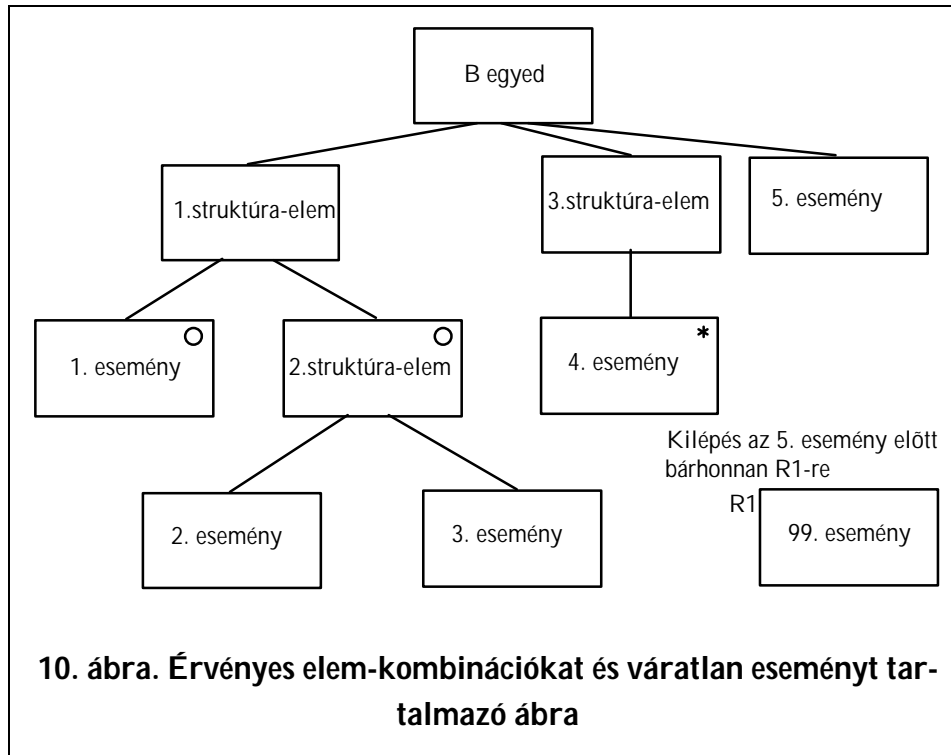
Az elemek kombinációi

Az elemek kombinációjával ki kell fejezni egy entitás életében bekövetkező minden lehetséges eseményt, a kapcsolódó hatásokat és a fontosabb műveleteket.

A hatásoknak mindig legalsó szinten kell megjelenniük, legfeljebb a műveleteket jelölő elemek lehetnek alattuk.

A jelölés elemeit struktúra-elemekkel kell összefogni azért, hogy a különböző típusú elemek ne keveredjenek azonos szinten.

A kilépést és folytatást fel lehet használni az előre nem látható vagy katasztrofális események jelzésére. A klasszikus esete ennek az entitás által jelölt személy halála. Ilyenkor egy ábrán kívüli szerkezetet kell létrehozni, amelyre a kilépés történik, és meg kell határozni az ábrának azt a részét, ahonnan ezt el lehet érni és azt a részt ahová vissza lehet térni az ábra törzsrészébe. Ez a különálló szerkezet lehet egy esemény, vagy egy eseményekből álló összefüggő szerkezet.



Köztes struktúra-elemek elnevezése

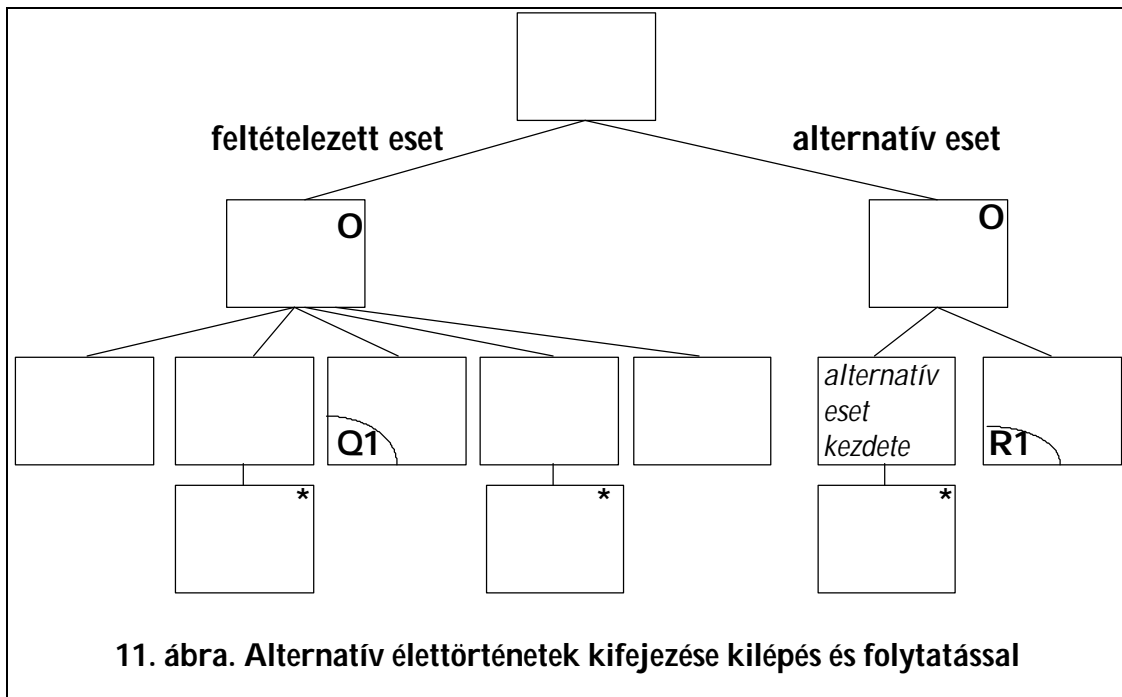
Egy struktúra-elemet értelmesen el lehet nevezni arról az időszakról, amely az elem alá sorolt eseményekre vonatkozik.

2.4.2.2 Eseményhatás-ábra és a lekérdezési út

Mindkét technika a fogalmi folyamat modellezés tartozik a rendszerfejlesztési alpmintán belül.

Az eseményhatás-ábra azt ábrázolja, ahogy egy esemény hatásai egymással összefüggenek, és megmutatja a módosítás végrehajtásához szükséges olvasási útvonalat az igényelt rendszer logikai adatmodelljén mint logikai adatbázison. A konstrukciók hasonlóak az entitás-élettörténetekben használtakhoz, de más módon vannak összekötve. A lekérdezési út pedig a logikai adatmodellen az olvasási útvonalat mutatja, a lekérdezés eredményének összeszerkesztéséhez szükséges adat hozzáféréseket és olvasási igényeket.

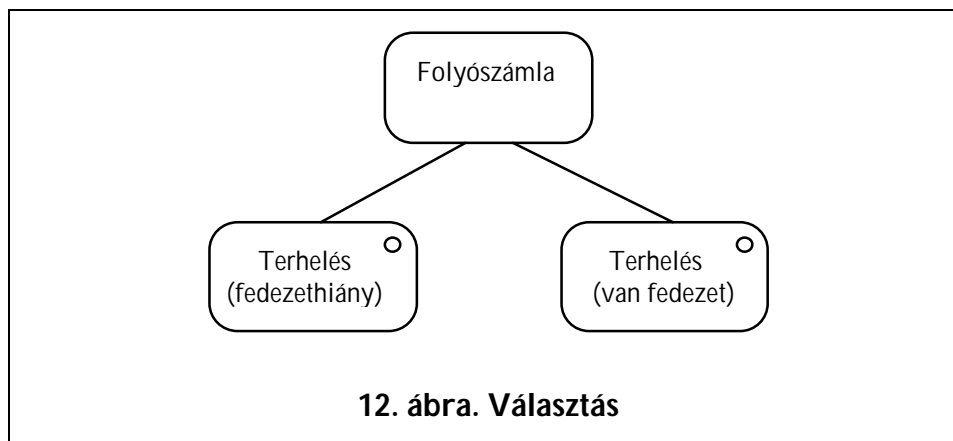
Egy eseményhatás-ábrán szerepelnie kell címként az ábrázolt esemény nevének. A hatásokat lekerekített sarkú dobozok jelölik az ábrán. Ahol az esemény egyetlen entitás-előfordulást érint és egyetlen módon, ott a hatás doboza az entitás nevét tartalmazza. Ezenkívül jelölni kell a belépési pontot, azaz azokat az adatelemeket, amelyeken keresztül az esemény vagy lekérdezés megkezdí működését a logikai adatmodellben; ezen a nyílra fel kell tüntetni azoknak az adatelemeknek a nevét, amelyek ezt az eseményt / lekérdezést kezdeményezik.



Az ábrákon kétféle szerkezet jelenhet meg, a választás és az ismétlődés.

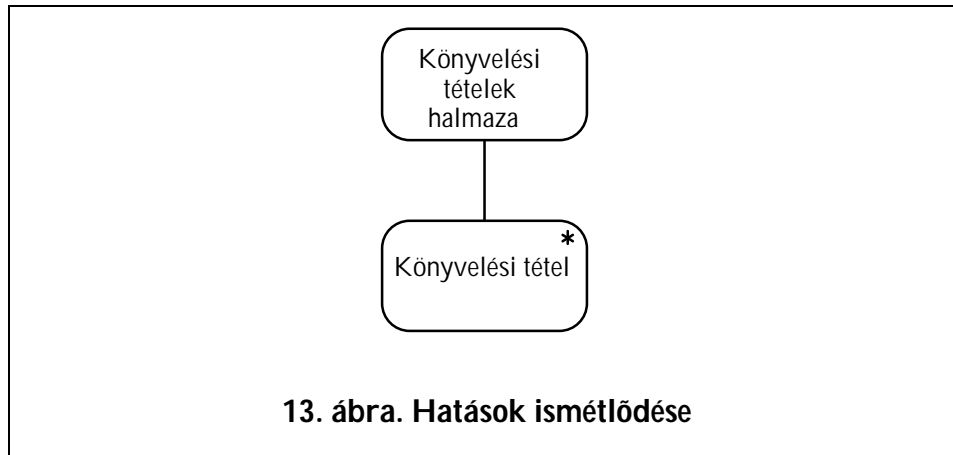
Választás

A választás azt jelöli, hogy egy esemény két vagy több, egymást kizáró módon hat egy entitásra. A következő ábrarészlet azt jelöli, hogy a "Terhelés" nevű esemény a "Folyószámla" nevű entitásra kétféleképpen hat, attól függően, hogy van-e fedezet, vagy nincs.



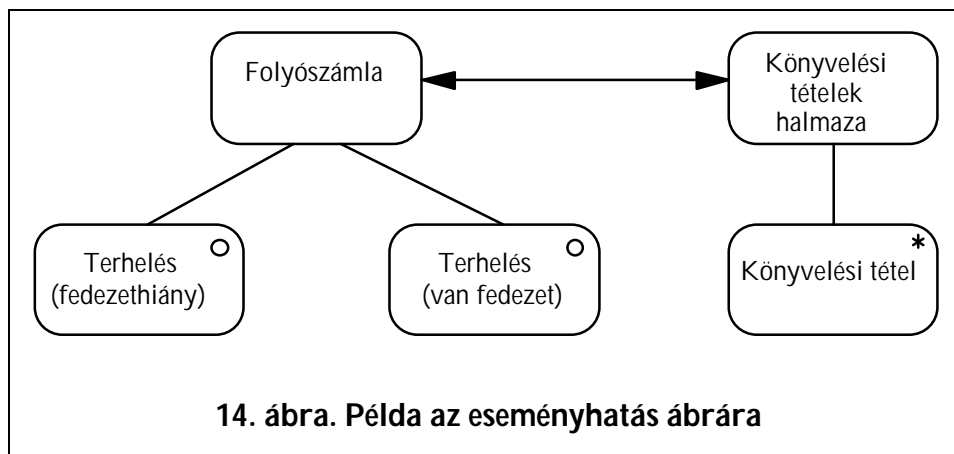
Ismétlődés

Ha egy esemény egynél több entitás-előfordulást érint, akkor egy ismétlődési szerkezetre van szükség. A következő ábra azt fejezi ki, hogy a "Terhelés" nevű esemény a "Könyvelési tétel" entitás több előfordulására hat.



Egy-egy megfelelés

Egy kétirányú nyíl jelzi a hatások közötti egy-egy megfeleltetést. Egy-egy megfelelés létezik akkor, ha egy esemény minden bekövetkezése esetén, az esemény "A" hatásának egy előfordulásához a "B" hatás egy előfordulása tartozik. A következő ábra az előző két részletből áll össze és azt fejezi ki, hogy a "Terhelés" nevű esemény egy folyószámlára két egymást kizáró módon hathat, és minden egyes ilyen hatás esetén a



"Könyvelési tételek halmaza" is érintve van (azaz több "Könyvelési tétel").

A megfelelést kifejező, kölcsönhatást jelző nyilat bármely két doboz közé be lehet húzni, kivéve:

- két ismétlődő elemként megjelölt alkotórész közé (csillaggal jelölik);
- a választás, szelekció két eleme közé (egy 'o' jelöli);

- a választás egyik eleme és egy ismétlődő elem közé.

2.5 Az entitás viselkedés modellezés lépései

2.5.1 Entitás-elérési mátrix létrehozása

Ez nem kötelező lépés és termék, de eredményesen használható kiindulási alap az élettörténeti ábrák rajzolásához. A funkció-meghatározás kezdeti eseményeit és az igényelt rendszer logikai adatmodelljét felhasználva egy entitás-elérési mátrixot kell létrehozni.

2.5.2 Kezdeti entitás-élettörténetek rajzolása (alulról-felfelé)

A "Feldolgozási folyamatok meghatározása" során a jelenlegi rendszer logikai adatmodelljének minden entitásához egy kezdeti entitás-történeti ábrát kell rajzolni.

Itt a logikai adatszerkezetben alulról felfelé kell haladni, először az alentitások életeit elemezve. Így a főentitás életét jobban meg lehet érteni, mintha elszigetelten vizsgáltuk volna. Segíthet, ha az alentitások és a hozzájuk tartozó főentitás életét párhuzamosan vizsgáljuk.

Az entitások életének és a közöttük lévő kapcsolatoknak a megértésében segíthetnek az entitásleírások, attribútum-leírások és kapcsolatleírások. A következő sorrendet érdemes figyelembe venni:

- Az entitás halála előtti, az entitás születését okozó események azonosítása. Több eseménytípus is lehet ilyen. A születési esemény létrehozza a rendszer számára az entitás elsődleges kulcsát.
- Az entitás alapvető életének változásait okozó események azonosítása. Ezeknek az eseményeknek a sorrendjét el kell dönteni, figyelembe véve az ismétlődéseket. Az entitás fontosabb állapotait módosító események hatásait az ábrán balról jobbra haladva fel kell venni, ezek közé kerülnek be szétszórva az egyszerűbb aktualizálást végző esemény hatások.
- Jackson szerkezet kialakítása a sorrendiségi, választási, ismétlődési és párhuzamossági alkotóelemeket felhasználva. (Ezt könnyebb lehet alulról felfelé haladva végezni, azaz először meghatározni a komponenseket, majd ezeket struktúra-dobozokkal összekötve beépíteni a teljes szerkezetbe.) Ekkor kell meghatározni azokat az eseményeket, amelyek esetleg többféleképpen befolyásolják az entitás életét, vagyis az entitás szerepek és hatás-minősítők meghatározását.
- A halálozási és / vagy törlési események felvétele.
- Ha entitás-elérési mátrix létezik, akkor ellenőrizni kell, hogy az entitáshoz bejelölt összes eseményt figyelembe vették-e.
- További eseményeket lehet találni a következő kérdések feltételével:

- Minden attribútum létrejön-e akkor, amikor az entitás születik? Ha nem, akkor milyen események hozzák őket létre?
- Hogyan módosulnak illetve szűnnek meg az egyes attribútumok?
- Mi okozza az entitás kapcsolatainak a megváltozását a főentitásai illetve alentitásai felé?
- Mi okozza a nem kötelező (opcionálisan) kapcsolatok születését/halálát?
- Mik a nyilvánvalóan fontos mérföldköveket alkotó események egy entitás életében?

Célszerű műveleteket rendelni az ábrákon azokhoz az eseményekhez, amelyek stabilnak tekinthetők, pl. a születésekhez, majd az ábra többi részéhez is meg kell határozni a műveleteket.

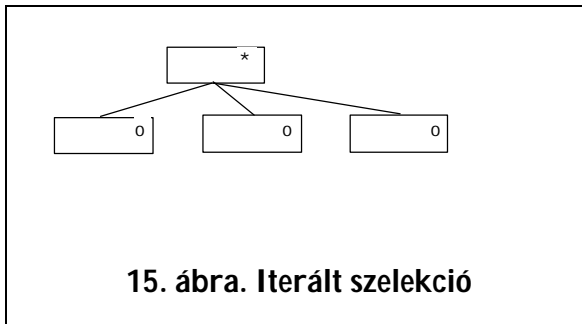
Amint az eseményeket már azonosították az eseményhatás-ábrák készítését el lehet kezdeni. Akkor kell ezeket továbbfejleszteni, ha egy eseményhez kapcsolódva további hatások kerülnek napvilágra a szóban forgó entításokra vonatkozólag.

A normális élet, az entitás alap viselkedését meghatározó elemek megtalálása és felvázolása után alulról felfelé a következő dolgokat kell megvizsgálni:

- A párhuzamos entitás nézetek szétválasztása. Ha az események által kifejtett hatások között valamilyen összeütközés van, nem lehet elrendezni az alapkoncepciókkal valamilyen ésszerű sorrendbe, ekkor alkalmazható a párhuzamos élet, ha ez sem megoldás akkor különálló entitás nézetek (aspektusok) felvétele a logikai adatmodellbe segíthet⁹. Az attribútumokat diszjunkt módon fel kell osztani az entitás nézetek között, majd mindegyik nézetre hozzák létre az entitástörténetek.
- A kölcsönösen kizáró esemény hatás kombinációkat kell vizsgálni. Az alternatív esetek szelekcióba rendezése, és a kilépés / folytatás használatával az alapesettől eltérő alternatív élettörténetek ábrázolása. Fő- / altípus esetében készüljön két entitás történet, az egyik a főtípusra, a másik pedig legyen egy olyan élettörténet, amelyik egy magas szintű szelekcióban szétválasztja az élettörténeteket résztípusokként.
- Elemezzék az iteráló, ciklikus esemény hatásokat. Vizsgálják meg az iterált szelekciókon belüli esemény sorozatot; ez egy nagyon eredményes kifejező eszköz fontos állapotokat módosító de mégis esetlegesen bekövetkező hatások érzékeltetésére. Ha van olyan esemény a szelekció elemei között, amely valójában nem illeszkedik ebbe az esemény hatás sorozatba, akkor emeljük ki ezt az esemény hatást egy párhuzamos életbe. Vizsgálják meg az entitás alentitásának életét is. Döntsék el, vajon az alentitás születését és halálát be kell-e illeszteni a főentitás életébe, akár azért, hogy a főentitás életében bizonyos kényszerfelteteleket fogalmazzanak meg, akár azért mert a főentitás attribútumainak a változásait idézheti elő az alentitás születése vagy halála. Az alentitás születésének beillesztése való-

⁹ Ez egy fajta résztípusa az adott entitás típusnak.

színűleg egy iterált esemény sorozat, szekvencia formájában jelenik meg a főentitás életének bizonyos pontján (iteráció az egy-sok kapcsolat következménye).



- Az iteráció befejeződésének feltétele. Meg kell határozni a ciklus befejeződésének feltételeit, akár ellenőrzött, előírt formában történik, akár véletlenszerűen. Lesznek olyan entitások, amelyeknek meg kell halniuk az életük bizonyos pontján, a felhasználó által irányított módon. Mások az életük

tetszőleges pontján halhatnak meg; a felhasználó nem tud bizonyos szervezeti, külső eseményeket kézben tartani, és ezek a szervezeti események kiváltak informatikai eseményeket, amelyek ezt az entitás halált, törlést előidézik.

2.5.3 Entitás-élettörténetek ellenőrzése (felülről-lefelé)

Ez is a "Feldolgozási folyamatok meghatározása" lépés során történik. A kiindulásként használt élettörténeti ábrák vizsgálatánál fontos azt kideríteni, hogy az adott entitás életét befolyásolják-e egy másik entitásra gyakorolt esemény hatások. Ha egy entitás életére ilyen befolyás történik, akkor azt az ábrán is jelezni kell.

A logikai adatszerkezeten felülről lefelé haladva, az élettörténetek közötti kapcsolatokat fel kell tárni és a kivételes hatásokat (a normális menettől eltérőket) fel kell venni. Nagyon fontos a felülről lefelé haladás az összes főentitás / alentitás kapcsolat figyelembevételé miatt. A következőket kell felmérni:

2.5.3.1 Véletlen események, extra entitás halált okozó események

A kezdeti élettörténetek felülvizsgálata során az elemző olyan eseményeket próbál azonosítani, amelyek eltérnek az eddig alap viselkedéstől. Ilyenkor olyan eseményeket lehet azonosítani, amelyek az entitás életének (vagy élete egy szakaszának) során bármikor bekövetkezhetnek. Az ilyen eseményekről tesszük fel, hogy "véletlenszerűen" következnek be. Meg kell keresni az eddig esetleg fel nem ismert entitás halálát okozó eseményeket.

Ha az entitás alap élete során próbálnánk meg kifejezni az összes olyan lehetséges esetet, amikor egy véletlen esemény bekövetkezhet, az ábra kezelhetlenné válna.

A véletlen eseményeket az élettörténetben a kilépés és folytatás egy speciális formájával ábrázoljuk. Minden véletlen eseményt egy olyan dobozba teszünk, amely nem kapcsolódik az általános élet szerkezetéhez. A folytatás jelzését a véletlen esemény elé tesszük ki. Ha a folytatáshoz tartozó kilépést nagyon sok helyen, vagy mindenütt kellene jelezni az ábrán, akkor az ábra aljára egy feliratot kell elhelyezni, "Kilépés bárholonnan Rn-be" szöveggel, ahol Rn a véletlen eseményt jelzi. (Ha a véletlen esemény az élet egy részében következhet csak be, akkor a megfelelő részt le kell írni a szövegben).

A véletlen események, természetüknél fogva, vagy bekövetkeznek, vagy nem, egy entitás-előfordulás élete során. Ha egy véletlen eseménynek mindenképpen be kell következnie, akkor az már nem véletlen esemény, és így be kell venni az általános életet leíró ábrába.

Ha a véletlen esemény bekövetkezte után szükség van az általános életbe való visszatérésre, akkor a kilépés és folytatás jelölésmódját lehet újra használni.

A kilépés és folytatást itt is az előírásoknak megfelelően kell használni.

2.5.3.2 Váratlan, abnormális halál és törlési események

A halál, törlési események vizsgálatával az entitások közötti kölcsönhatásokat lehet felismerni, különösen a főentitásból és alentitásból álló párosok esetében. A következő helyzeteket fordulhatnak elő:

I. A főentitásbeli előfordulás törlése az alentitásbeli előfordulást törli ('kaszád', 'dominó hatás' halálozás):

A. Amikor az alentitás felől a kapcsolat kötelező, akkor a főentitás halála az aktív alentitások halálát is okozza. Ilyenkor a főentitás halálát okozó eseményt fel kell venni az alentitás élettörténetébe mint halált okozó eseményt.

- KÖZÖS:

az alentitás halála akkor következik be, amikor a főentitásé; az alentitásnak nincs külön halála

- alternatív:

az alentitás halála a főentitás halálakor következik be, vagy saját halálakor

KORAI:

az alentitás élekciklusa olyan sorozatot tartalmaz, amelyet a főentitás halála több ponton is befejezhet

I. A főentitás előfordulása nem törölhető addig, amíg az összes alentitása nem törölődött ('korlátozott halál').

A. Vannak olyan esetek, amikor a főentitás halála csak akkor megengedett, ha az összes alentitás példánya már meghalt. Ilyenkor a főentitás élettörténeti ábrájára fel kell venni az utolsó alentitás kitörlésének eseményét, illetve esetleg az alentitás entitástörténeti ábrájára fel lehet venni az alentitás logikai törlése után a főentitás törlését. Egy másik alternatíva lehet az, ha egy olyan műveletet helyezünk el a főentitás halálát jelölő esemény alá, amely leellenőrzi annak az előfeltételnek a teljesülését, hogy vajon az ösz-

szes alentitás példány kitörlése, inaktiválása megtörtént-e, mert különben addig ennek az eseménynek hatása nem folytatódhat.

- ELLENŐRZÖTT:

Ahol az alentitás(ok) nem törölődnek ki a főentitás halála előtt, a főentitás halála bekerül az alegeyed életciklusába

Ahol az alentitás(ok) kitörlődnek a főentitás halála előtt, a főentitás egy "hiba" ("fail") műveletet fog tartalmazni

- BETERVEZETT:

A főentitás olyan állapotba kerül, hogy automatikusan eltávolítódik utolsó alentitásának halálakor

- I. A főentitás halála az alentítások főentitás példányának a cseréjéhez vezet

- A. Amikor a főentitás példány halála az alentításhoz vezető kapcsolat megszakítását jelenti ("cut / leválasztás") akkor a főentitás halálát az alentitás életciklusának fő részében (közepén) egy "leválasztás/csere" ("cut / swap") művelettel szemléltetjük, mivel ekkor az alentitás az életét még folytatja.

- II. A főentitás halála nincs hatással az alentításra

- A. Ilyenkor nincs egymásra hatás az entitások között. Meg kell viszont vizsgálni a két entitás közötti kapcsolatot. Ha a kapcsolat kötelező, akkor a főentitás törlése esetén az összes alentítást át kell kötni egy másik főentításhoz. Ha mégis létezik alentitás a főentitás nélkül, akkor a kapcsolatot kell megváltoztatni nem kötelezővé.

- III. Az alentítások váratlan halálát fel kell tüntetni a főentitás élettörténetén is.

- A. Minden olyan entitásra, amelyet több főentitás is birtokolhat meg kell vizsgálni az alentitás születésének és halálának a körülményeit, és ha szükséges ezeknek a következményét érzékeltetni kell a főentitás élettörténetében.

- IV. A felettes-események (szuper események) meghatározása

- A. A különböző entitástörténetekben meg kell keresni a hasonló esemény hatás kombinációkat. Amikor két vagy több esemény hatása pontosan egyforma az entitás életciklusának ugyanazon a pontján, akkor ezeket az eseményeket és hatásait össze lehet gyűjteni egy felettes-esemény alá. Amikor az esemény hatásoknak ugyanez a kombinációja más entitástörténetekben felbukkan akkor a felettes-esemény nevét lehet használni ezeknek a hatásoknak a megjelölésére, természetesen feltéve, hogy az esemény hatások pontosan egyformák. A dominó effektussal bekövetkező halált kiváltó eseményeknél a legvalószínűbb a felettes események felismerése.

- V. A törlési stratégia meghatározása.
 - A. Ha egy entitás példány megőrzésére nincs szükség statisztikai és visszamenőleges történeti elemzések céljára, akkor az entitás halála és a végleges törlés egybeesik, az állapotjelzőjét ekkor nullára lehet állítani. Ha viszont egy ideig még őrizni kívánják a rendszerben akkor a végleges törlés feltételeit meg kell határozni (ez gyakran a megfelelő főentitás példány kitörlésekor következik be.) Esetleg a halál és a végleges törlés közötti eseményeket is specifikálni kell az entitástörténeti ábrán.
- VI. A korábbi állapot visszaállítása.
 - A. Ekkor az esemény hatására az entitás példány életének korábbi szakaszába tér vissza. Pl. egy már lezárt folyószámla újra megnyitása, vagy egy törölt foglalás folytatása.
- VII. Nem módosító hatású események
 - A. A nem módosító hatású események lehetnek ellenőrzések illetve lekérdezések. Az ellenőrzéseket (az entitás állapotának vagy más attribútumainak ellenőrzése) itt kell felmérni és bevenni az élettörténetbe. Az események lekérdező hatásait később kell felvenni, a eseményhatás-ábrákra.

2.5.4 Műveletek hozzáadása

Szintén a "Feldolgozási folyamatok meghatározása" lépés során kell az egyes fontosabb műveleteket meghatározni. Minden élettörténeti ábrához fel kell venni egy művelet listát, megszámozott műveletekkel. Az élettörténeti ábrán a fontosabb műveleteket fel kell tüntetni a hatásokat reprezentáló dobozok alatt. Ezek a műveletek az entitás attribútumainak aktualizálását és kapcsolatrendszerének a karbantartását végzik.

- I. Le kell ellenőrizni az attribútumok aktualizálását.
 - A. Meg kell vizsgálni, hogy az entitás mindegyik attribútumára létezik legalább egy esemény hatás alatti művelet, mely biztosítja az attribútum aktuális értékének a karbantartását. Ha szükséges a művelet halmazt az attribútumok értékének helyettesítését, módosítását végző műveletekkel kell bővíteni, esetleg további esemény hatásokat kell az ábrára fel venni, gyakran a szelekciók alatt;
- II. Le kell ellenőrizni a kapcsolatok aktualizálását.
 - A. Meg kell vizsgálni, hogy a fő - és alentitás közötti kapcsolat mindkét végét karbantartó műveletek megvannak-e az ábrán, általában az alentitás élettörténetében jelennek meg ezek a műveletek. Ha szükséges bővítsük a művelet listát a kapcsolat aktualizáló műveletekkel.

2.5.5 Állapotjelzők hozzáadása

Az entitások élettörténetében bekövetkező hatások egy másik kifejezési módja az állapotjelző.

Az állapotjelző egy további attribútumot jelent minden entításban (példányban), amit az aktuális állapot feljegyzésére lehet használni. Ezt az állapotértéket a későbbiek során fel lehet használni a folyamatok lefutása helyességének ellenőrzésére (pl. egy adott értéknél egy adott művelet nem végezhető el). Ha az élettörténeti ábra tartalmaz párhuzamosságot, akkor több állapotjelzőt is lehet használni.

- I. Ki kell egészíteni az ábrát az állapotjelzőkkel.
 - A. Az érvényes 'megelőző' és 'végállapotokat' el kell helyezni az esemény hatások alatt. Ha lehetséges végezzünk állapotjelző optimalizálást: a szelekciók és az iterációk esetében. Amikor egy entitásnak csak két állapota van ('null', '1'), akkor nincs szükség ezek feltüntetésére az ábrán.

2.6 Kapcsolat más technikákkal és a strukturális modellel

2.6.1 Kapcsolat más technikákkal

Funkciómeghatározás

A funkció-meghatározás azonosítja az eseményeket és módosító funkciókba csoportosítja őket, valamint a lekérdező funkciókat is. Ezeket lehet felhasználni az entitás-elérési mátrix kiindulópontjaként. Ha egy lekérdező funkció hatással van egy entitás életére, akkor a funkcióleírásban módosító jellegűre kell változtatni a besorolást. A funkció-meghatározási technika nem határozza meg a adatfeldolgozás részleteit. Az entitás viselkedés modellezés határozza meg az igényelt rendszer funkcióleírásaihoz tartozó adatfeldolgozási folyamatokat.

Egy eseményt több funkció indítását is kezdeményezheti és így szerepelhet több funkcióleírásban is. Minden funkcióra az elemzőnek meg kell vizsgálnia, hogy az eseményt alkotó attribútumok szerepelnek-e a bemenetek között, illetve a funkció elő tudja-e állítani belőlük.

A felhasználóval történt megbeszélés után az entitás viselkedés modellezés kimenetét fel kell használni a funkcióleírások módosítására:

- új eseményeket lehet hozzávenni a létező funkcióleírásokhoz;
- új funkciókra vonatkozó igényeket lehet azonosítani;
- lekérdező funkcióról módosító funkcióra lehet változtatni funkcióleírásokat (Ha egy lekérdező funkcióról az ELH felépítése során kiderült, hogy megváltoztatja az entitás állapotát, akkor módosító funkcióvá kell tenni.);
- események gyakoriságára vonatkozó információkat lehet felvenni az funkcióleírásokba;
- események leírását lehet bevinni a funkcióleírások leíró részébe.

Ellenőrizni kell, hogy minden esemény hozzá van-e rendelve a megfelelő funkcióhoz. A legtöbbször ez egy az egyhez hozzárendelést jelent, de ahol bonyolultabb kapcsolatok vannak funkciók és események között, ott az ellenőrzést segítheti egy funkció/esemény mátrix használata.

Logikai adatmodellezés

A logikai adatmodellezés hozza létre a logikai adatszerkezetet, entitásleírásokat, attribútum-leírásokat és kapcsolat-leírásokat. Mindez szükséges bemenete az entitástörténeti elemzésnek. A logikai adatmodell tartalmazza azokat az entitásokat, amelyeknek az életét meg kell vizsgálni. Fel lehet használni a kezdeti entitás-elérési mátrix felállításában. Ezzel együtt, az entitástörténeti elemzés nagy része az entitások közötti kapcsolatok felderítésére, egymásra hatásának a leírására használható, olyan szervezeti és működési szabályokat tud megragadni, amelyeket a logikai adatmodell képtelen rögzíteni.

Az entitástörténeti technikában a részletes adatokra vonatkozó elemzés nagyban segíti a fejlesztőket a rendszer entitásainak jobb megértésében. Szükséges lehet ennek kifejezése a logikai adatmodellben is:

- újonnan azonosított entitásokat lehet felvenni;
- entitásokat lehet megszüntetni összevonás miatt;
- kapcsolatokat és/vagy attribútumokat lehet megváltoztatni.

Az entitásokhoz tartozó állapotokat jelző értékeket és jelentésüket is fel kell venni az entitásleírásokba mint az entitás attribútumait.

A logikai adatszerkezetet fel lehet még használni esemény hatások közötti kapcsolatok, kölcsönhatások felderítésére is az eseményhatások elemzésénél.

Fogalmi folyamat modellezés

Az eseményhatás-ábrákat és a lekérdezési-utakat használja fel kiindulópontként a logikai adatfeldolgozó folyamatok tervezése. Módosító feldolgozási modelleket kell létrehozni minden eseményhatás-ábrából, amit a fizikai tervezés bemeneteként lehet majd felhasználni. Az entitás-élettörténetek ábrázolják az igényelt feldolgozási folyamatok sorrendjét és az azonosított műveleteket, így szintén bemeneteként szolgálnak a logikai adatfeldolgozó folyamatok tervezéséhez. A lekérdező utakból készülnek a lekérdezést végző folyamatok.

2.6.2 Eseményhatás-ábrák létrehozása

A "Feldolgozási folyamatok meghatározása" lépés során kell létrehozni a eseményhatás-ábrákat is, mivel ez az entitás-élettörténetek érvényességének ellenőrzésének fontos technikája.

Minden eseményhez, amelyet az élettörténeti elemzés során azonosítottak, egy eseményhatás-ábrát kell rajzolni. Ezt akkor lehet elkezdni, amikor az események már ismerté váltak. Egy esemény összes hatását fel kell venni. Az eseményt kezdeményező adatokat meg kell határozni. Általában ez az entitás kulcsa, ami a belépési pont a logikai adatszerkezetbe, kiegészítve néhány módosítandó attribútummal.

2.6.3 Funkcióleírások módosítása

A "Feldolgozási folyamatok meghatározása" lépés során, az entitás-esemény modellezés eredményét vissza kell vezetni a funkció-leírásokba is.

2.6.4 Szervezeti tevékenység modellezése

A szervezeti tevékenység modellezése tartalmazza a szervezeti események meghatározását, ebből kiindulva lehet felismerni azokat az eseményeket, amelyek a fogalmi modell folyamatainak indítását fogják kezdeményezni.

2.7 Műveletek

Az esemény hatások dokumentálása után az egyes hatásokhoz tartozó entításra vonatkozó műveleteket kell ábrázolni.

Egy művelet a logikai feldolgozás olyan egyedileg megkülönböztetett egysége, amely önmagában, vagy más műveletekkel együtt, egy esemény hatását alkotja.

A műveletek hasznosak lehetnek az élettörténeti elemzés által figyelmen kívül hagyott események felismerésében, például olyan kérdések vizsgálatakor, mint:

- Mikor történik ennek a számításnak az elvégzése?
- Mikor aktualizálják ennek az attribútumnak az értékét?
- Mikor törlik ennek az attribútumnak az értékét?

Minden entitástörténeti ábrára fel kell venni a hatásokhoz tartozó fontosabb műveletek listáját. Minden műveletet egyenként meg kell számozni, ezek az azonosítók az egész rendszerben egyedileg azonosítják a műveleteket (noha az egyes ábrákon belül a helyi azonosítók használata megengedett¹⁰). A műveletek számát kis dobozok tartalmazzák az ábrán, amelyek a megfelelő esemény hatás alá vannak elhelyezve, abban a sorrendben, ahogy végre kell hajtani őket. Egy hatás egynél több művelet eredménye is lehet. Egy hatásnak lehet, hogy nincs művelet az itt megengedett művelet halmazban (pl. az esemény hatás egyszerűen csak az entitás állapotát állítja). Az állapotjelző értékét állító műveleteket a későbbi logikai adatfeldolgozások tervezésénél kell figyelembe venni.

¹⁰ Ez az egész rendszerre kiterjedő azonosító rendszer természetesen csak számítógépes támogatással tartható karban, egy megfelelő tulajdonságú CASE eszköz alkalmazásával. Különben valamilyen ésszerű kompromisszumot kell keresni.

A következő művelet típusok alkalmazhatók az entitástörténetben, ezt azonban csak egy kiinduló, törzs művelet halmaznak kell tekinteni. Az adott megvalósítási környezetre vagy CASE eszközre kell alkotó módon testre szabni az itteni művelet típusokat, illetve az eszközben rendelkezésre álló műveletekhez illeszteni.

Az entitástörténeti elemzésben megengedett műveletek típusai:

| Művelet | Magyarázat |
|---|---|
| <attribútum> beállítása (set <attribute>) | Az attribútum értékének beállítása az esemény által hozott bemeneti értékre. |
| <attribútum> beállítása <kifejezés> értékre set <attribute> using <expression> | Az attribútum értékének beállítása a kifejezés kiértékelésének eredményére. A születési hatásnál akkor használandó, ha az eltárolt értékek különböznek az esemény bemeneteként kapott adat- elemek értékeitől. |
| <entitás> létrehozása (create <entity>) | Ez a művelet az entitás születéséhez kapcsolódik és csak a születési hatásnál érvényes. Az entitás elsődleges kulcsa és a kötelezően kitöltendő attribútumok értékének beállítása történik ekkor. Az opcionálisan értéket kapó attribútumok vagy az alapértelmezést vagy 'null' értéket kapják. A fogalmi folyamat modellezésben ez a művelet azt jelenti, hogy a folyamaton belül, helyileg ez az entitás példány létrejött, de addig háttértárolóban tárolt adatokra nincs hatással, amíg egy 'írás' ('write') műveletet ki nem adnak. |
| <Főentitás> -hoz kötés (tie to <entity>) | Az adott entitás és egy főentitása közötti kapcsolat megteremtése. |
| leválasztás <főentitás> -ról (cut from <entity>) | Az adott entitás és egy főentitása közötti kapcsolat megszüntetése. |
| <almentitás> nyerése (gain <entity>) | Az adott entitás és egy almentitása közötti kapcsolat megteremtése. |
| <almentitás> elvesztése (lose <entity>) | Az adott entitás és egy almentitása közötti kapcsolat megszüntetése. |

| Művelet | Magyarázat |
|--|--|
| <főentitás> csere (swap <entity>) | egy főentitás példányhoz tartozó kapcsolatok átkötése ugyanakkor a főentitásnak egy másik példányához |
| <folyamat> indítás (invoke <process>) | Meghívja, elindítja a <folyamat> nevű folyamatot, amelyet valahol másutt specifikáltak. Lehetőség, hogy egy felettes-eseményt, elemi folyamat specifikációt vagy közhasznú lekérdezést indítsanak el ezen a módon. |

Az SSADM nem korlátozza a használt <kifejezés> formáját.

Az entitástörténeti elemzésben a "Nyerés" és "Elvesztés" típusú műveletek csak szintaktikus ellenőrzési segédletként használandók:

- Minden főentitásbeli "Nyerés" ('Gain') művelethez kell lennie egy "Hozzákötés" ('Tie') műveletnek a megfelelő alentitásban.
- Minden főentitásbeli "Elvesztés" ('Lose') művelethez kell lennie egy "Leválasztás" ('Cut') műveletnek a megfelelő alentitásban.

Ha egy esemény hatás bekövetkezése előtt több érvényes megelőző állapot van, akkor ehhez a hatáshoz olyan feltételeket csatolni, aminek alapján el lehet dönteni, hogy melyik műveletet hajtsák végre, ha az állapotjelző értéke valamilyen konkrét értéket tartalmaz (vagy valamilyen érték tartományba esik). Ebben az esetben a műveletet egy "HA SI = <érték> ... (If SI = <value>)".

Az entitástörténeti elemzésben nem megengedett műveletek

- adatbázisbeli navigálás céljából entitás elérése, olvasása;
- törlés;
- adatérvényesítés, validálás, helyesség ellenőrzés;
- adatelemek kiírás előtti kezelése / sorba rendezése;
- entitás módosítás előtti olvasása.

2.8 Entitás-elérési mátrix

Az entitás-elérési mátrix nem kötelezően előírt termék hanem egy jól használható munkaanyag, ami segít azonosítani az események által befolyásolt entitásokat. Két egyszerű ellenőrzésre ad lehetőséget, amelyek sokat segíthetnek:

- minden entitásra legalább egy esemény hat;

- minden esemény hat legalább egy entitásra.

A mátrix felső részére kell felvenni az igényelt rendszer logikai adatszerkezetének entitásait. A funkció-meghatározás során felderített eseményeket a mátrix baloldalán kell szerepeltetni. Ezek után kapcsolatba kell hozni az entításokat az eseményekkel, amiben segíthet a logikai adattár/entitás megfeleltetés.

Az esemény entitásra gyakorolt hatásának fajtáját eldöntve a mátrixban a megfelelő helyen a következő jelzést kell tenni:

| Röv. | Angol megnevezés | Röv. | Magyar megnevezés | Magyarázat |
|------|------------------|------|--------------------------|--|
| I | INSERT | B | beszúrás | |
| M | MODIFY | M | módosítás | |
| D | DEATH | H | halál | az entitás példányok nem aktívak a rendszerben de esetleg bizonyos lekérdezések hozzájuk tudnak férni |
| B | BURIED (delete) | T | törlés | az entitás példányok végleg eltűnnek a rendszerből. |
| G | GAIN DETAIL | N | alentitás nyérése | Főentítésra vonatkozik és az alentitás entitás-történetében meg kell jelennie a főentítéshez kapcsolás műveletének ('T', 'K') |
| L: | LOSE DETAIL | V | alentitás elvesztése | Főentítésra vonatkozik és az alentitás entitás-történetében meg kell jelennie a főentítésről való leválasztás műveletének ('C', 'L') |
| T | TIE | K | főentítéshez kapcsolás | Alentítésra vonatkozik, kötelező párja a ('G', 'N'). |
| C | CUT | L | főentítésről leválasztás | Alentítésra vonatkozik kötelező párja az ('L', |

| Röv. | Angol megnevezés | Röv. | Magyar megnevezés | Magyarázat |
|------|------------------|------|---|--|
| . | | | | 'V'.) |
| X | SWAP DETAIL(S) | X | almentítások cseréje | Az adott főmentítés egyes példányaihoz kapcsolt almentítés példányokat más főmentítés példányokhoz kapcsolják, kötelező párja az ('S', 'C'). |
| S | SWAP MASTER(S) | C | főmentítások cseréje | Az adott almentítés egyes példányaihoz kapcsolt főmentítés példányokat más főmentítés példányokra cserélik ki, kötelező párja az ('X', 'X'). |
| R | READ | O | olvasás, eseményekben / lekérdezésekben | |

2.9 Eseményhatás-ábrák, lekérdezési utak

Az eseményhatás-ábra jelzi egy esemény összes hatását a rendszerbeli adatokra, a logikai adatmodellre mint logikai adatbázisra, és jelzi a hatások összefüggéseit.

Az eseményhatás-ábrákat a hatások közötti összefüggések ábrázolásra használják. A logikai adatfeldolgozások tervezésénél azokat a hatásokat, amelyek egy-egy megfelelésben vannak egymással, összevonják és az ábrát felhasználják a módosító feldolgozási modellek létrehozására.

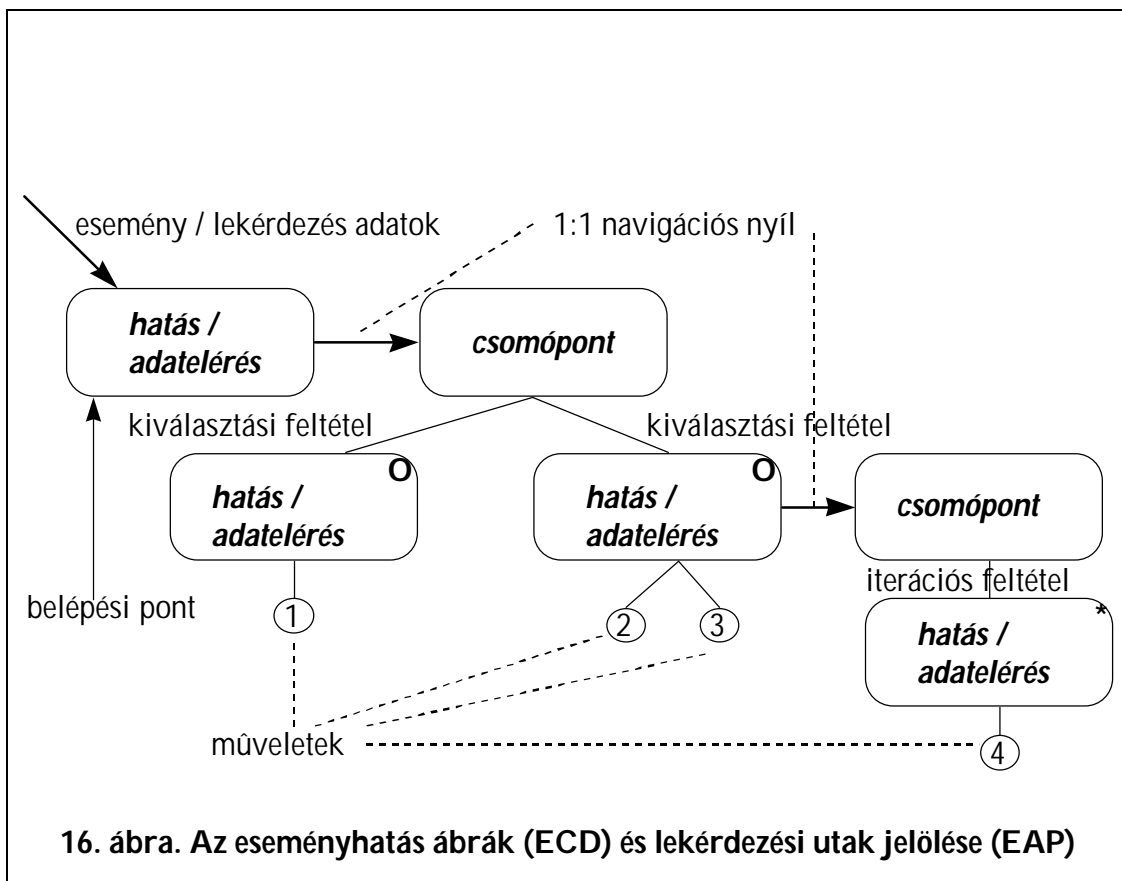
A logikai tervezésben az eseményhatás-ábrák adják a módosító, aktualizáló folyamatok adat-elérési útjait.

Az entitástörténeti ábra egy entitás nézőpontjából adja meg a kapcsolódó események (és hatásaik) sorrendjét. Az eseményhatás-ábra egy esemény nézőpontjából sorolja fel az entitásokra gyakorolt hatásokat.

A gyakorlatban a következő hét lépés során lehet az eseményhatás-ábrákat előállítani:

- Minden egyes eseményhez, amely megjelenik az entitástörténeti ábrákon, vegyünk fel minden érintett entitás és entitás nézet, jelzésére egy-egy dobozt.
- Rajzoljunk külön dobozokat az egyidejű hatások jelzésére. (entitás-szerepek).

- Vegyük fel az opcionális hatásokat, ahol több közül pontosan egy végrehajtandó hatást kell kiválasztani.
- Adjuk hozzá az ismétlődéseket a hatásokhoz, ismétlést jelző dobozok formájában.
- Egészítsük ki az ábrát az eseményt kezdeményező adatokkal, amelyek a logikai adatmodell belépési pontját jelölik ki.
- Vegyük fel a hatások közötti egy-egy megfeleléseket.
- Egészítsük ki az ábrát az ismétlődések és szelekciókra vonatkozó feltételek megfogalmazásával.
- Adjuk hozzá a műveleteket az ábrához.



A lekérdezési utak készítése:

- Határozzuk meg a lekérdezéseket;
- Azonosítsuk azokat az entitásokat, amelyeket érint a lekérdezés;
- Készítsük elő a logikai adatmodell vonatkozó részét;
- Hozzuk létre az első vázlatot a lekérdezési útra;

- Egészítsük ki az ábrát az lekérdezést kezdeményező adatokkal, amelyek a logikai adatmodell belépési pontját jelölik ki.
- Adjuk hozzá a műveleteket az ábrához s ellenőrizzük le a logikai adatmodellt.

2.9.1 A fogalmi folyamat modellezésben megengedett műveletek

Ezek a műveletek használhatók az eseményhatás ábrákon, lekérdezési utakon, a lekérdezési és aktualizálási folyamatok modellezésébe.

| Művelet | Magyarázat |
|--|---|
| Read <entitás> by key olvasd be az <entitás>-t kulcs szerint | entitás beolvasása (adatbázisból) kulcs szerint a bemenetként megadott kulcsértékek felhasználásával |
| Define set of <entitás> matching input data | entitáspéldányok olyan halmazának kijelölése, melynek elemei kielégítik a bemeneti adatokban megadott kritériumokat, ez kiegészíthető sorba rendezési utasítással is, pl. 'order by...' |
| Read next <entitás> in set | az utoljára kijelölt halmazból a következő entitáspéldány beolvasása (adatbázisból). Ezt a műveletet mindig meg kell előznie egy 'Define set' műveletnek. |
| Read next <alentitás> of <főentitás> [via <kapcsolat>] | a főentitás aktuális példányához kapcsolódó alentitások közül a soron következő beolvasása (az adatbázisból). Az opcionális 'via <kapcsolat>' kifejezésrészlet egy konkrét kapcsolat azonosítását szolgálja arra az esetre, amikor egynél több kapcsolat van a főentitás és az alentitás között. |
| Read <főentitás> of <alentitás> [via <kapcsolat>] | az alentitás aktuális példányához tartozó főentitás beolvasása (az adatbázisból) a megadott kapcsolaton keresztül. Az opcionális 'via <kapcsolat>' kifejezésrészlet egy konkrét kapcsolat azonosítását szolgálja arra az esetre, amikor egynél több kapcsolat van a főentitás és az alentitás között. |
| Invoke <közös folyamat> | az itt megadott, máshol már definiált közös folyamat elindítása (ua. mint az ELH-nál) |
| Fail if <feltétel> | a teljes folyamat leállítása (abortálása), ha az előre meghatározott feltételek fennállnak. Egy aktivált abortálás következtében természetesen a további műveletek már nem |

| Művelet | Magyarázat |
|---|--|
| | hajtódnak végre. |
| Fail if state indicator of <entitás> outside <értékkészlet> | entitás beolvasása után annak a hibának a jelzése, hogy az állapotjelző értéke kívül esik a megengedett határokon. |
| Create <entitás> | entitás létrehozása (az adatbázisba történő kiírás nélkül) (ua. mint az ELH-nál) |
| Set SI of <entitás> to '<érték>' | az entitás állapotjelzőjének beállítása kiírás előtt. Az érték lehet szöveges vagy numerikus |
| Write <entitás> | az entitás adatbázisba történő kiírása. |
| Delete <entitás> | az entitás kitörlése az adatbázisból |
| Get <adatelemek> | Az eseményt / lekérdezést kezdeményező funkciótól az adatelemek átvételének kezdeményezése, kérése |
| Output <adatelemek> | Az eseményt / lekérdezést kezdeményező funkciónak a kimenő adatelemek átadása. |

Több műveletet is ki lehet egészíteni egy 'On Error <action / tevékenység>' kifejezéssel, ha esetén végrehajtandó tevékenységek megadásával. A tevékenység általában az állapotjelző állítása, a folyamat abortálása, vagy egy iteráció befejezése és a folyamat következő részére való ugrás lehet.

2.9.2 Rajzoljunk egy-egy dobozt az esemény által befolyásolt entitások jelzésére

Az esemény által érintett entitásokat az entitástörténeti ábrákról lehet átvenni. Meg kell keresni az összes olyan entitástörténeti ábrát, amelyen az adott esemény szerepel. Minden ilyen ábra egy-egy entitást ír le, így az eseményhatás-ábrára az entitástörténeti ábrák entitásai kerülnek.

Az entitás nézeteket is fel kell tüntetni, amelyek az entitás-elérési mátrixban szerepelnek.

Az entitás fő- és altípusok kezelése a következőképpen történhet. Az eseményhatás ábrán a főtípusra gyakorolt hatásokat különülten kell megjeleníteni az altípusokra kifejtett hatásoktól. Egy téglalap reprezentálja a főtípuson előidézett hatásokat, egy másik pedig az altípusok csoportjára vonatkozókat.

2.9.3 Tüntessük fel az események összes hatását

- Minden azonosított egyidejű (szimultán) hatást külön dobozként fel kell venni (entitás-szerepek jelzik).

Az entitástörténeti ábrát lehet használni egy adott entitáshoz tartozó egyidejű hatások felismerésére. Az egyidejű hatás azt jelenti, hogy egy adott esemény egyetlen előfordulása az adott entitás egynél több előfordulását érinti, és minden előfordulást különbözőképpen. Az entitástörténeti ábrán ilyenkor az esemény hatása többször szerepel, és minden egyes helyen az esemény nevét minősíti egy entitás-szerepkör megnevezése.

Az ilyen módon összetartozó, egy entitást érintő egyidejű hatásokat az eseményhatás-ábrán be lehet keretezni, és ezt a keretet mint önálló objektumot is lehet használni (pl. a megfelelések jelzésénél).

- Vegyük be a kölcsönösen kizáró hatásokat (hatás-minősítők jelzik)

Ha egy esemény egy entitásra két vagy több egymást kizáró módon hat (az esemény különböző előfordulásaikor), akkor az összes hatást fel kell venni az entitást jelző doboz alá, a szelekció, 'kizáró vagy' választást jelezve.

2.9.4 Határozzuk meg a belépési pontot

A belépési pont valójában azt az entitást jelzi a logikai adatmodellben, amelyet az esemény elsőnek érint, és ez a hatás az esemény bemenő adatelemeit használja, ezeket is érzékeltetni kell az ábrán.

2.9.5 Adjuk hozzá a hatások közötti egy-egy megfeleléseket

A logikai adatszerkezet vizsgálatával meg lehet állapítani, hogy egy adott entitás egy-egy kapcsolatban van-e más entitásokkal az adott eseményhatás-ábrán. Ez általában akkor fordul elő, ha alentitás felől kell főentitást elérni. A következő kérdésre kell választ keresni:

- Amikor ezen entitás-előfordulások közül egy módosul, van-e olyan másik entitástípus, amelynek pontosan egy előfordulása módosul?

Itt az a cél, hogy az egy-egy megfelelések felderítésével a hatásokat csoportokba soroljuk, ami a módosító adatfeldolgozó folyamatok szerkezetének kialakításában fog segíteni.

Az azonosított egy-egy megfeleléseket nyíllal kell összekötni, ezek általában a logikai adatmodell egy kapcsolatának fognak megfelelni. Ez alól csak az a kivétel, ha a bemenő adatok között van olyan, amelyik egy vagy több entitásnak az elsődleges kulcsa, mert ekkor közvetlenül ez az entitás elérhető, például a hierarchiában feljebb elhelyezkedő főentitások szerint.

2.9.6 Adjuk hozzá az ismétlődéseket

Azokat az entitásokat, amelyeknél az adott esemény több előfordulásra is hat, meg kell jelölni és fel kell venni föléjük egy dobozt az ismétlődés jelzésére, ami az előfordulások "halmazát" nevezi meg.

- Az ismétlődő hatást a logikai adatszerkezet kapcsolatai alapján lehet azonosítani, a fő-entitás és az alentitás közti egy-sok kapcsolat alapján. Ha egy esemény egy főentitásra és alentitásra is hat, akkor valószínűleg az alentítások több előfordulására is hat.
- Ez nem feltétlenül van így minden eseménynél. Például lehet olyan adatbeviteli esemény, amely egy főentitás egy előfordulását viszi fel a hozzátartozó alentitás egyetlen előfordulásával együtt. Ekkor egy-egy megfelelést jelző nyílat kell beiktatni közéjük. Ez általában akkor fordul elő, amikor a bemenő adatok tartalmazzák az alentitás kulcsát is, vagy egyéb szűkítő feltétel jelenik meg mint pl. a 'legutolsó példányra' van szükség.

2.9.7 Egészítsük ki az ábrát az ismétlődések és szelekciókra vonatkozó feltételek megfogalmazásával

Mindegyik szelekciós és iterációs komponenshez hozzá kell rendelni a végrehajtás feltételeit.

Nem kell olyan iterációhoz feltételt rendelni, ahol az alentítások teljes halmazát fel kell dolgozni. Ugyancsak nincs szükség azoknál a szelekcióknál a kiválasztási feltétel megadására, amikor a szelekció az altípusok kezelését ábrázolja.

2.9.8 Adjuk hozzá a műveleteket az ábrához

Az entitástörténetek mutatják az esemény hatásokat és az általuk okozott műveleteket. Az entitástörténeten megjelenő hatások és az eseményhatás ábrán megjelenő hatások ugyanazok. Az egyetlen különbség az, hogy az entitástörténet ábrán az esemény nevével jelöljük míg az eseményhatás ábrán az entitás nevével ezeket a hatásokat.

Az entitástörténeti ábrákból kiolvasható a műveleteket a következő sorrendben adjuk hozzá az eseményhatás ábrához:

- entitás olvasások;
- az állapotjelző (SI, state indicator) vizsgálatának következményeként előálló hiba helyzeteket kezelő művelete. Az entitástörténeten megjelenő hatásokat 'megelőző' érvényes állapotok vizsgálata vezethet ezekhez a műveletekhez;
- az állapotjelző (SI, state indicator) értékét állító műveletek. Az entitástörténeten megjelenő olyan hatások elemzésekor keletkezik ilyen művelet, amelyek használják a 'set to SI' műveletet;
- az entitás példányok kiírása ('write');

A műveleteket a végrehajtási sorrendjükben kell hozzárendelni az ábrához. Műveletek kapcsolhatók az ábra csomópontjaihoz és hatásokat jelölő elemekhez is.

2.10 Állapotjelzők

Az entitástörténeti ábrák meghatározzák az entitásokra ható események sorrendjét. Az állapotjelzők az entitástörténeti ábra szerkezetének egy olyan leírási módját jelentik, amelyet a logikai tervezés során fognak felhasználni arra, hogy az események meghatározott sorrendjét a betartassák, bekövetkezésük érvényességét ellenőrizzék.

Az állapotjelző az entitás egy további attribútumaként jelenik meg. Amikor egy esemény bekövetkezett, akkor az állapotjelző értékét automatikusan módosítják egy új, egyedi értékre.

Egy entitástörténeti ábrán belüli állapotjelzők vizsgálatával bármikor megállapítható, hogy mi az adott entitás-előfordulás aktuális állapota, valamint az, hogy mely események fogják legközelebb módosítani az entitás-előfordulást. Az állapotjelzőkben áttételesen kifejezett érvényesítési szabályok a későbbi logikai tervezés során a fogalmi modell folyamatainak belső szerkezetébe épülnek be.

Mivel az állapotjelzők az ábra szerkezetének egy másfajta kifejezési módját adják, ezért a felvételük voltaképpen mechanikus eljárást jelent.

2.10.1 Állapotjelző jelölésmódja

Az állapotjelző jelölésmódja a "szám(ok)/szám" alakot követi, ahol:

- a "/" előtti számok az állapotjelző lehetséges értékeit jelzik az adott hatás bekövetkezése előtt (megelőző állapotok),
- a "/" utáni szám az állapotjelző értéke az adott hatás lezajlása után (beállítandó, rákövetkező állapot, vagy végállapotot).

Ezeknek a megelőző állapotoknak az ellenőrzése az, ami miatt az állapotjelzőt használni érdemes. Ha egy esemény feldolgozása során az állapotjelző értékének ellenőrzésekor kiderül, hogy az aktuális állapot nincs a felsorolt érvényes, megelőző állapotok között, akkor a hatásnak nem szabad bekövetkeznie és egy kivételkezelési folyamatot kell elindítani.

Az állapotjelző értéke csak az entitástörténeti ábrán belül értelmes, más ábrákon lévő hatásokhoz nem kapcsolódik.

Az érték, amelyre egy hatás állítja az állapotjelzőt, bármi lehet, ami egyértelműen megkülönbözteti az egyes hatások bekövetkezését. Általában a születési hatás az állapotjelzőt "1"-re állítja, minden további hatás pedig eggyel növeli ezt az értéket. Ha kifejezőbb vagy hasznosabbnak tűnik, az állapotoknak nevet is lehet adni, szöveges kifejezéssel is le lehet írni.

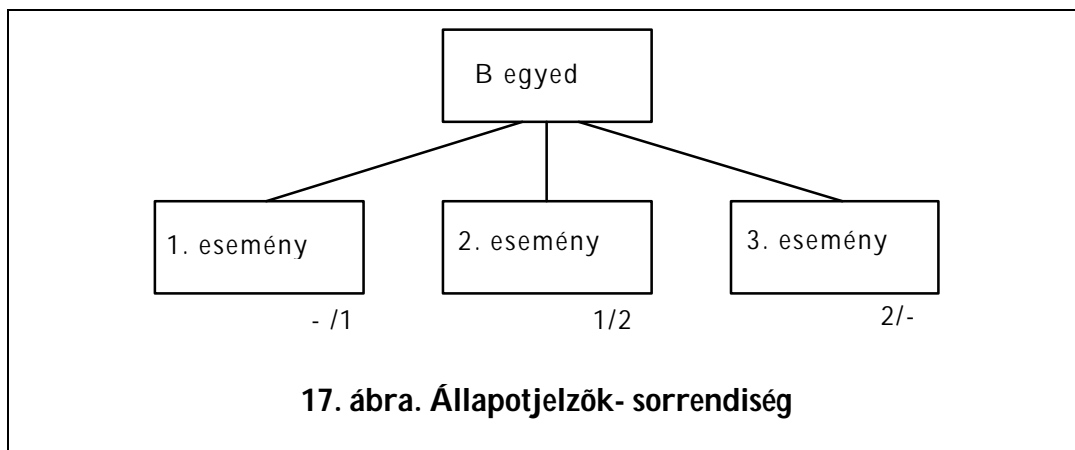
Azoknál az eseményeknél, amelyek létrehozzák az entitás-előfordulást, természetesen nem lehetnek érvényes megelőző értékek. Ilyenkor a megelőző érték az "üres", amit egy "-" jellel lehet jelölni. A születési esemény állapotjelzője tehát a "-/ szám" alakú. Ehhez hasonlóan a törlési eseményeknél nincs rákövetkező érték, amit ugyanúgy kell jelölni, azaz a "szám(ok)/-" alakban.

2.10.2 Alapszabályok az állapotjelzők felvételénél

Az állapotjelzőket két lépésben kell az ábrákra felvenni. Először az első születési eseménytől kezdődően minden hatást jelző dobozhoz egy egyedi számot kell rendelni, ami a hatás által beállítandó értéket jelöli majd. A törlési események után az üres ("-") értéket kell beállítani szám helyett. A második menetben meg kell határozni az érvényes megelőző értékeket.

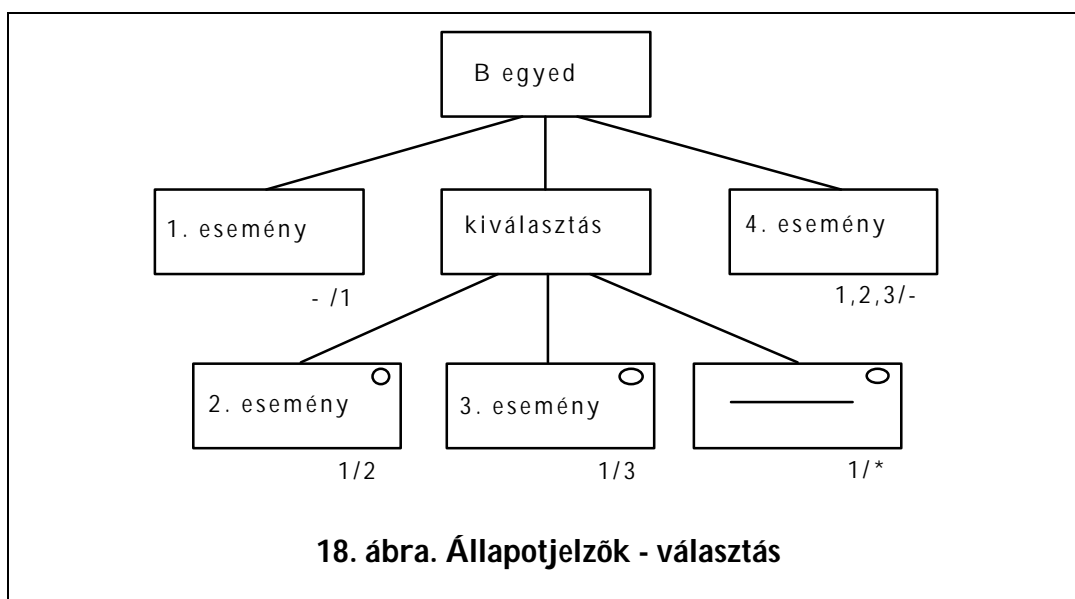
Sorrendiség

Sorrendiség esetén az egy hatás által beállított állapotjelző érték a rákövetkező hatás érvényes megelőző értéke lesz.



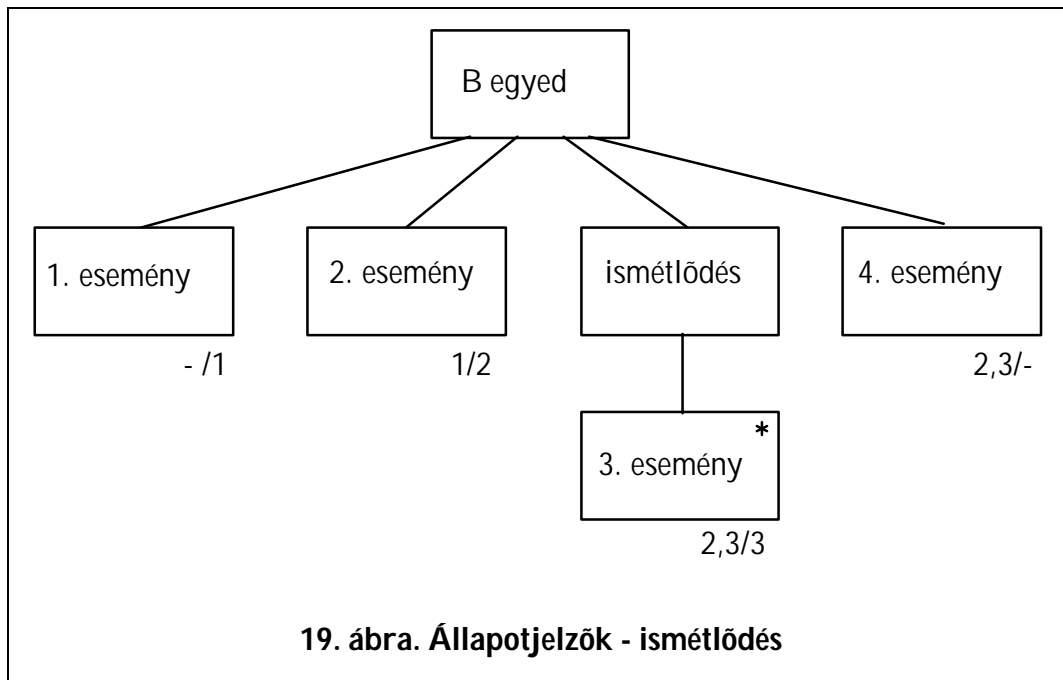
Választás

Hatások közötti választási lehetőségek esetén, minden egyes választható hatásnak ugyanazt az érvényes megelőző állapothalmazt kell feltételeznie. A választási szerkezet utáni hatás érvényes megelőző állapotai között kell lennie a megelőző választási szerkezetben lévő hatások által beállított állapotoknak. Ha a választások között az "üres" lehetőség is benne volt, akkor a választási szerkezetet megelőző állapotot is fel kell sorolni, mint érvényes megelőző állapotot.



Ismétlődés

Az ismétlődés esetén az érvényes megelőző állapotok közé fel kell venni az ismétlődő hatás által beállított állapotot is. Az ismétlődést követő hatás megelőző állapotai között kell jelezni az ismétlődő hatás megelőző állapotait is, ami az ismétlődés be nem következését is megengedi.

*Kilépés és folytatás*

Az összetartozó kilépések és folytatás esetén a kilépéssel megjelölt hatás által beállított állapotnak a folytatással jelölt hatás érvényes megelőző állapotai között kell szerepelnie.

Párhuzamos szerkezet

A párhuzamos szerkezet egyik ága lehet csak az, amelyik az elsődleges állapotjelzőt állítja, ez megállapodás szerint a szerkezet legelső ága. A további ágak hatásainak változatlanul kell hagyniuk az elsődleges állapotjelzőt. Ezt a beállított állapot száma helyett egy csillaggal lehet jelezni. Ha a további ágakban szükség van az események által beállított állapotok azonosítására, akkor másodlagos állapotjelzőket lehet felvenni minden egyes további ágon, ahol ez szükséges. Minden ilyen másodlagos állapotjelzőt ugyanúgy külön attribútumnak kell tekinteni, mint az elsődleges állapotjelzőt és ugyanazok a szabályok érvényesek rá.

2.10.3 Az állapotjelző optimalizálása

Az állapotjelzőket két egyszerű elv segítségével lehet optimalizálni:

- egy szelekció minden hatásának a végállapotát ugyanazzal az értékkel jelölhetjük;
- a megismételt, ismétlődő hatások végállapotát az ismétlődés megkezdése előtti értékkel jelölhetjük (vagyis az ismétlődés nem változtatja meg az állapotjelző értékét.)

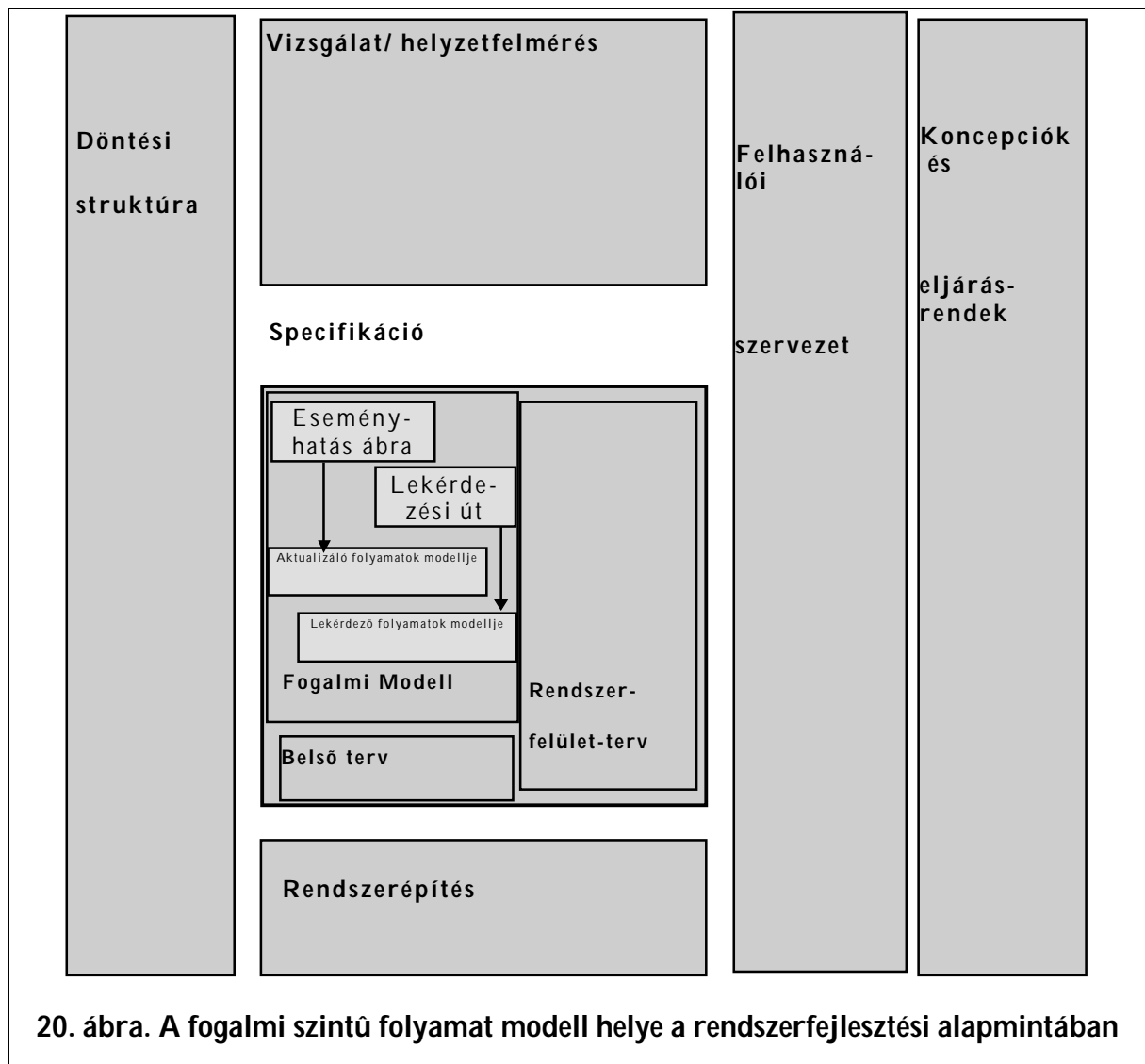
Az optimalizálás előnyei:

- az események érvényességének ellenőrzését egyszerűsíti, rövidítve a megelőző állapotok listáját;
- növeli a folyamatok újra felhasználhatóságának lehetőségét, megengedve a felesleges események felismerését;
- segíti az állapotok megnevezését, ezáltal könnyíti a felhasználókkal zajló kommunikációt.

3. Logikai rendszer specifikáció az SSADM 4+-ban (Fogalmi folyamat modell)¹¹

A 'Fogalmi modellen' manipuláló, működő folyamatok leírására a 'Fogalmi szintű folyamat modellt' vagy rövidebben a 'Fogalmi folyamat modellt' készítik el. Ez a modell definiálja a rendszer adatfeldolgozási folyamatok formájában megfogalmazott reakcióit az eseményekre és a lekérdezésekre. A fogalmi folyamat modell elkészítése során egy sor ide tartozó technikát használunk, amelyek az igényelt rendszer logikai adatmodelljét kezelő, azon operáló események, lekérdezések és logikai adatbázis műveletek adatfeldolgozási tevékenységét végzik.

A fogalmi folyamat modellezés helyét a rendszerfejlesztési alapmintában az ábra érzékelteti.



¹¹ [CCTA95], [CCTA95A], Reference Manual Part 6: Modelling from the System's Perspective, Conceptual Process Modelling, 6-59—6-90, Users Guide Part 3: Specification (Conceptual Model), Conceptual Process

3.1 Cél

A fogalmi folyamat modellezés azoknak a folyamatoknak a specifikálását és tervezését végzi el a 'Fogalmi modellen' belül, amelyek a események és lekérdezések hatására kezdeményeződnek és a 'Logikai adatmodell' elemeit érik el és azokat manipulálják.

Az egyes technikák céljai:

- Lekérdezési út a következő célokra szolgál:
- az igényelt rendszer logikai adatmodellje helyességének és érvényességének az ellenőrzése a szervezet információ-támogatási igényeivel szemben;
- egyértelmű módon a lekérdezések megfogalmazása / specifikálása;
- a lekérdezési folyamatok (Jackson-szerű) szerkezetének előállítás¹².
- Eseményhatás ábrák az aktualizáló, módosító folyamatok nem-procedurális, deklaratív¹³ szerkezetét írják le;
- A 'Lekérdezési folyamat modell' a lekérdezési utat alakítja át Jackson-szerű ábrává, ;
- Az 'Aktualizáló / módosító folyamat modell' az eseményhatás ábrát alakítja át Jackson-szerű ábrává amely bizonyos rendszer és programozási környezetekre közvetlen tervezési utat biztosít¹⁴ a módosító folyamatok részletes (fizikai) program tervének elkészítésére.

3.2 Összefoglaló áttekintés

Az eseményhatás ábra és a lekérdezési út az események és lekérdezések adatfeldolgozási tevékenységének a specifikációja. Lényegében az eseményhatás ábra tartalmát az entitáshasználati mátrixból, az entitás élettörténet diagramból és a logikai adatmodellből vezetik le, nevezetesen a következőképpen:

- mely entitásokra gyakorolnak hatást az egyes események, és vajon kifejtjenek-e többszörös, ismétlődő hatást;
- ha van ismétlődő hatás, akkor vajon az entitás ugyanazon példányára gyakorolt hatások között vannak-e alternatívák, vagy egyidejűleg, szimultán módon az adott entitás több entitás példányára fejt ki hatást a szóban forgó esemény;

Modelling, 3-91—3-109. Továbbá [CCAT90].

¹² A lekérdezési folyamatokhoz tartoznak nem csak a felhasználó által kezdeményezett interaktív adatvisszakeresési kérelmek, hanem a rendszer által indított jelentés készítések is.

¹³ A nem-procedurális, deklaratív leírási mód azt jelenti, hogy a 'Mit kell csinálni' kérdésre ad választ és nem a 'Hogyan kell csinálni' kérdésre, vagyis bizonyos adatfeldolgozási / programozási részletek érdektelenek ezen ponton. Pl. az ideiglenes adattároló helyek mérete (buffer), a háttértároló és a buffer között végrehajtott műveletek szervezése, stb., hasonló programozási részlet kérdések ezen ponton érdektelenek.

¹⁴ Ezeket a környezeteket nevezzük összefoglalólag harmadik generációs programozási nyelveknek, környezeteknek (3GL).

- az entitás élettörténet ábrán feltüntetett logikai adatbázis műveletek illetve az ábra logikájából következő műveletek összegyűjtése;
- az események között milyen kölcsönhatások fellépése, létrehozása célszerű.

Ezeket az információkat az eseményhatás ábra megrajzolásának kezdetekor kiindulásként használják, majd rendszeres átalakítások után az események által kiváltott folyamatok specifikációját kapjuk.

A lekérdezési út hasonló célokra szolgál mit az eseményhatás ábra; az entitás-használati mátrixból és a logikai adatmodellből alakítják ki és a szervezet információ-támogatási igényének kielégítése végett hozzák létre.

A lekérdezési út és az eseményhatás ábra szisztematikus átalakítása az aktualizáló (módosító) folyamat modell és a lekérdező folyamatok modellje, amelyek a korrekt Jackson jelölés technika szerint készülnek.

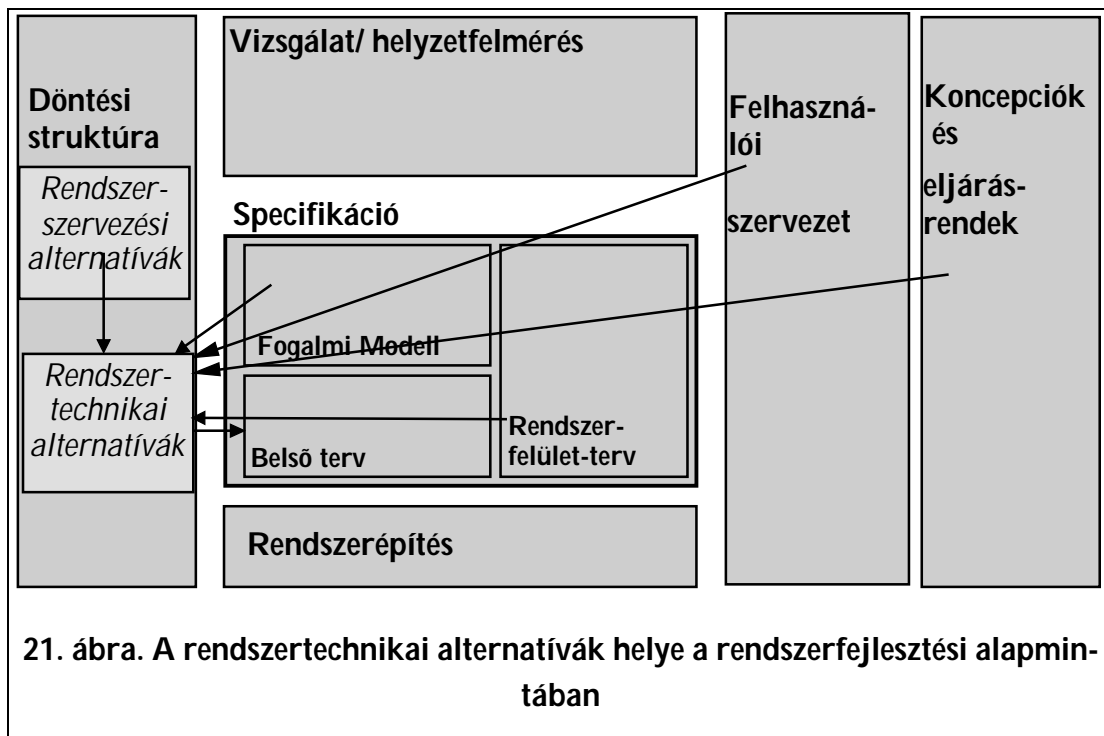
3.3 Termékek

A fogalmi folyamat modellezés termékei:

- Folyamatok specifikációja;
- Eseményhatás ábra;
- Lekérdezési út;
- Logikai rendszer specifikáció / Logikai rendszerterv / Logikai folyamat modell;
- Lekérdező folyamatok modellje;
- Aktualizáló / módosító folyamatok modellje;
- Eseményhatás ábra;
- Lekérdezési út.

4. Áttekintés a rendszerteknikai alternatíváról / javaslatról (TSO¹⁵) az SSADM 4+-ban¹⁶

A rendszerteknikai javaslat a kiválasztott rendszerszervezési alternatíva kivitelezési tervét fogja szolgáltatni. A rendszerteknikai alternatívák helyét az ábra mutatja a rendszerfejlesztési alapmintában (21. ábra.).



Ahogy azt az ábra is érzékelteti, a rendszerteknikai alternatívák egy kiválasztott rendszerszervezési javaslat részletes kidolgozása révén jönnek létre. A rendszerteknikai alternatívák kialakításának legfontosabb bemenetét a szervezet 'Konceptiók és eljárásrendek' összefoglaló néven nevezett alrendszeréből kapják, és általában a következőket tartalmazza:

- a technikai / műszaki architektúrát, ebbe beleértve a hardvert, a rendszer szoftvert, a kommunikációs, hálózati szolgáltatásokat, legalábbis ott, ahol ezeket az elemeket a szervezetben stratégiai szinten határozták meg;
- az alkalmazási architektúra jellemzőit, méretét, szervezési elveit, az alkalmazások közötti kölcsönhatásokat, a közösen használt, megosztott adatokat és a felhasználói felületet;
- beszerzési politikát, az előnyben részesített hardver és szoftver szállítókat, a jelenlegi informatikai vagyonnal kompatibilis, összhangban álló beszállítókat és a beszerzési folyamat előírásait;

¹⁵ Technical System Options

- a fentiek bármelyikét szabályozó szervezeti szintű előírásokat.

A rendszertechnikai alternatívák kidolgozása során a kiválasztott rendszerszervezési javaslat és a szervezet **politikájából**, céljaiból, szándékaiból valamint a szervezeti működési szabályzat előírásaiból, a szervezet **eljárásrendjéből** következő információkat kell alkotó módon összekombinálni a fogalmi modell és a rendszerfelület-terv elkészítése során létrehozott termékekben tárolt információkkal azért, hogy életképes alternatívákat tudjanak a tovább folytatás érdekében javasolni.

A rendszertechnikai alternatívák a következő kérdésekkel foglalkoznak:

- a technikai / műszaki környezet specifikációja, pl. hardver eszközök szállítása és elosztása, szoftver környezet, üzemeltetési rendszer;
- a végrehajtandó funkciók és megvalósításuk módjának elfogadása és megerősítése;
- (technikai / műszaki) változások hatása a szervezetre és a munkamódszerre;
- a projekt további részében az informatikai fejlesztő szervezetre és a fejlesztési infrastruktúrára gyakorolt hatás.

A rendszertechnikai alternatívák révén a projekt irányító közvetíteni tudja a felhasználók vezetése felé azokat a műszaki információkat, amelyek a fejlesztés további szakaszára vonatkoznak, továbbá a kapcsolódó költségeket, a környezetre gyakorolt hatásokat, és az elképzelt ütemezést. Ezzel a felhasználók vezetése lehetőséget kap arra, hogy megalapozott döntéseket hozzon, a lehető legjobb alternatíva kiválasztásával tekintettel a szervezet adottságaira és a projekt célkitűzéseire.

A kiválasztott rendszertechnikai alternatíva tartalmazni fogja a 'Technikai / műszaki rendszer architektúrát', amely valójában a fizikai környezet specifikációja, és a 'Fizikai tervezés' legjelentősebb bemenete.

4.1 A rendszertechnikai alternatívák kidolgozásához szükséges tapasztalatok és szakértelem

A rendszertechnikai alternatívák kidolgozásához szükséges tapasztalatok és szakértelem lényegesen eltér a rendszerelemzés és -tervezéshez általában elvártaktól. Azért, hogy életképes alternatívákat tudjanak kidolgozni a fejlesztő csoportnak értenie kell egy csomó technikai részlet kérdéshez.

Fontosabb nem-SSADM szakismeretet igénylő területek:

¹⁶ [CCTA95], [CCTA95A], Reference Manual Part 8: Formulating Options, Technical System Options, 8-21—8-37. Továbbá [CCAT90].

- jelenleg, a piacon rendelkezésre álló technológiák műszaki teljesítőképessége, mi az ami most reális;
- a technológiák lehetséges kombinálhatósága — adatbáziskezelő rendszerek (DBMS), tranzakció-monitorok (TP software), kommunikációs, hálózati szolgáltatások, felhasználói felületet kezelő eszközök, 'polcra megvásárolható' termékek mint például elektronikus levelező rendszerek, iroda automatizálási csomagok (szövegszerkesztők, kalkulációs lapok);
- informatikai rendszerek méretezése és kapacitás tervezés¹⁷;
- hardware és szoftver beszerzés¹⁸;
- költség / haszon elemzés.

Ideális esetben ezek az ismeretek rendelkezésre állnak a fejlesztő csoportban, ha még sem akkor külső szakértők bevonására van szükség.

4.2 A rendszertechnikai alternatívák termékei

A rendszertechnikai alternatívák SSADM termékei kiválasztás előtt:

- Költség-haszon elemzés;
- Hatáselemzés;
- Vázlatos fejlesztési terv;
- A technikai / műszaki rendszer architektúra vázlatos leírása;
- Rendszerleírás.

A kiválasztás után:

- a kiválasztott rendszertechnikai javaslat;
- Vázlatos fejlesztési terv;
- a választás indoklása;
- A technikai / műszaki rendszer architektúra;
- Hatáselemzés;
- Rendszerleírás.
- (Alkalmazásszintű környezeti útmutató)¹⁹.

¹⁷ CCTA, "A Guide to SSADM and Information Systems Procurement", ISE Library, HMS, 1994.

¹⁸ CCTA, "SSADM and Capacity Planning", ISE Library, HMS, 1994.

¹⁹ Az alkalmazásszintű környezeti útmutatót a szervezetszintű környezeti útmutatóból alakítják ki. Mindkét útmutatót a rendszertechnikai alternatívák előtt fejlesztik ki.

A technikai / műszaki rendszer architektúra leírása

A rendszertechnikai alternatívák részeként a technikai architektúra a kiválasztás előtt csak vázlatosan kerül leírásra. Csak a megfelelő alternatíva kiválasztása után kell a technikai architektúrát önálló terméként részletesen leírni.

A célja az, hogy elegendő információt nyújtson a felhasználóknak a rendszer működésének megértéséhez, a (felhasználók szempontjából) lényeges tervezési paraméterek kifejtése (pl. válaszidő, erőforrások kihasználtsága, áteresztőképesség), illetve a részletes költségbecslések elvégzéséhez. Tartalmaznia kell információkat a hardverről, szoftverről, fejlesztői környezetről, rendszer-méretről (adat és feldolgozás szempontjából), valamint bármely további jelentős tényezőről, mint például meghibásodások között eltelt átlagos idő és visszaállítás erő- és időigénye, biztonsági módszerek, eljárások.

Rendszerleírás

Ez azt írja le, hogy a követelmény-specifikációt hogyan lehet az alternatíva által megvalósítani. A legtöbb esetben a fontosabb döntéseket már a rendszerszervezési alternatívák kiválasztása során meghozták. Ide tartozó termékek:

- Igényelt rendszer logikai adatmodellje;
- Funkció-meghatározások;
- Követelményjegyzék.

Hatáselemzés

Ez a dokumentum az alternatíva környezetre gyakorolt hatását írja le és a szervezetre, eljárásrendekre, megvalósításra vonatkozó megfontolásokat tartalmazza. A követelmény-specifikációra vonatkozó hatásokat is fel kell jegyezni. Ide tartozó termékek:

- Oktatási igények leírása;
- A felhasználói kézikönyvvel szemben támasztott követelmények;
- A tesztelési eljárások, követelmények vázlatos leírása;
- A rendszer áttéréssel kapcsolatos követelmények, előírások.

Vázlatos fejlesztési terv

Az adott alternatívához a projekt további menetére vonatkozó fejlesztési stratégiát kell meghatározni azért, hogy a projekt tervezett időtartamát és az erőforrás-igényeket, és ezáltal a fejlesztés költségeit meg lehessen becsülni.

Költség-haszon elemzés

A formális költség-haszon elemzés egy olyan objektív eszköz, amellyel össze lehet vetni két alternatíva számszerűsíthető adottságainak értékét. Emiatt a költség-haszon elemzés egy nagyon fontos (pénzügyi) része az alternatívák specifikációjának. Meg kell próbálni a nem számszerűsíthető előnyöket is egymáshoz viszonyítva kiértékelni, bár ezekhez nehéz költségeket rendelni.

5. Rendszertехnikai alternatívák kialakítása

5.1 A technika rövid leírása

A rendszertехnikai alternatívák kialakítása az az eszköz, amellyel a projektirányító információt nyújt a felhasználói vezetés részére a továbbhaladás módjáról, költségeiről, feltételeiről és időtávjáról. Ennek alapján a felhasználói vezetés döntést hoz, kiválasztva a szervezet és a projekt célkitűzései szempontjából legmegfelelőbb továbbhaladási irányt.

Ezt a választott rendszertехnikai alternatívát ki kell egészíteni a választás indoklásával, a technikai környezet leírását pedig ki kell egészíteni a fizikai környezet specifikációjával, ami bemeneteként szolgál a fizikai tervezésnek.

Az alternatívák kialakítása itt is hasonlóan történik mint a megvalósíthatóság elemzése vagy a rendszerszervezési alternatívák esetén:

- jelentősebb korlátok azonosítása
- lehetséges megoldások általános vázlatának kialakítása
- vázlatok kiegészítése
- alternatívák bemutatása
- a döntések részleteinek feljegyzése és a választott alternatíva kiegészítése

5.2 Kapcsolat más technikákkal

Fizikai tervezés

A fizikai tervezés technikáit (fizikai adattervezés, fizikai feldolgozás tervezése) fel lehet használni a rendszer durva méretezésére (pl. kezdeti adatterv készítése). Ha ez nem elegendő akkor a fizikai terv első változatát teljesen el kell készíteni.

Rendszerszervezési alternatívák

A kiválasztott rendszerszervezési alternatíva az alapja a lehetséges rendszertехnikai alternatívák kidolgozásának. Gyakran a rendszerszervezési alternatívában már egy vázlatos rendszertехnikai alternatívát megadnak. Ahol a kiválasztott rendszerszervezési alternatívában megfogalmazott feltételezések nem állják meg a helyüket, ott ezt ki kell emelni.

Követelmény meghatározás

Az egyik legfontosabb bemenete a rendszertехnikai alternatívák kialakításának, különösen a nem-funkcionális követelmények, amelyek a rendszer méretezéséhez és kapacitás tervezéshez szükségesek.

Dialógus tervezés

Ha a rendszertechnikai alternatívák készítése során az alkalmazás szintű környezeti útmutató elkészült, akkor ez a dialógus tervezés egyik fontos bemenete lesz.

Projekt-eljárások

A rendszertechnikai alternatívák kialakítása során sok olyan területet kell érinteni, amely nem tartozik az SSADM módszerbe. Két fajta tevékenység kapcsolódhat ide, az egyik információt nyújt a rendszertechnikai alternatívák kialakításához, a másik nyers adatokat vagy információkat kap a rendszertechnikai alternatívák kialakításának tevékenységeitől.

A következő területeket kell érinteni:

- kapacitástervezés, amit nyers adatokkal lát el a rendszertechnikai alternatívákat kialakító tevékenység, illetve ahonnan ugyanez a tevékenység információt kap az alternatívák ellenőrzéséhez
- becslés (az SSADM tevékenységekre), ami nem része az SSADM módszernek, de szükséges a rendszertechnikai alternatívák kifejlesztésének megtervezéséhez
- helyi jellegű és a projektre vonatkozó szabványok, amelyek az alternatívák készítésének és dokumentálásának módját befolyásolják
- kockázatelemzés és -kezelés, amely a kialakított alternatívákat felméri a biztonságtechnikai követelmények kielégíthetősége szempontjából és információt ad az elemzőknek az alternatívák kifejlesztéséhez
- tesztelési előírás, amelyet az rendszertechnikai alternatívák által nyújtott adatok alapján lehet kialakítani
- képzés, amelyre képzési stratégiát lehet kialakítani a alternatívák által leírt szervezeti hatások felmérése alapján
- felhasználói kézikönyv, amelynek kialakítását el lehet kezdeni a választott alternatívában meghatározott felhasználó és rendszer közötti felület leírása alapján
- projekttervek, amelyeket a rendszer kifejlesztésére ki lehet alakítani

5.3 A rendszertechnikai alternatívák kialakítói

5.3.1 Szerepek

A következő szerepeket kell betölteni a rendszertechnikai alternatívák kialakítása során:

Rendszerelemző

A rendszerelemző felméri és dokumentálja a követelményeket, valamint összeállítja a rendszertechnikai alternatívákat a projektvezetés számára.

Felhasználó

A felhasználó:

- felveti a követelményeket, amelyeket a rendszerelemző értelmez és feljegyez
- megszabja a projekt irányát az szervezeti célkitűzéseknek megfelelően
- sok szerepben jelenik meg a projekt során, a végfelhasználótól kezdve a felső vezetés szintjéig.

Minden felhasználó a beosztásának megfelelő információt és útmutatást adja. Ebben a szakaszban felhasználónak a projekt közvetlen befolyásolására jogosult vezetői szintet kell tekinteni.

Projekt irányító

A projektirányító véglegesíti a rendszertechnikai alternatívákat és bemutatja őket a projektvezetésnek, kiemelve az előnyeiket és hátrányaikat.

Projektvezetőség

A projektvezetőség kiértékeli az alternatívákat és választ közülük. Dönthet úgy, hogy befejezi a projektet, ha nincs megfelelő alternatíva, amellyel el lehetne érni a projekt célkitűzéseit.

5.3.2 A döntéshozó folyamat

Az SSADM csak egy általános megközelítést ad a projektirányítás számára, amelyet a konkrét körülményekhez kell igazítani. Célszerű felmérni, hogy kiket kell bevonni a döntéshozásba. A projekt munkacsoport tagjait természetesen be kell vonni. Azokat is be kell vonni, akik a kiadásokért, költségekért felelnek, valamint akik az üzletpolitikát jól ismerik. A kiválasztásért felelős csoport összetétele a következő lehet:

- a projektvezetés a vezető üzleti felelős elnöklésével, valamint az érintett részlegek vezetői
- egy speciális vizsgáló csoport, amely főleg a felhasználók képviselőiből áll, de részt vehetnek benne az információ-technológia termékeinek szállítói is
- az általános minőségbiztosító csoport, ha létezik
- a felhasználók széles körének megkérdezése után a projektvezetőség hozza a döntést.

5.4 Korlátok

Az egyes alternatívák megfontolása előtt hasznos lehet felmérni azokat a rendszerkorlátokat, amelyek leszűkítik az elemzők előtt álló lehetőségeket.

Az elemzőnek meg kell vizsgálnia a rendelkezésre álló termékeket. Azonosítania kell a rendszernek és környezetének azokat az elemeit, amelyek körvonalazzák a végső alternatívát. A rendszertechnikai alternatívák szempontjából relatív fontossági sorrendet kell felállítani a perem feltételek kielégítése között, feloldva ezzel a konfliktust olyan egymásnak ellentmondó célkitűzések között, mint pl. a teljesítmény, a kapacitás, tárolási igények stb. Kétféle korlátot kell figyelembe venni:

- "Külső", a projektre kívülről ható korlátok;
- "Belső", a projekt határain belül felismert és dokumentált célkitűzésekre és szolgáltatási szintekre vonatkozó korlátok.

5.4.1 Külső korlátok

A legfontosabb korlátozások a választott rendszerszervezési alternatívából származnak, amelyet szintén korlátoz az információrendszerekre vonatkozó stratégia.

A külső korlátok az összes alternatívára vonatkoznak, így a rendszertechnikai alternatívák általános határai, kiterjedését és kereteit határozzák meg. Ilyen korlátok lehetnek pl.:

- idő, "Az új rendszernek működnie kell legkésőbb ..."
- költség, "A teljes fejlesztési költségek nem léphetik túl a Ft-ot"
- üzleti / működési / szervezeti teljesítmény a projekt értékével összevetve, "A jelenlegi kiadásokat n év alatt évi x Ft-tal kell csökkenteni"
- hardver/szoftver, "Az új rendszert a létező gépeken kell megvalósítani a jelenleg használatos adatbázis-kezelőre alapozva"

Meg kell vizsgálni a felhasználókkal együtt, hogy a külső korlátok valós megfontolásokat tükröznek-e vagy önkényesen lettek meghatározva.

5.4.2 Belső korlátok

Azokat a jelentősebb korlátokat kell meghatározni, amelyeket a projekten belül a felhasználók fogalmaztak meg. A következő területeket kell figyelembe venni:

- kötelezően nyújtandó szolgáltatások: interaktív hozzáférés, szövegszerkesztés
- minimális általános szolgáltatási szintek:
- meghibásodások közötti átlagos időszak
- a rendszer-visszaállítás maximális időtartama
- a mentési rendszer teljesítőképessége
- rendelkezésre állás
- megbízhatóság

- katasztrófa helyzetek kezelése (kontingencia terv)
- adattárolási előírások, az igényelt rendszer adatmodellje alapján:
- maximális állományméretek
- rendszer- és adatmentéshez szükséges anyagfelhasználás
- kritikus időtényezők előírása, a funkcióleírások alapján:
- a legmagasabb interaktív terhelési csúcsok
- a legkritikusabb azonnali (on-line) válaszidő
- a legnagyobb tranzakció-mennyiség
- Az információk célkitűzések, amelyeknek a relatív fontossági sorrendjét meg kell állapítani a logikai adatmodell és a funkcióleírások együttes használatával. Meg kell jelölni azokat az adatelemeket, amelyek elérését semmiképpen nem lehet korlátozni a teljesítményre vonatkozó előírások betartásának érdekében.
- Egyéb célkitűzések, mint pl.:
- működtető környezet feltételei
- biztonsági követelmények
- adatbázis-kezelő rendszer átszervezésének időzítése és gyakorisága
- kapcsolatok más információs rendszerekkel

5.5 A rendszertехnikai alternatívák kifejlesztése

5.5.1 Vázlatos alternatívák készítése

Miután a korlátok azonosításra kerültek, lehetővé válik néhány, a rendszer követelményeit kielégítő, vázlatos alternatíva kifejlesztése. Néha lehet "ötletbörzét" tartani, ami nagyon szubjektív, de egyben kreatív is. Hasznosabb néha, ha egy kisebb, három fős, csoport fogalmazza meg a kezdeti felvetéseket, főleg ha külső felmérésre is szükség van. A külső felmérés technikai adatok összegyűjtését jelenti, általában maguktól a szállítóktól, olyan dolgokról, mint költségek, szolgáltatások, teljesítmény. Itt nem szállítót kell választani, hanem inkább bizonyos konfigurációkról kell eldönteni, hogy megfelelnek-e a követelményeknek illetve korlátoknak.

Általában háromtól hatig terjedhet a kezdeti alternatívák száma, ami a következőktől függhet:

- meg kell vizsgálni a megvalósíthatóságot, ha a projekt kiterjedését elfogadott módon megváltoztatták
- ha a projekt egy manuális rendszer helyettesítésére irányul, akkor be kell venni a "számítógép nélkül" alternatívát

- ha egy létező számítógépes rendszert kell felváltani, akkor a "változtatás nélkül" alternatívát is meg kell vizsgálni, aminek kiválasztása esetén a teljes projektet be kell fejezni.

5.5.2 A vázlatok számának csökkentése

Mivel a hat alternatíva részletes kidolgozása túl sok munkába kerülne, ezért el kell érni egy kezelhetőbb mennyiséget, általában hármat. A következőket kell figyelembe venni:

- Minden vázlatos alternatívához kell egy vázlatos hatáselemzést végezni, felsorolva a fontosabb előnyöket és hátrányokat a felhasználók szempontjából. Meg kell próbálni valamilyen értéktartományt rendelni minden felsorolt tényezőhöz.
- Mindig át kell nézni a vázlatos alternatívákat a felhasználókkal, azért, hogy ki lehessen szűrni az elfogadhatatlan tényezőket. Természetesen teljes alternatívákat nem kellene megszüntetni emiatt, de a kezdetben vonzónak és megvalósíthatónak tűnő elemeket össze lehet gyűjteni három erős alternatívában.
- Nem szabad megszüntetni minden alternatívát a "teljesen alkalmatlanon" kívül, kiválasztva azt a részletes kiértékelés előtt.

5.5.3 Alternatívák kialakítása

Itt ki kell terjeszteni és átfogóbbá kell tenni a fentiek szerint kialakított, kezelhető számú alternatívát.

A rendszertechnikai alternatívákat a hardver/szoftver környezetre épülve kell specifikálni. Lehet sok olyan szempont, ami választási lehetőséget rejt. A kezelhetőség érdekében ezeket az rész-alternatívákat a fő alternatívák köré kell csoportosítani.

Ha szükséges a rendszer teljes méretével számolni egy adott hardver/szoftver konfiguráció megfelelőségének eldöntése érdekében, akkor egy kapacitástervezési elemzést lehet elvégezni az SSADM termékek alapján.

A rendszer műszaki architektúrájának korlátait, az alkalmazás felépítését és a beszerzési politikát szem előtt tartva mérlegelni kell:

- a rendszerfelület tervének megvalósítására szóba jövő technológiákat (karakter alapú vagy GUI);
- a fogalmi modell folyamatainak megvalósítására alkalmas technológiákat;
- adattárolás és adat-visszakeresési technológiákat;
- a fogalmi modell, a rendszerfelület-terv és a belső terv közötti kommunikációt megvalósító technológiákat.

5.5.4 A kiválasztás lépései

Az alternatívák kialakítása után be kell őket mutatni a felhasználói képviselőknek. Négy lépésben lehet ezt megtenni:

- felkészülés a bemutatásra
- bemutatás
- további részletezés és kérdések megválaszolása
- a választás indokainak feljegyzése

5.5.5 A döntéshozatal

A projekt menetének szempontjából fontos, hogy a kiválasztás indokolatlanul ne húzódjon el. A döntés előírt dátumát fel lehet venni a projektervbe, amivel elkerülhető a felesleges időhúzás.

Sajnos a kiválasztási döntés ritkán jelenti egyetlen alternatíva kiválasztását. Általában a választott alternatíva egy "vegyes saláta", ami egy alternatíván alapul, de több másik alternatíva elemeit is tartalmazza.

5.5.6 A választás dokumentálása

A döntés részleteit érdemes feljegyezni, hogy biztosítani lehessen a projekt további menetében az igazodást mind a döntés szelleméhez, mind pedig a betűjéhez. A döntés után szükség lehet a választott rendszertechnikai alternatíva és a technikai rendszer architektúra leírásának kiegészítésére. A választott alternatívát ismét meg kell vizsgálni a kapacitástervezés segítségével az igényelt szolgáltatási szintek betarthatósága szempontjából. Ha ezek nem tarthatók, akkor három lehetőség van:

- nagyobb kapacitású architektúrát lehet javasolni;
- csökkenteni lehet az szolgáltatási szintekre vonatkozó előírásokat;
- javasolni lehet a követelmény-specifikáció megváltoztatását.

5.6 A technikai rendszer architektúra leírásának kiegészítése

A technikai rendszer architektúra leírása az, amit a fizikai tervezés felhasznál. A rendszertechnikai alternatíva nem technikai részei, amelyek a vezetői információkat és indoklást tartalmaznak, továbbra is benne maradnak a választott alternatívában. A technikai környezet leírása a rendszer fejlesztési és megvalósítási környezetének leírásával segíti a követelmény-specifikáció megvalósítását. Módosítani kell ahhoz, hogy napra készen tükrözze a választást. Tartalmazni fogja az előzőleg meghatározott részeket, valamint a választott alternatíva bizonyos további részeit:

Rendszerleírás

Itt a követelmény-specifikációban leírt funkcionalitás esetleges megváltozását kell hangsúlyozni, a változások szöveges leírásával és hivatkozásokkal a specifikáció érintett részeire.

Hatáselemzés

Ez a rendszertехnikai alternatíva hatáselemzésén alapul és információkat tartalmaz azokról a döntésekről, amelyek közvetlenül befolyásolják a rendszer megvalósítását:

- az új rendszer felhasználói szervezete és személyzete, beleértve esetleg az informatikai szállítókat is
- a felhasználói felület, illetve egyéb rendszerekkel való kapcsolódási felület eljárásainak vázlatos leírása
- a projekt elérendő céljainak meghatározása, ami főleg az alternatívában leírt előnyöket jelenti, ahogy azt a költség-haszon elemzés számszerűsítette. Ezekre a jövőben lesz szükség:
 - annak ellenőrzésére, hogy a rendszer ténylegesen hozza a várt hasznot
 - a javasolt módosítások fontosságának és jelentőségének ellenőrzésére.

5.7 A rendszertехnikai alternatíva alkotóelemei

Egy rendszertехnikai alternatíva a következő dokumentumokból áll:

- Technikai környezet leírása
- Rendszerleírás
- Hatáselemzés
- Vázlatos fejlesztési terv
- Költség/haszon elemzés

5.7.1 Technikai környezet leírása

5.7.1.1 Hardver

Ez egy áttekintő ábrából álló leírás, kiegészítve az eszközök típusának, számának és elhelyezkedésének részleteivel. A következő tényezőket kell érinteni:

- szabványok
- kommunikáció/hálózatok
- környezet

- üzem behelyezés
- működtetés
- az újabb változatok bevezetéséről szóló megállapodás
- megbízhatóság
- hatékonyság
- rendelkezésre állás
- karbantarthatóság

5.7.1.2 Szoftver

Ez egy leírás az igényelt rendszer-szolgáltatásokról, a beszerzés módjáról, és az alkalmazói szoftver mennyiségi adatairól. Tipikus dolgok, amiket figyelembe kell venni, a következők:

- az adatkezelő rendszer típusa
- az igényelt kiegészítő szolgáltatások, pl. memória listázás (dump) vagy visszaállási lehetőségek (recovery)
- az operációs rendszer lehetőségei
- alkalmazói csomagok
- alkalmazói programok előállításának módja, pl. 3GL vagy 4GL
- alkalmazói programok száma
- fejlesztői környezet

5.7.1.3 Rendszer méretezése

A hardver- és szoftverkörnyezet leírása előtt szükség lehet a rendszer méretezésére, a következő területeken:

- az adatok, melyeket százalékosan lehet kifejezni az adott hardver/szoftver környezet ismeretében a környezet által nyújtott szolgáltatások mennyiségi adataira vetítve. A következő módon lehet számítani:
- módosítsuk az igényelt rendszer logikai adatmodelljét az alternatíva alátámasztása érdekében (ha szükséges)
- a létrejövő adatmodellt egészítsük ki mennyiségi adatokkal
- becsüljük meg minden egyed egy rekordjának méretét
- számoljunk ki egy teljes becsült értéket a logikai adatokra
- adjuk hozzá a becsült további terhelést (túlcsordulás, kiterjesztés, mutatók, indexek).

- a feldolgozás, amit nehezebb számolni. Egy lehetséges megközelítés a következő:
- válasszuk ki az alternatívának megfelelő funkcióleírásokat, eseményhatás-ábrákat és B/K adatszerkezeteket
- gondoskodjunk róla, hogy a mennyiségi és gyakorisági adatok meglegyenek
- becsüljük meg az átlagos feldolgozási időt egy entitás aktualizálására, beleértve olyan tételeket, mint a bemenő/kimenő adatforgalom, alkalmazói program, tranzakció figyelés (monitor) stb.
- számítsuk ki minden egyes funkció feldolgozási idejét
- számítsuk ki a becsült feldolgozási terhelést minden feldolgozási ciklusra, felhasználva az adott eseményhez tartozó mennyiségi és gyakorisági adatokat és az esemény számolt feldolgozási idejét
- a funkcióleírások alapján vegyük hozzá a nem aktualizáló eseményekkel kapcsolatos feldolgozási becsléseket (pl. lekérdezések, belső állományok aktualizálása stb.)
- A fizikai adatterv illetve fizikai terv első változatának elkészítésére esetleg szükség lehet, ha az alternatívák különbözősége miatt másképpen nem lehet a fizikai megvalósítás hatásait felmérni.

5.7.1.4 További részek

- rendszer-összeomlási és - visszaállítási megfontolások
- hozzáférési jogok
- hozzáférés-ellenőrzési és biztonsági módszerek
- hardver/szoftver karbantartás

5.7.2 Rendszerleírás

Ez azt írja le, hogy az adott alternatíva hogyan tesz eleget a követelmények specifikációjának. Általában a fontosabb döntéseket ezen a területen már a rendszerszervezési alternatíva kiválasztásakor meghozták. Ennek ellenére, néha szükség lehet olyan alternatívákat felvetni, amelyek az igényelt rendszert különböző szintig érik el, mérlegelve például a szolgáltatásokat a költségekkel és fejlesztési idővel szemben.

A rendszer követelményeinek kielégítettségi fokát jelezni kell. Általában ez a meglévő SSADM termékek módosítását jelenti, különösen a következőkre vonatkozóan:

- igényelt rendszer logikai adatmodellje,
- funkcióleírások,
- követelményjegyzék (az alternatívát tükröző megoldásokkal).

Egy alternatíva viszonylagos súlyát lehet jelezni egy olyan listával, amely a nem megvalósítandó funkciókat/szolgáltatásokat tartalmazza.

5.7.3 Hatáselemzés

Ez a dokumentum az alternatíva felhasználói környezetre gyakorolt hatásait írja le. A hatáselemzés lehetőséget ad olyan kérdések felvetésére, amelyek ugyan közvetlenül nem érintik az SSADM-et, de befolyásolni fogják a megvalósítandó információrendszer minőségét. A főbb témák a következő termékekben jelennek meg:

- oktatási előírások,
- felhasználói kézikönyvre vonatkozó leírások,
- tesztelési előírások,
- áttérési előírások.

További témák lehetnek:

- szervezet és személyzet,
- jelentősebb változások a felhasználókra vonatkozó működési és szervezeti szabályzatban,
- megvalósítási megfontolások (adatátvétel, a betanulási időszak hatásai a szolgáltatási színvonalra),
- megtakarítások, a helyettesített berendezések, karbantartások tekintetében,
- előnyök és hátrányok a többi alternatívával összehasonlítva,

előnyök lehetnek:

- növekvő forgalom és termelékenység,
- elért üzleti célkitűzések,
- könnyű és gyors kivitelezés,
- alacsonyabb fejlesztési költségek,
- megbízhatóság,
- munkaerő megtakarítás,
- jobb teljesítmény és szolgáltatás,

hátrányok lehetnek:

- a javulás korlátai,
- nem elért üzleti célkitűzések,

- kivitelezési nehézségek, illetve hosszabb időtáv,
- magasabb megvalósítási költségek,
- alacsonyabb teljesítmény.

5.7.4 Vázlatos fejlesztési terv

Ez alkotja a kiindulópontot a projekt további menetére vonatkozó fejlesztési stratégia kialakításához az adott alternatívában. A cél az, hogy előzetes idő tartamokat és erőforrás-igényeket, és ezzel együtt fejlesztési költségeket lehessen megbecsülni. Csak a következő modult lehet részletesen becsülni, a fizikai rendszertervezés utáni tevékenységek becslése pontatlanabb. A következőket kell a tervnek tartalmaznia:

5.7.4.1 Rendszertervezés

Az igényelt munka és az erőforrás-igény együttes becslése, a projekt időtartamára, azaz:

- a projekt további menetének vázlatos ütemterve,
- a fizikai rendszertervezés részletes terve:
- részletes feladatlista, az SSADM feladatokat a projekt körülményeihez igazítva,
- a feladat elvégzéséhez szükséges munka becsült mennyisége,
- az igényelt erőforrások becsült mennyisége,
- a projekt végrehajtásának ütemezése,
- a következő fázis részletes költségvetése.

5.7.4.2 Programtervezés és programozás

A rendszer felépítésére (pl. kódgenerátorok, "testre szabás", csomagok stb.) és kifejlesztésére (pl. szerződéses, saját erős, kulcsrakész stb.) vonatkozó stratégia megfogalmazása, az igényelt erőforrások és időtávok becslésével.

5.7.4.3 Beszerzés

Ez a beszerzési stratégia (kulcsrakész, több szállító, stb.) és a becsült időtávok megfogalmazása, világosan azonosított mérföldkövekkel.

5.7.4.4 Rendszertesztelés

Az erőforrás- és időigények becslése.

5.7.4.5 Megvalósítás

Az adatátvétel és az új rendszerre való áttérés stratégiájának megfogalmazása, az erőforrás- és időigények becslésével.

5.7.5 Költség-haszon elemzés

A formális költség-haszon elemzés egy olyan objektív eszköz, amellyel össze lehet vetni két alternatíva számszerűsíthető adottságainak értékét. Emiatt a költség-haszon elemzés nagyon fontos pénzügyi része az alternatívák specifikációjának. Ez a projektirányítás hatáskörébe tartozik ugyan, de a rendszerelemző rendelkezik az adatokkal, ami alapján ezt a pénzügyi elemzést el lehet végezni. A fő területek a következők:

5.7.5.1 Fejlesztési költségek

Két dokumentumból lehet kiindulni:

- technikai rendszer architektúra leírása, ahol a hardver és szoftver költségek vonatkozhatnak egy tipikus szállítóra.
- Vázlatos fejlesztési terv

5.7.5.2 Üzemeltetési költségek

Kiindulópont:

- Technikai környezet leírása
- Hatáselemzés

5.7.5.3 Megtakarított költségek

Ezek olyan költségek, amelyeket a jelenlegi rendszer támasztott, de az új rendszer nem fog támasztani. Kiindulópont:

- Technikai környezet leírása
- Hatáselemzés

5.7.5.4 Hasznok

A hatáselemzésből lehet ezeket meghatározni, a következő három besorolás szerint:

- a kézzel fogható hasznok, pl. megnövelt nyereség;
- költség megtakarítás, az az összeg, amit ki kellene adni, ha az új rendszer nem állna üzembe, pl. a munkaerő mennyiségének növelése a növekvő munkaterhek ellensúlyozására;
- nem kézzel fogható hasznok, amelyeket nem lehet számszerűsíteni. Hasznos lehet mégis valamilyen értéket rendelni ezekhez, ami utal a jelentőségükre. Általában megvan a veszélye annak, hogy ide sorolunk olyan dolgokat, amelyek nem igazából megfoghatatlanok, hanem inkább nehezen számíthatók.

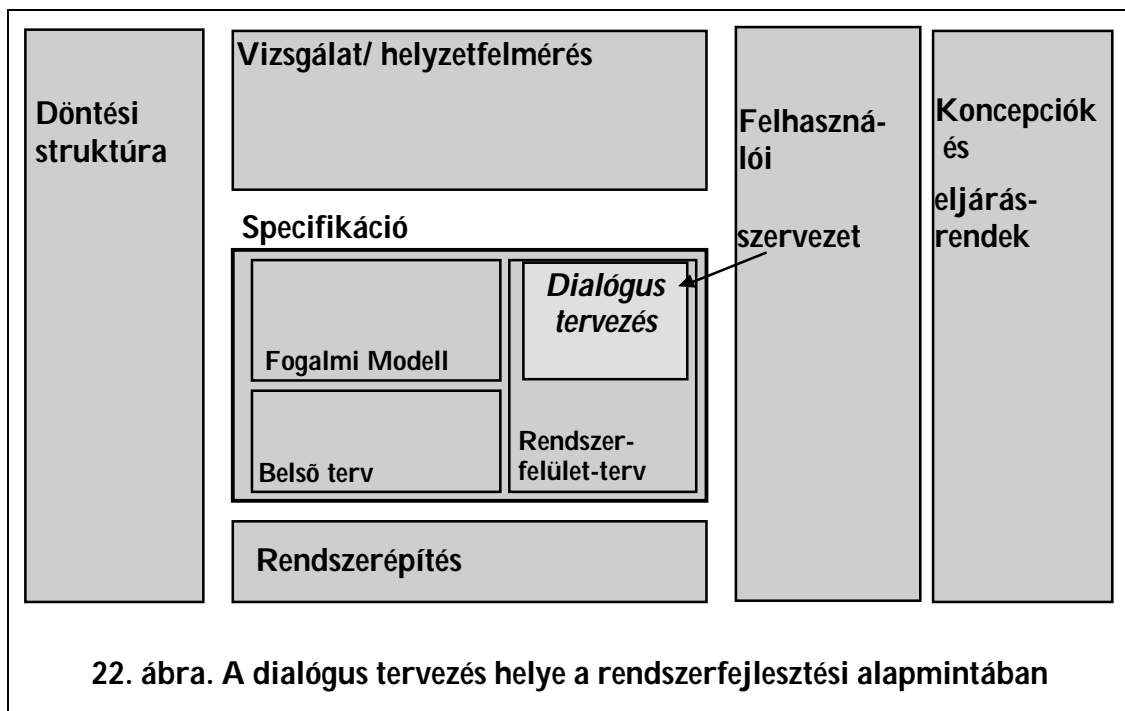
6. Áttekintés a dialógus tervezésről az SSADM 4+-ban²⁰

A dialógus tervezés a rendszerfelület tervezés része, amely a rendszerfejlesztési alpminta specifikációs részéhez tartozik. A dialógus tervezéshez szükség van a szervezet alapos ismeretére a 'Felhasználó jegyzék' elkészítéséhez és a felhasználói szerepkörök azonosításához.

A dialógus tervezés tartalmazza a dialógusok felismerésének, azonosításának és megtervezésének technikáját. Ezeket a technikákat az interaktív tevékenységek megragadására és ábrázolására lehet használni. Ahol csak lehetséges a dialógus tervezését olyan eszközben kell elvégezni, amely a cél környezet szolgáltatásaihoz és sajátosságaihoz nagyon közel áll. A papír alapú specifikációhoz képest ezt a megoldást érdemes előnyben részesíteni.

A dialógusokat két szinten kell specifikálni:

- az adat elemek logikai szintű specifikációjával, amely a felhasználók és a funkciók között lezajló párbeszéd adatelemei és üzenetei tartalmának az értelmében történik;
- a logikai dialógusok leképezése az új rendszer cél környezetének technológiájára, ami lehet valamilyen ablak-alapú²¹ vagy karakteres párbeszédet lehetővé tevő technológia.



Noha valójában a második specifikációs szint a fizikai tervezés részének tekinthető volna, azonban a logikai tervezés sem hajtható úgy végre, hogy nem vesszük figyelembe a cél tech-

²⁰ [CCTA95], [CCTA95A], Reference Manual Part 5: Modelling from the User's Perspective, Dialogue Design, 5-57—5-89, User Guide Part 3: Specification (External Design), Dialogue Design, 3-137—3-162. Továbbá [CCAT90].

²¹ Ezek a grafikus felhasználói felületek (GUI, Graphical User Interface). További információk találhatóak, CCTA, 'SSADM and GUI design: A Project Managers Guide', ISE Library, HMSO, 1994.

nológia tervezési korlátait, jellemzőit. Enélkül a logikai terv teljesen hatékonytalan megoldásokra vezethet, akár a felhasználók ergonómiai szempontjaira akár a rendszer teljesítményére gondolunk. Ezért a cél technológia a tervezést már a fizikai tervezésnél sokkal korábban befolyásolja.

6.1 Cél

Az összes interaktív, párbeszédés felhasználó-gép információ cserét a dialógusok formájában dokumentálni kell. A rendszer funkciói, szolgáltatásai alapján történik a dialógusok azonosítása és megtervezése. A felhasználókkal folytatott sűrű egyeztetések során alakítják ki a dialógusokat és specifikálják azokat az adatokat, amelyek a rendszer és a felhasználó közötti párbeszédben átadnak. A dialógusok közötti közlekedést — mely dialógusokhoz milyen módon lehet hozzáférni — a felhasználók munkaköri leírásának, az alkalmazandó technológiának és a felhasználók tapasztalatai és ismeretei fényében kell megtervezni. Ez a technika a felhasználók és a rendszer közvetlen kapcsolatára koncentrál és ezért a rendszer minősége szempontjából kritikus tényező.

6.2 Összefoglaló áttekintés

A dialógusok felismerése és megtervezése független a rendszer adatfeldolgozási követelményeitől. Ezért a dialógusok specifikálása a rendszer funkcióira és a felhasználói szerepkörökre támaszkodik és nem a rendszer informatikai eseményeiből, lekérdezéseiből származtatják a dialógus tervet. Ezért a 'rendszer oldalt' a funkciókhoz kötődő események jelenítik meg, míg a 'felhasználó oldalt' a dialógusok ábrázolják²².

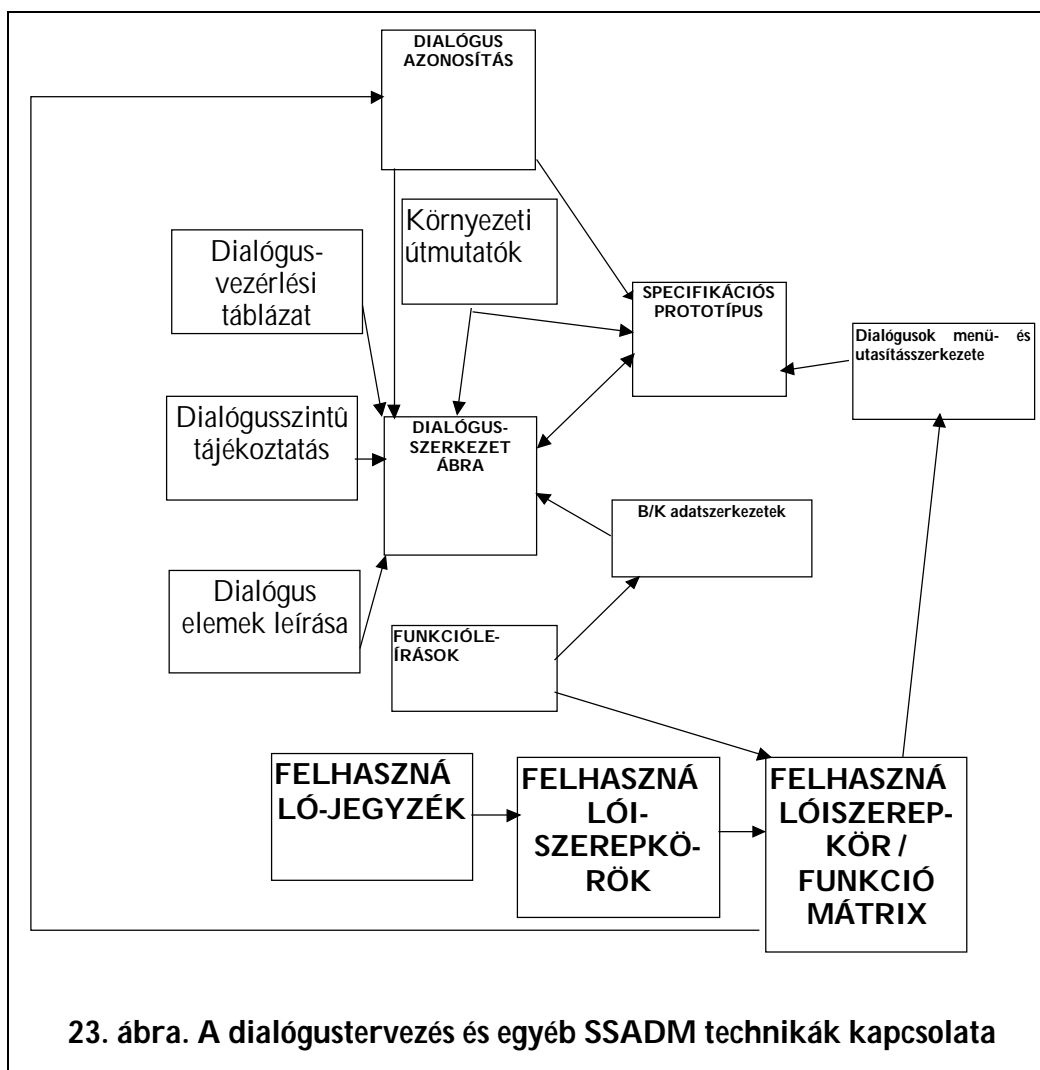
A dialógusok kifejlesztése történhet valamilyen prototípus-készítő eszköz bevonásával, így a dialógusok közvetlenül a cél technológiai környezetben készíthetők el; ebben az esetben a projektben a logikai dialógusok specifikációjának elkészítésére nem lesz feltétlenül szükség. Ez valójában egy projektvezetési döntés, amiről a projektvezetőséget informálni kell, mivel ennek a döntésnek vannak bizonyos kockázatai, nevezetesen a dialógusok leírásai egy adott technológiára lesznek specifikusak és ezért a későbbiekben kevésbé rugalmasan lesznek felhasználhatók a dokumentációk²³.

Az szervezeti- és alkalmazásszintű környezeti útmutató gondoskodik arról a keretről, amiben biztosítani lehet, hogy az üzemeltetésre, használatra, az alkalmazásokra vonatkozó szabványokkal, helyi szabályokkal, előírásokkal összhangban készülnek el a képernyő és a jelentés / beszámoló tervek.

²² Ez a "filozófiai" megközelítés az SSADM egyik legfontosabb módszertani megközelítése, nevezetesen az, hogy egy adott 'tervezési termék' több oldalról közelíti meg, más-más szemléletet használva fel a leírásra. Ezzel a megközelítési móddal lehetett találkozni az adatfolyamok racionalizálásakor a folyamat és adatszemplélet ütköztetések, a relációs adatelemzések, a relációs és a fogalmi szemlélet ütköztetések, az entitás viselkedés modellezések az esemény és adatoldal összehangolásakor.

²³ Karbantartás, tovább fejlesztés egy hosszú életciklusú terméknel mint az információrendszer kritikus kérdés. Egy információrendszer átlagos életciklusa több évtizedet foghat át (OECD statisztika 30 év).

7. Dialógustervezés



7.1 Kapcsolata más technikákkal

7.2 Dialógusazonosítás

Dialógusokkal kapcsolatos elemzési feladatokból tevődik össze. Minden lehetséges dialógust azonosítani kell a követelményspecifikációs modul végére.

A dialógusok nem eseményeken, hanem funkciókon és felhasználói szerepkörökön alapulnak, ezért a felhasználó igen fontos szerepet kap a dialógusazonosítás folyamán.

Az elkészített dokumentáció a specifikációs prototípus készítése, a rendszertechnikai alternatívák megfogalmazása és a dialógustervezés során közvetlenül felhasználásra kerül.

7.2.1 Eljárás lépései

7.2.1.1 Felhasználójegyzék készítése²⁴

A végfelhasználókat, beosztásukat és végrehajtandó feladataik leírását tartalmazza a felhasználójegyzék. Kezdetben a jelenlegi rendszeren alapul, majd később az új rendszer követelményeinek figyelembe vételével változik, a potenciális felhasználók listája módosulhat, egészen a rendszerszervezési alternatíva kiválasztásáig, mert ekkor véglegesítődik a rendszer határa.

Felhasználójegyzék létrehozásakor támaszkodni kell az összes szervezetben fellelhető dokumentumra, ami szervezet felépítését tartalmazza, kiegészítve a kulcs pozíciót betöltők véleményével.

7.2.1.2 Szerepkörök azonosítása²⁵

A szerepkörök a felhasználójegyzékből származnak, és olyan munkavégzők csoportjaként határozható meg, akik nagy számú közös feladaton osztoznak, ugyanaz a munkaköri leírásuk. Azon személyek összevonása egy csoportba, akik ugyanazt a funkciót használják, elkerülhetővé teszi a dialógusok felesleges megismétlését.

További szerepkörök elkülönítése válhat szükségessé biztonsági megfontolások következtében.

A szerepkörök meghatározásának forrásai:

- jelenlegi fizikai adatfolyam modell (folyamat helye);
- igényelt rendszer adatfolyam modellje (szervezetben belüli, de rendszeren kívüli külső entitások);
- szervezeti hierarchiában elfoglalt hely, és információ hozzáférési jogosultságok (főnök-beosztott jogainak különbsége különböző szerepkörökhöz vezethet).

7.2.1.3 A dialógusok szemben támasztott követelmények azonosítása²⁶

Itt a funkciók és szerepkörök összevetése történik a szerepkör-funkció táblázat alapján. A táblázat minden egyes kitöltött rubrikája dialógust határoz meg — a funkciók indításának interaktív igényét jelezve —, így a táblázat egy során végighaladva egy adott szerepkör összes dialógusa azonosítható. Ezt a táblázatot a későbbiekben a felhasználó felülvizsgálja. (Az azonosított dialógusok némelyikét lehet, hogy még optimalizálják).

Általában egy funkcióhoz egy dialógus tartozik, de előfordulhat, hogy több szerepkör használja ugyanazt a funkciót, ekkor szükség lehet több dialógus kialakítására is.

²⁴ (120-es lépés)

²⁵ (310-es lépés)

²⁶ (330-as lépés)

7.2.1.4 A kritikus dialógusok azonosítása²⁷

A kritikus dialógusok meghatározásnak az a célja, hogy a dialógusok teljes listájából a rendszer szempontjából legfontosabb, leglényegesebb, a rendszer sikere szempontjából kritikusnak tekintett dialógusok azonosítsák.

A kritikus dialógusok azért kerülnek meghatározásra ebben a szakaszban, hogy a prototípus-készítés megtörténhessen, és hogy elősegítse a rendszertechnikai alternatívák kiértékelését:

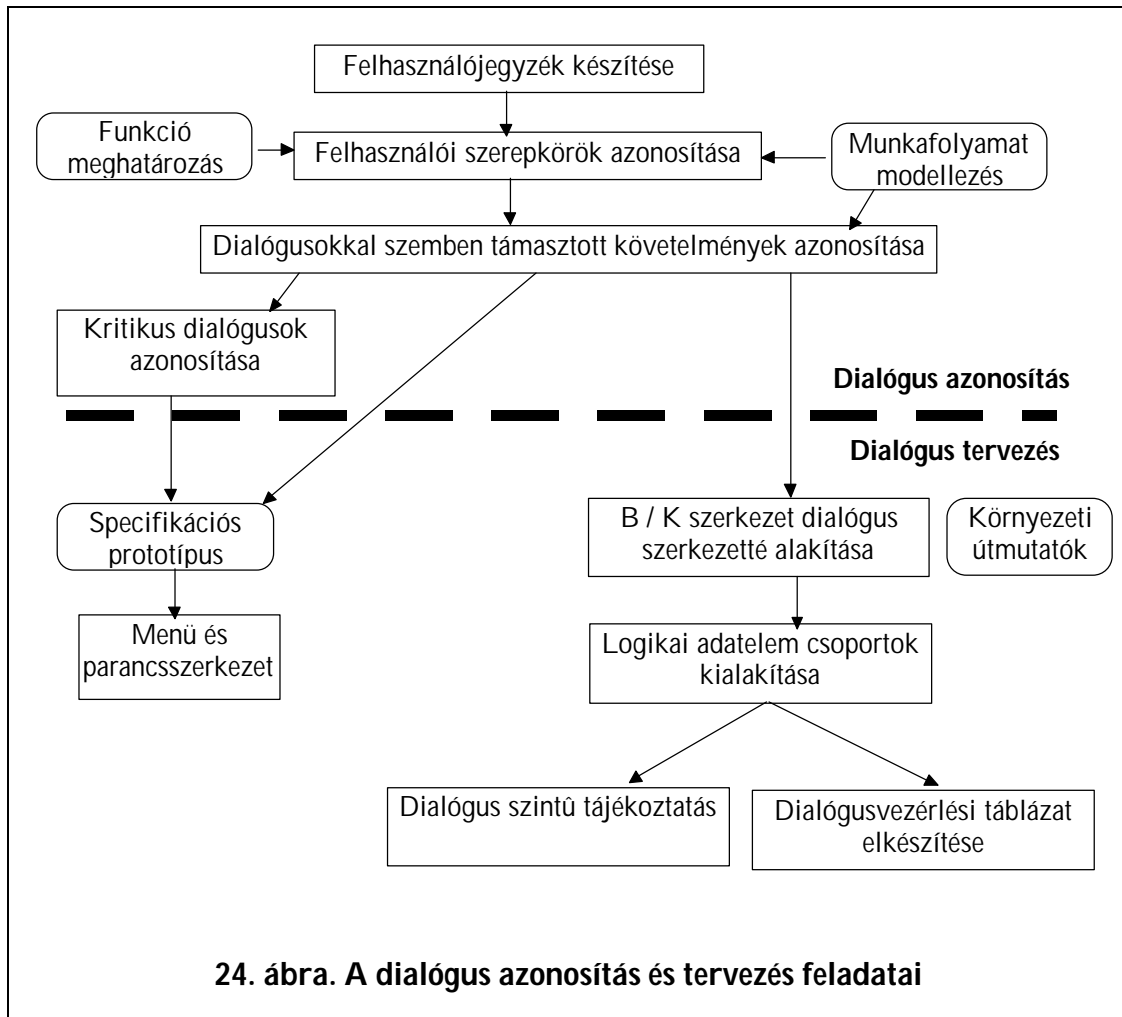
- a fontosságuk és bonyolultságuk miatt prototípust kell ezekre a dialógusokra készíteni;
- a teljesítmény tervezéskor a jelentős volumenű dialógusok mennyiségi adatai felhasználhatók a rendszer teljesítmény becslésére, tervezés alatti hangolásra, a megfelelő rendszertechnikai alternatíva kiválasztására.

A szerepkör-funkció táblázatban az 'x' helyére 'c' írandó jelezve, hogy a dialógus kritikus.

Kiválasztási szempontok:

- A felhasználó kritikus fontosságúnak tekinti-e?
- A dialógus kapcsolódó funkció kritikus fontosságú-e a szervezet, az üzlet szempontjából?
- Megváltoztatja-e a végfelhasználó feladatai végrehajtásának a módját?
- Sok felhasználó használja ugyanazt a dialógust?
- Nagyon gyakran használt dialógus?
- A dialógus sok entitást használ fel a logikai adatmodellből?
- Sok adatelem kapcsolódik a dialógushoz?
- A dialógushoz kapcsolódó navigációs igények bonyolultak-e?
- Szükség van-e bonyolult hibakezelésre, tájékoztató és segítő információkra?

²⁷ (330-as lépés)



7.2.2 Dialógustervezés

Ez a dialógusazonosítást követő tervezési feladatokból áll, és a létrejövő dokumentáció egy a dialóguselemekig lebomló, strukturált jelölésrendszer segítségével készül el.

7.2.3 Eljárás

7.2.3.1 A B/K adatszerkezet átalakítása²⁸

Az interaktív funkciók leírásaiban található B/K adatszerkezetek jelentik a dialógustervezés kiindulási pontját. Ezeket a dialógustervezés dialógusszerkezet ábrákká alakítja át.

Ezzel egyidejűleg dialóguselem-leírások készülnek, amelyek dialóguselemek és a hozzájuk kapcsolódó adatelemek formájában tartalmazzák azt az információt, ami a B/K adatszerkezetek kiegészítő dokumentációjából származik.

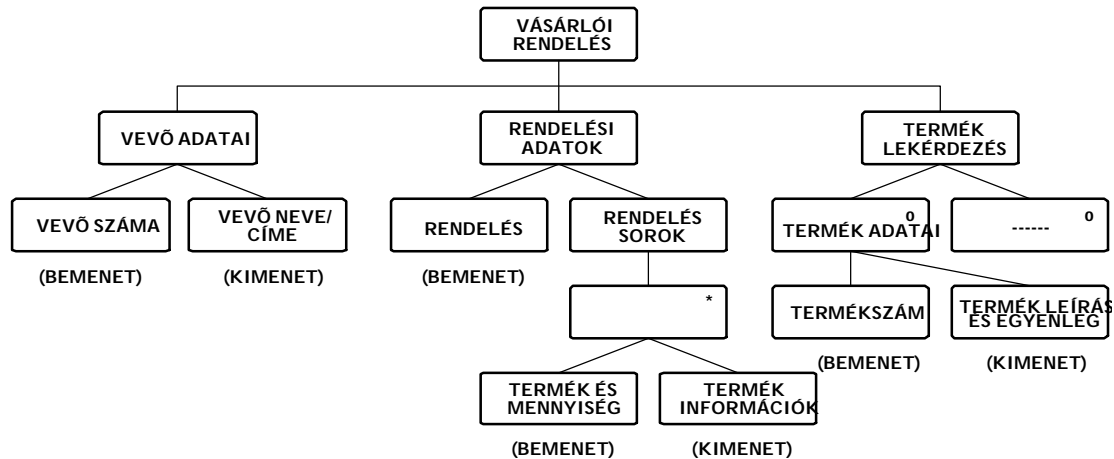
A B/K adatszerkezeteket a következőkkel egészítik ki az átalakítás folyamán:

- a felhasználó és rendszer közti párbeszédhez társuló üzenetekkel;

²⁸ (510-es lépés)

- a dialógus elemekhez a funkciók adatfeldolgozási műveleteit hozzá kell kapcsolni;
- a szelekciók és iterációkhoz a végrehajtási feltételek hozzákapcsolása.

Az alkalmazás szintű környezeti útmutató meghatározhatja vagy befolyásolhatja a dialógusok adatbeviteli fázisainak kialakítását, egyszerre kell az összes adatot bevinni, vagy közben is kap a felhasználó visszacsatolást.



25. ábra. Példa B / K szerkezetre

7.2.3.2 Dialóguselemek logikai csoportjának azonosítása²⁹

A dialóguselemek logikai csoportjait (LGDE) azért hozzák létre, hogy a dialóguson belüli navigálás, közlekedést meg lehessen fogalmazni. Kialakításuk a felhasználóval folytatott megbeszéléseken keresztül, a dialógusszerkezeti ábrák felhasználásával történik. Ha egy felhasználói tevékenység logikailag egy másikat követ akkor, célszerű összecsoportosítani őket.

Egy LGDE több fizikai képernyőre vonatkozhat, ill. egy fizikai képernyő több LGDE-re is vonatkozhat. Az LGDE-k a dialógusszerkezeti ábrákra képződnek le, és általában mind-egyiknek van be- ill. kimenete, tulajdonképpen ilyen "párokból" áll, hacsak nem kell nagy mennyiségű bemeneti adattal foglalkoznia.

Az alkalmazás szintű környezeti útmutató orientálhat az LGDE-k, logikai csoportok megállapításában, a dialógus stílusok előírásán keresztül.

Az LGDE azonosítóját feljegyzik a dialóguselem-leírásokon.

7.2.4 Dialógusvezérlő táblázat létrehozása

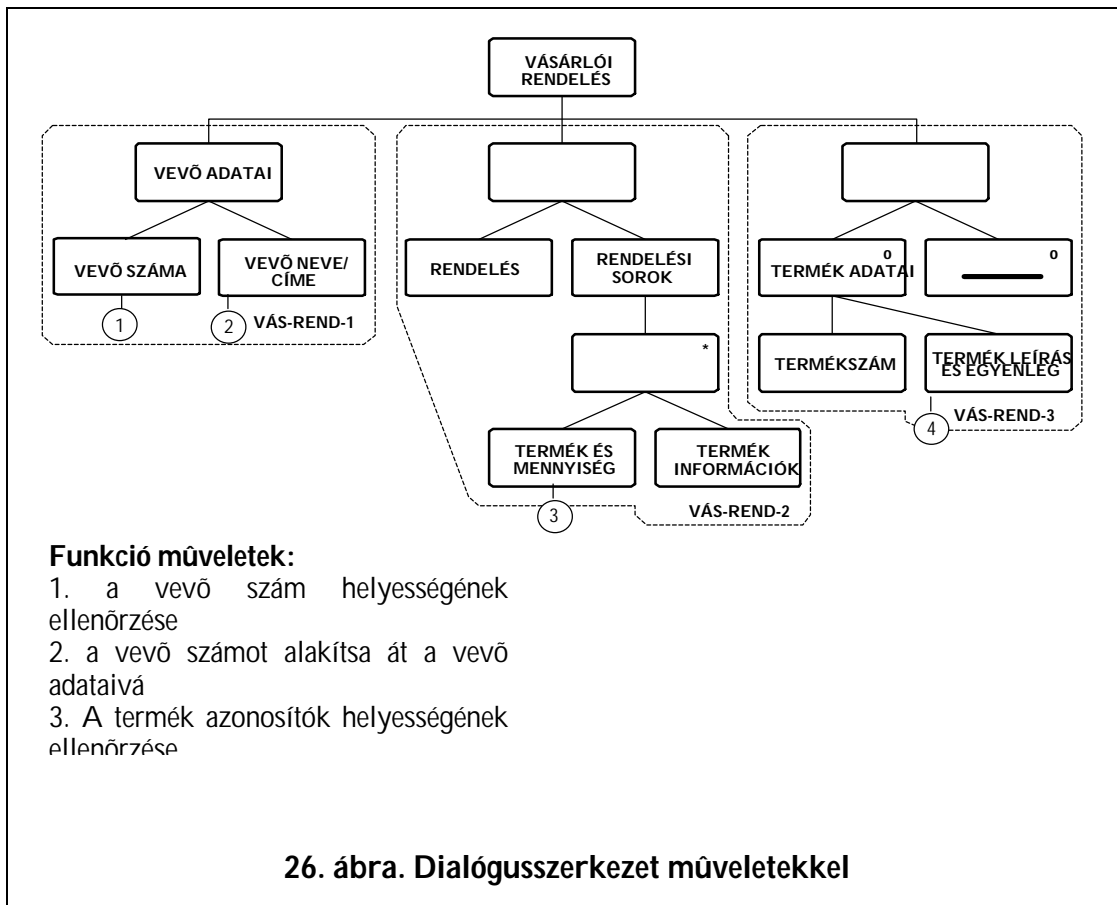
Ez a feladat nem kötelező, opcionális. Dialógusvezérlő táblázat készítése csak nagyon bonyolult navigálást igénylő dialógusokra szükséges. Az alkalmazás szintű környezeti útmutató segíthet ezt a kérdést eldönteni. A felhasználók kívánhatnak nagyon merev dialógus szerke-

²⁹ (510-es lépés)

zetet, de követelhetnek teljes rugalmasságot, kötöttség mentes közlekedést a dialógusok belül és kívül.

Dialógusvezérlő táblázat segítségével lehet azon navigációs utakat részletesen leírni, amelyeket a dialógusban követni lehet az egyes LGDE-ken keresztül. Az alábbi táblázat mind az alap utat, mind az összes alternatív utat mutatja.

Az LGDE-ket kötelező és opcionális jelleg szerint is osztályozzák.



| DIALÓGUS-ELEM | ADATELEM | LGDE AZONOSÍTÓ | KÖTELEZŐ/OPCIONÁLIS |
|---------------------------|-------------------------------------|----------------|---------------------|
| VEVŐ SZÁMA | VEVŐ SZÁMA | VEVŐ-REND-1 | K |
| VEVŐ NÉV/ CÍM | VEVŐ NEVE VEVŐ CÍME VEVŐ TEL. | | |
| RENDELÉS | RENDELÉSI SZÁM REND. DÁTUM | VEVŐ-REND-2 | K |
| TERM. SZÁM/ MENNY. | RÉSZ SZÁM RENDELTE MENNY. | | |
| TERM. LEÍRÁS | TERM. LEÍRÁS TERM. EGYENL. | | |
| TERM. SZÁM | TERM. SZÁM | VEVŐ-REND-3 | O |
| TERM. LEÍRÁS/ EGYENLEG | TERMÉK LEÍRÁS R/V EGYENL. | | |

7.2.5 Menüszerkezetek és parancsszerkezetek megtervezése

A szerepkör-funkció táblázatból kiindulva lehet a menüszerkezeteket elkészíteni. A funkciókat szerepkörök szerint csoportosítják, és ezek alkotják a menüszerkezet hierarchiájának legalsó szintjét. A menüszerkezeteket alulról felfelé haladva alakítják ki. A felhasználó szerint logikailag összetartozó dialógusokat ugyanabba a menüfa ágba célszerű elhelyezni. Ugyanaz a dialógus a menüfa több helyén is felbukkanhat.

Az parancsszerkezetek azt az utat mondják meg, amit a vezérlés követhet, amikor egy dialógus befejeződik. Ez tartalmazza vezérlés átadását egy menünek, valamint az átugrást egy dialógusból közvetlenül egy másikba. Általában minden azonosított dialógusra lesz egy parancsszerkezet.

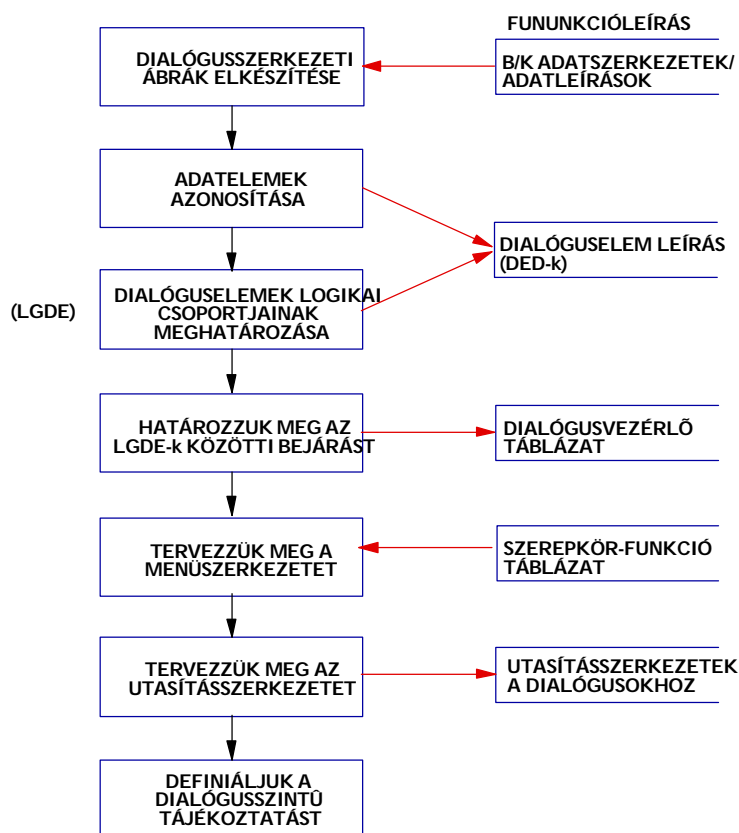
| Menüpont | Dialógus vagy menü | Dialógus vagy menü neve |
|--------------------------|--------------------|-----------------------------|
| Újabb rendelés felvétele | Dialógus | Rendelésfelvétel |
| Kilépés a menüből | Menü | Rendelésfeldolgozás és menü |
| Információk vevőkről | Dialógus | Vevőlekérdezés |
| Kilépés a dialógusból | Menü | Fő alkalmazási menü |
| stb ... | | |

A fizikai funkciógombok részletezése és az egyes dialógus navigálási tevékenységekhez rendelése általában a fizikai tervezésig halasztódik, de minden ismert hardvermegszorítást figyelembe kell venni.

7.2.6 Dialógus közbeni tájékoztatás, segítő információk

A dialógus közbeni tájékoztatás magasabb szinten van, mint a képernyő szokásos tájékoztató lehetőségei, és három alapvető területre lehet bontani:

- Hol vagyok a menüszervezetben?
- Mit kell a következőkben tennem?
- Hogy kerülhetek ki innen?



7.3 Grafikus felhasználói felület³⁰ alkalmazása

7.3.1 Munkafolyamat modell

A párhuzamos vagy konkurens dialógusok alkalmazásának vizsgálata nem dialógus tervezési probléma, hanem a munkaszervezési kérdés:

- Milyen mértékben lehet ugyanakkor a feladatnak olyan különböző komponenseit informatikailag támogatni, amelyeknek a lezárása, befejezése a felhasználóra van bízva?

- A szervezet folyamatait tekintve, vannak-e olyan feladatok, amelyeket a felhasználónak párhuzamosan kell végrehajtania?
- Egy bizonyos feladatot kell-e a felhasználónak több példányban, párhuzamosan egy időben végrehajtania?

7.3.2 Dialógus tervezés grafikus felhasználói felületre

- Induljanak ki a B / K szerkezetből;
- Jelöljék ki a párhuzamos és újra felhasználható részeket a B / K szerkezetből;
- Határozzák meg a dialógusok közötti szinkronizációs pontokat;
- Határozzák meg az aldialógusok függőségét a fődialógustól:
- vajon a fődialógus vagy a felhasználó vezérli-e az aldialógust vagy önálló független dialógusként jelenik meg időnként.
- A megjelenítés módjának meghatározása
- A szervezeti szintű környezeti útmutató tartalmazhat előírásokat a megjelenítés mikéntjére.

8. Fogalmi folyamat modellezés

A fogalmi folyamat modellek kialakítása lényegében logikai adatkezelő folyamatok megtervezését jelenti, ennek a tervezésének technikája két célt szolgál.

- A követelményspecifikálás során összegyűjtött információt részletes Logikai specifikációvá alakítja át, amit majd Fizikai tervvé lehet fejleszteni.
- A rendszer olyan Logikai specifikációját hozza létre, amit fel lehet használni a karbantartás során felmerülő tovább fejlesztési igények következményeinek felmérésére, még azelőtt, hogy a fizikai megvalósításra sor kerülne.

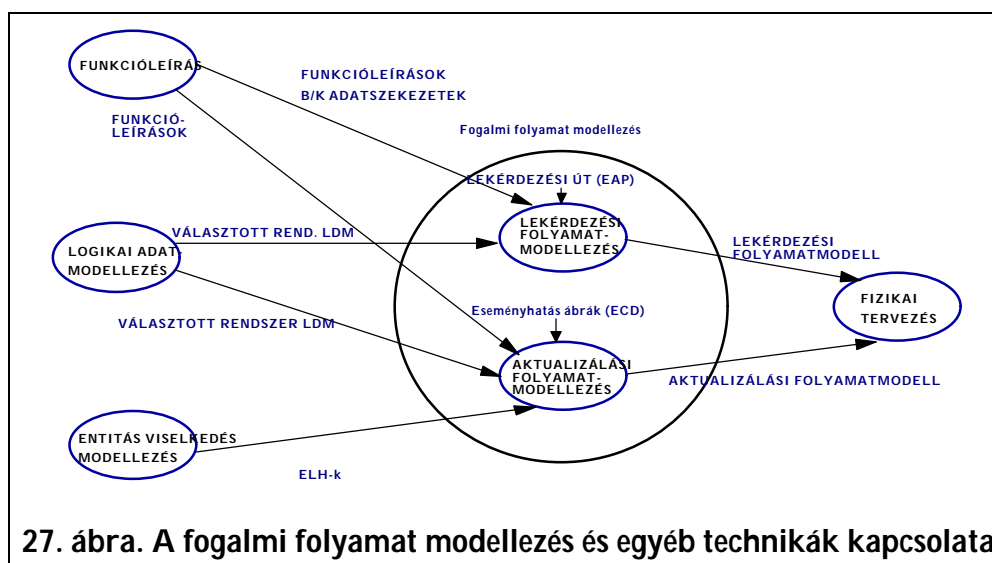
Az fogalmi folyamat modellezés meghatározza a logikai adatbázisba bekerülő adatok feldolgozásának ill. az abból kijövő adatok előállításának folyamatát.

Az fogalmi folyamat modellezés során részletes folyamatmodellek készülnek a lekérdezési funkciókra, és minden aktualizálási funkcióban szereplő eseményre.

A lekérdezési folyamatmodellek származtatása a lekérdezési utakból³¹ történik.

Az aktualizálási folyamatmodellek az entitástörténet-ábrákból és az eseményhatás-ábrákból származtatandók le, amelyek a rendszer adatközpontú ill. eseményközpontú nézőpontját reprezentálják.

A fogalmi folyamat modellezésnek ezeket a technikáit általában csak akkor kell használni, ha a megvalósítás valamilyen harmadik generációs környezetre készül. Ha korszerűbb, negyedik generációs környezetre készül az alkalmazás akkor elegendő az eseményhatás ábrákat és a lekérdezési utakat elkészíteni, ezekből már a programozók / programtervezők közvetlenül el tudják a program kódot készíteni egy 4GL eszközben.



27. ábra. A fogalmi folyamat modellezés és egyéb technikák kapcsolata

³¹ Ez eltér az előző SSADM változattól abban, hogy nem használja funkció meghatározás során elkészített B/K adatszerkezeteket

8.1 Kapcsolat más technikákkal

8.2 Eljárás lekérdezési folyamatmodellek (EPM) készítésére

Mindegyik lekérdezési utat át lehet alakítani lekérdezési folyamat modellé. Annak eldöntése, hogy erre egyáltalán szükség van-e és ha igen melyeket kell átalakítani, ez projektvezetés feladata (projektirányító, modul / szakaszirányító, projekt biztosító csoport).

A 3. szakasz során elő kellett állítani a lekérdezési utakat.

- Az EAP (lekérdezési út) adateléréseinek csoportosítása;
 - Az EAP-n az egy-egyértelmű adateléréseket, adat-hozzáférési megfeleltetéseket össze kell vonni egy csoportba.
- Át kell alakítani a Jackson-féle jelölésrendszerbe;
- A Jackson-féle jelölésrendszer használatával az összevont adateléréseket és a megmaradó egyedi elemeket egy új struktúrában kell megrajzolni.
- Össze kell állítani a műveletek listáját, és a műveleteket az ábra megfelelő részeihez kell kapcsolni;
 - Ezeket a lekérdezési utakról kell átmásolni.
- A műveletek előfeltételeket meg kell fogalmazni és az ábra megfelelő részeihez kell kapcsolni;
 - A szelekciók és az iterációk ciklus feltételét kell meghatározni.
- Le kell ellenőrizni, hogy a kapott ábra szerkezet értelmes-e.

8.3 Eljárás aktualizálási folyamatmodellek (UPM) készítésére

A 3. szakaszban az entitás viselkedés elemzés során el kellett készíteni az eseményhatás ábrát.

- Össze kell vonni az egy-egy megfelelésben lévő hatásokat;
 - Egy téglalapot kell húzni azok köré a hatások köré, amelyek egy-egy megfelelésben állnak, és az így kapott csoportot névvel kell ellátni. A megmaradt egyedi elemek szintén nevet kapnak, és a folyamat struktúra felépítése közben különálló komponensként kezelendők.
- A Jackson-féle jelölésrendszerre történő átalakítás;
- A műveletek felsorolása és az ábra szerkezetének megfelelő elemeihez kapcsolása;
A műveletek listáját össze kell állítani, és hasznos entitásokként csoportosítani ezeket a későbbi ellenőrzés megkönnyítésére.

A sorrend, ahogy a műveleteket figyelembe kell venni az alábbi:

- Műveletek, amelyek entitásokat olvasnak;
- Műveletek, amelyek hibát jeleznek (SI aktuális értéke érvénytelen az élet adott pontján);
- Műveletek, amelyek entitásokat hoznak létre ('create');
- Műveletek, amelyek kibővítik az ELH műveletek hatásait. Ezeket a műveleteket ki kell bővíteni, mivel a cél entitást itt nem lehet megállapítani az ELH ábra kontextusának hiánya miatt. Az ELH műveletek alábbi kibővítéseire van szükség. Az eredeti műveletek dőlt betűvel vannak jelezve (a [szögletes zárójelbe] tett szintaktikus egység elhagyható) :
 - *Create* <entitás>
 - *Set* <attribútum> of <entitás> [*using* <kifejezés>]
 - *Set* <attribútum> of <entitás>
 - *Tie* <entitás> to <főentitás> [*using* <kapcsolat>]
 - *Cut* <entitás> from <főentitás> [*using* <kapcsolat>]
 - Műveletek, amelyek állapotjelző értékét állítják;
 - Műveletek, amelyek kiírják az entitás példányokat az adatbázisba, vagy törlik (delete / write)

A "Gain" és "Lose" műveleteket el kell hagyni.

Minden olvasási műveletre lennie kell egy olyan műveletnek, ami hibát jelez az állapotjelző érvénytelen értéke esetén kivéve, ha minden állapotérték érvényes a hatásra.

Ha az állapotjelző értéke, vagy az entitás bármilyen attribútuma megváltozik, akkor lennie kell egy kiírási műveletnek erre az entitásra.

A lehetséges műveletek listája megegyezik a az entitástörténetben (ELH) és az eseményhatás ábrákon használt műveletekkel (ECD).

- A műveletek elhelyezése a struktúrán
- Helyezzük el a műveleteket a struktúra megfelelő csomópontjára abban a sorrendben, ahogy azokat végre kellene hajtani.
- A műveletek rátehetők a folyamatelemekhez és nem a struktúra elemekhez kapcsolhatók, valamint az szelekció és az iteráló, ciklikus elemekre, de - az utóbbi esetben - nem a felettük lévő struktúraelemekre, amelyek csak a Jackson-féle szintaktikus szabályok miatt jönnek létre és sem adat sem folyamat tartalmuk nincs.
- Adjuk meg a műveletek előfeltételeit a struktúrán
 - Helyezzünk el feltételt minden szelekcióhoz, választási konstrukcióhoz és minden ciklikus komponens fölé.

Integritási hibafeltételek megadása, ha ezeket az eseményhatás elemzés még nem tárta fel:

Az entitástörténet-elemzés eredményeképp elvben minden integritási hibát meg lehet adni az esemény által érintett entitások érvényes megelőző állapotainak segítségével. A gyakorlatban egy-két integritási hiba elsikkadhat az entitástörténet-elemzés folyamán, és ezeket a logikai adatkezelő folyamat tervezése során kell a többihez hozzávenni, mint pl. egy attribútum érvényes értékészletére történő tesztelés (pl. fail if 'jelenlegi adósság' + 'hitelkérelem' túllépi a 'hitelkeret'-et). Bizonyos szabályok lehet, hogy a kiválasztott rendszer logikai adatmodelljében adatelemek értéktartományára vonatkozó szabályok formájában már dokumentáltak.

Minden integritássértési feltételnek meg kell jelenni a műveletek között a megfelelő entitás beolvasása után.

A hibakimenetek megadása:

Minden integritási hibára szabványos kódokat és hibaüzeneteket kell meghatározni.

A struktúra felülvizsgálata:

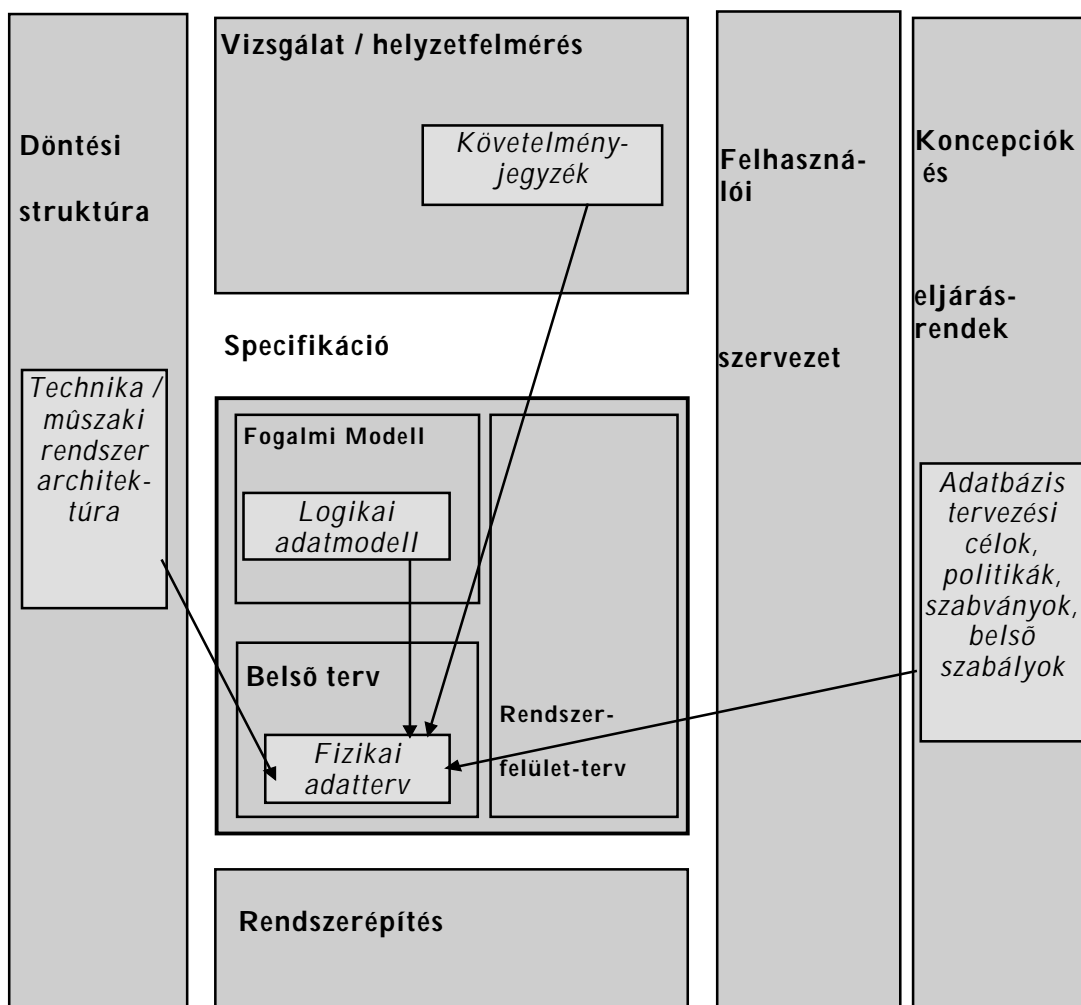
Minőségellenőrzés minden, hibafeltárásban érdekelt fél bevonásával.

9. A fizikai tervezés az SSADM 4+

9.1 Fizikai adattervezés³²

A fizikai adattervezés az adatok fizikai elhelyezésével és ennek az elhelyezkedésnek az adatok elérésére gyakorolt hatásával foglalkozik. A fizikai adattervezés speciális szakértelmet igényel, a szóba jövő adatbázis kezelő rendszerek mély ismeretét, és egyéb algoritmusok és matematika modellek biztos kezelését, amelyek a tervezési döntések kiértékelését segítik a rendszer által nyújtott teljesítmény szempontjából.

Csak egy SSADM projekt végrehajtásához szükséges korlátozott ismeretek tartoznak az SSADM módszertan keretébe és a fejlesztők két csoportját célozza meg ez a szabály halmaz, akik esetleg ezeket alkalmazhatnák:



28. ábra. Fizikai adattervezés helye a rendszerfejlesztési alapmintában

³² [CCTA95], [CCTA95A], Reference Manual Part 9: Physical Design, Physical Data Design, 9-19—9-40, User Guide Part 6: Physical Design, 6-5—6-51. Továbbá [CCAT90].

- SSADM rendszerelemzőket, akiknek a felelőssége, hogy az adatbázis tervezőket ellássák a szükséges ismeretekkel, azaz az adattervre vonatkozó követelményeket kell közölniük.
- az adatbázis tervezőket és az adatbázis adminisztrátort³³, akiknek tudniuk kell azt, hogy hogyan alkalmazzák az SSADM termékeket. Ismerniük kell a folyamat-adat kapcsolat jelentőségét, valamint a fizikai adat és folyamat tervezés összehangolásának szükségességét.

A fizikai adattervezés célja az, hogy egy olyan tervet hozzon létre:

- amely, az igényelt rendszer logikai adatmodelljét valósítja, amely tartalmazza az új rendszerrel szemben támasztott adatkövetelményeket;
- illeszkedik a rendszer adatfeldolgozási igényeihez;
- kielégíti a rendszerrel szemben támasztott háttértár kapacitásra és válaszidőkre vonatkozó korlátokat.

9.1.1 A fizikai adattervezés során figyelembe vett szempontok

A fizikai adatterv kialakítása során a következő kérdésekkel kell foglalkozni:

- a logikai adatmodellből egy hatékony adatbázis tervet kell létrehozni, ezért a tervezőnek alaposan ismernie kell a megvalósítási környezetet;
- a tervezőnek tudnia kell azt is, hogy milyen mértékben optimalizálja a tervet. A cél az, hogy olyan terv készüljön, amely a rendszer teljesítményre vonatkozó célokat a eléggé jól megközelíti;
- a rendszer teljesítményre vonatkozó követelmények általában konfliktusban vannak egyéb nem funkcionális követelményekkel, mint például a karbantarthatóság, fejlesztési idő, rendelkezésre álló háttértár kapacitás.

A fejlesztő csoport számára az SSADM egy olyan általános keretet nyújt, amelyben ezeket a feladatokat meg tudják oldani. Természetesen az adott környezethez, és ismeretekhez lehet igazítani a javasolt megközelítést:

- határozzák meg a fizikai adattervezés során követett stratégiát;
- állítsák elő a fizikai adattervet;
- ellenőrizzék, teszteljék, majd optimalizálják a fizikai adattervet azokon a pontokon, ahol nem éri el a teljesítmény célkitűzéseket.

³³ Az adatbázis adminisztrátor egy projekttől független szervezethez tartozik, amelyik az adatgazdálkodásért, -managementért felel. További idetartozó szerepkörök: adat adminisztrátor, adatszótár adminisztrátor.

9.1.2 A fizikai adattervezés termékei

A fizikai adattervezés végterméke a fizikai adatterv. Ennek formáját, a helyi eljárások, szabályok és szabványok valamint a megvalósítási környezet igényei szabják meg. A fizikai adattervezés során több közbenső termék keletkezik, nevezetesen:

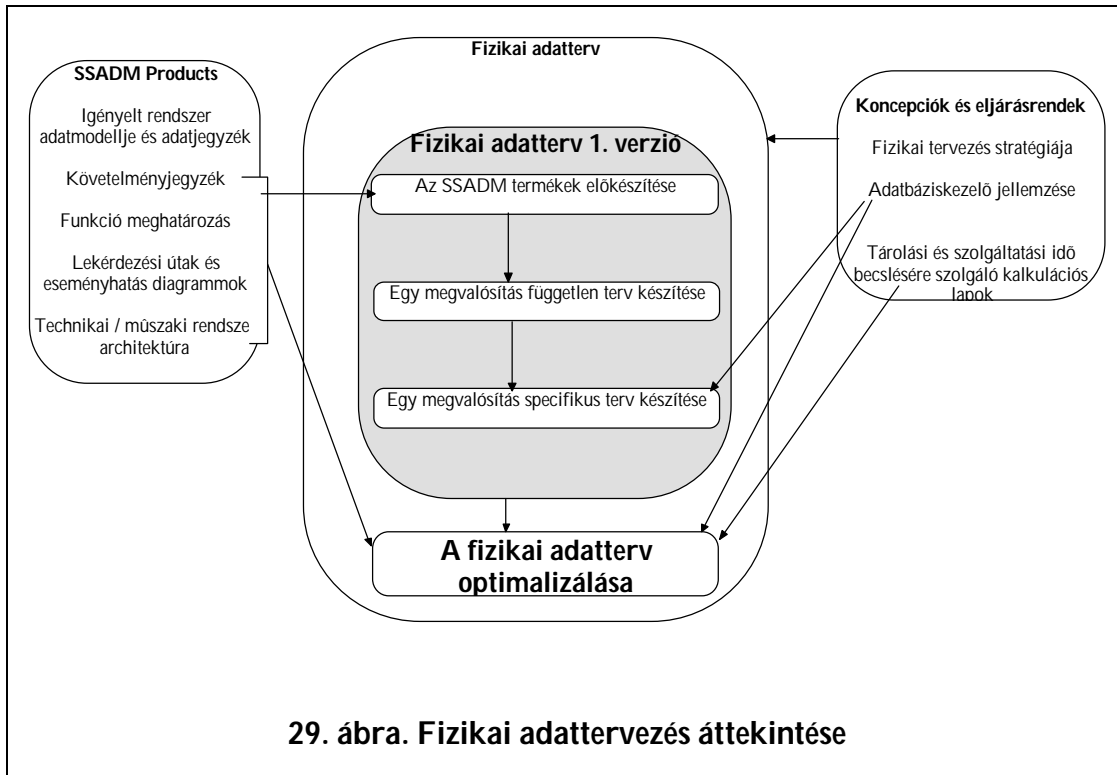
- **Fizikai tervezés stratégiája.** Ahol egy új adatbáziskezelő rendszert vezetnek be, vagy ez az első SSADM stílusban végrehajtott projekt, amelynek során ebben a bizonyos adatbázis-kezelőben készítik el az alkalmazást, akkor célszerű lehet egy olyan stratégia kialakítása, amelyet a fizikai adattervezés során követnek. A legtöbb szervezetnél az adatbázis tervezés módszerét a szervezet saját maga alakítja ki.
- **Adatbáziskezelő jellemzése.** Az adatbázis tulajdonságainak, szolgáltatásai sajátosságainak leírásából elkészíthető az adatbáziskezelő tárolási kapacitásának és általa nyújtott teljesítmény paramétereknek a jellemzése.
- **Tárolási és szolgáltatási idő becslésére szolgáló kalkulációs lapok**³⁴. Ezeket annak a durva ellenőrzésére lehet alkalmazni, hogy egyáltalán a fizikai terv megközelíti a háttértár kapacitásra és a válaszidőkre vonatkozó korlátokat. Ezeket az adatokat a terv optimalizálása során lehet felhasználni³⁵.

9.1.3 Áttekintés a fizikai adattervezésről

Az igényelt rendszer logikai adatmodellje egy megvalósítástól független specifikáció. Amikor, a megvalósítás eszközeit kiválasztják, akkor a logikai adatmodellt át kell alakítani adatbázis tervvé vagy állomány tervvé, amely az adott eszköz halmaz tárolási tulajdonságai és adatelérési szolgáltatásai segítségével fogalmazza meg az adattervet.

³⁴ Akiket esetleg érdekel az, hogy ezeken az egyszerű és triviális számításokon kívül milyen módon lehet pontosabban modellezni a szoftver- hardver konfiguráció teljesítmény viszonyait azoknak az irodalom jegyzékben felsorolt szakkönyveket ajánlom figyelmébe. Természetesen ennek a tevékenységnek költségei vannak: idő, pénz és munkaerő tekintetében. [Allen78], [Kant92], [Lazowska84], [Menascé94], [Molnár95], [CCTA90A], [David92], [Johnson91].

³⁵ Ez persze csak meglehetősen durva közelítést ad, és még gyakorlatilag is használhatatlan eredményre vezet, ha ugyanazon a konfiguráción több alkalmazási rendszer működik. Szintén problémát okozhat az adatok elosztottsága, akár több diszken, vagy hálózatban. A becslés pontosságát tovább rontja, ha a tervezett alkalmazási funkciók között erős konkurens használat áll fenn. Ezeket a tényezőket csak sokkal bonyolultabb algoritmusokkal, és matematikai modellekkel lehet figyelembe venni. A dolog jó oldala viszont, hogy az ilyen modellezéshez szükséges összes bemeneti paramétert az SSADM összegyűjtötte.



9.1.4 A fizikai adatterv első verziójának elkészítése

Ebben a szekcióban leírjuk a logikai adatmodell átalakítását fizikai adattervvé. Ez a terv a választott adatbáziskezelő rendszerben megvalósítható terv, de még annak ellenőrzése nem történt meg, hogy vajon a rendszer funkcióira vonatkozó, a szolgáltatási szint követelményekben rögzített paramétereket a terv teljesíti-e, ezért hívják ezt első verzióknak vagy változatnak.

9.1.4.1 Az adatterv első változatának általános célkitűzései

Az igényelt rendszer logikai adatmodellje nem tartalmazza azoknak a fizikai adatállományoknak a leírását, amelyek az entitásokat tárolnák; az entitás példányok mennyiségi adataitól eltekintve nincsenek fizikai vagy teljesítmény paraméterek rögzítve a modellben.

A fizikai adattervezés fő célja az, hogy létrehozzon egy olyan fizikai adattervet, amely:

- amely lehetővé teszi, hogy a teljesítmény paraméter előrejelzések a valóságot jól közelítsék;
- gyorsan készüljön el és egyszerű ökol szabályokat használjon;
- minimalizálni fogja a programozási költségeket.

Ezeket a célokat az igényelt rendszer adatmodelljén végrehajtott átalakítások sorozatával lehet elérni³⁶.

9.1.4.2 Az adatbáziskezelő feltételezett tulajdonságai

A szóba jöhető adatbáziskezelők között felfedezhetők olyan általános elvek, tulajdonságok, amelyek az eszközök széles körére igazak, ezeket a tulajdonságokat adottnak véve egy olyan fizikai adattervet, illetve annak első verzióját lehet létrehozni, amely az adatbáziskezelők nagy részében alkalmazható lesz.

9.1.4.2.1 Az entitások rekord típusként jelennek meg

A logikai adat visszakeresés alapegysége a rekord. Minden entitás előfordulás (az attribútumaival együtt) rekordok formájában jelenik meg az adatbázisban (az adatelemeivel együtt). Továbbá, az entitás nézetek³⁷ vagy megjelenések előfordulásai különböző táblázatokba (relációkban) kerülhetnek rekordként megvalósításra vagy egy adott táblának különböző nézeteiként (view).

9.1.4.2.2 A rekordokat blokkokban tároljuk

A fizikai adat elérés egysége a blokk (vagy adatbázis lap). A legtöbb adatbáziskezelő rendszernek van valami olyan szolgáltatása, amely segíti a rekordok elhelyezésének befolyásolását a blokkokon belül. Egy adott blokkon belül elhelyezkedő rekordok elérése lényegesen gyorsabb mint azoké, amelyek különböző blokkokban vannak. Ezenkívül a blokkok mérete úgy szabályozható, hogy a rekordok méretének és számának növekedését be tudja fogadni.

9.1.4.2.3 A rekordokból fizikai csoportokat hoznak létre

Azoknak az adatoknak a csoportját, amelyeket gyakran együtt használnak vagy nagyon gyorsan kell elérni és visszakeresni célszerű együtt, egy helyen tárolni. Vagyis, ugyanahhoz a főentitáshoz tartozó alentitásokat össze kell csoportosítani és lehetőleg "fizikailag" is közel kell elhelyezni.

Ez azt jelenti, hogy a fizikai adatterv a főentitások csoportjainak blokkokba helyezésének meghatározását, továbbá a főentitások alentitásainak, majd azok alentitásait, és így tovább, összecsoportosítását és fizikai adatbázis blokkokba helyezését fogja leírni. Ennek a leírási módja egy hierarchikus adatszerkezet lesz, amelyet az igényelt rendszer adatmodelljéből lehet levezetni.³⁸

³⁶ Meg kell jegyezni, hogy az igényelt rendszer adatmodellje ebben a szakaszban is létezik a saját jogán, a fizikai adatterv nem fogja el venni a helyét vagy kiszorítani.

³⁷ entitás aspektus

³⁸ Ez a hierarchikus adatszerkezet még olyan adatbáziskezelők esetében is hasznos, amelyek nem erre a filozófiára épülnek, illetve híveik elborzadnak annak hallatától, hogy a terv hierarchikus, az adatbáziskezelő pedig "tisztá relációs". Ez a leírás a programozóknak és programtervezőknek is segít abban, hogy bele tekintsenek a háttérben meghúzódó gondolatokba.

9.1.4.2.4 A fizikai csoportokon belüli elsődleges kapcsolatok hatékony támogatása szükséges

Definíció 9-1 Elsődleges kapcsolat

Az igényelt rendszer adatmodelljében azon főentitás és az alentitás között létező kapcsolatokat, amelyek ugyanabba a fizikai adatcsoportba kerülnek, *elsődleges kapcsolatnak*, vagy elsődleges fontosságú kapcsolatnak nevezik.

Az adatbáziskezelők széles köre nyújt olyan szolgáltatásokat, amelyek ezeknek a kapcsolatoknak, illetve ezekkel összefüggő adat-visszakeresési feladatoknál hatékony megvalósítási megoldásokat nyújtanak. Erre lehet példa az az utasítás, amely így szól "Olvasd be az adott főentitás példányhoz tartozó következő alentitás példányt"³⁹, ez az utasítás lényegesen hatékonyabban hajtódik végre egy fizikai adatcsoporton belül és kevésbé hatékony eltérő fizikai adatcsoportok között.

9.1.4.2.5 A fizikai csoportokon belüli másodlagos kapcsolatok hatékony támogatása szükséges

Definíció 9-2 Másodlagos kapcsolat

A különböző fizikai adatcsoportok között átvezető kapcsolatokat *másodlagos kapcsolatoknak* nevezik.

Ezeket a kapcsolatokat tipikusan más eszközökkel, megoldásokkal lehet megvalósítani az adatbáziskezelő eszközökben mint az elsődleges kapcsolatokat. Gyakran különleges feltételeknek kell eleget tenni vagy ilyeneket meg kell teremteni, pl. teljes index tábla létrehozására lehet szükség ott, ahol egyébként valamilyen részleges vagy tömörített tábla is elegendő lenne ahhoz, hogy az adatcsoportot a kulcs szerint összecsoportosítva el lehessen érni ('grouped by key').

9.1.4.3 A fizikai adatterv első verziója elkészítésének lépései

A fizikai adattervet három nagy lépésben javasolja az SSADM elkészíteni:

- az SSADM termékek előkészítése;
- a logikai adatterv implementáció független fizikai tervvé alakítása;
- a fizikai adatterv megvalósítás specifikus megoldássá alakítása;

9.1.4.3.1 Az SSADM termékek átalakítása

Ezt a lépést egy az SSADM-ben járatos rendszerelemző hajtja végre, aki jól ismeri az SSADM termékeket, azok szerkezetét.

³⁹ Read next detail of current master

Mivel a fizikai adatterv egy jól körülhatárolt célt szolgál, ezért az igényelt rendszer adatmodelljében rögzített információk egy részére nincs szükség; a következő különbségek állnak fenn a két adatterv között:

- az igényelt rendszer adatmodelljében használt kapcsolat nevekre nincs szükség a fizikai adattervben⁴⁰;
- a kapcsolatok opcionálításának csupán azt az oldalát kell felhasználni a fizikai adattervben, amelyik megmutatja, hogy vajon egy alentitás példányhoz kell-e egy főentitás példánynak léteznie, vagyis csak az alentitás felől nézve kötelező kapcsolatokat kell annak megállapítására felhasználni, hogy vajon egy közönséges entitás⁴¹ melyik fizikai adatcsoportba kerüljön be;
- a 'kizáró vagy' kapcsolatok megvalósítása nem befolyásolja a fizikai adattervet, mivel az csak az adatok elhelyezésével és elérésével foglalkozik.

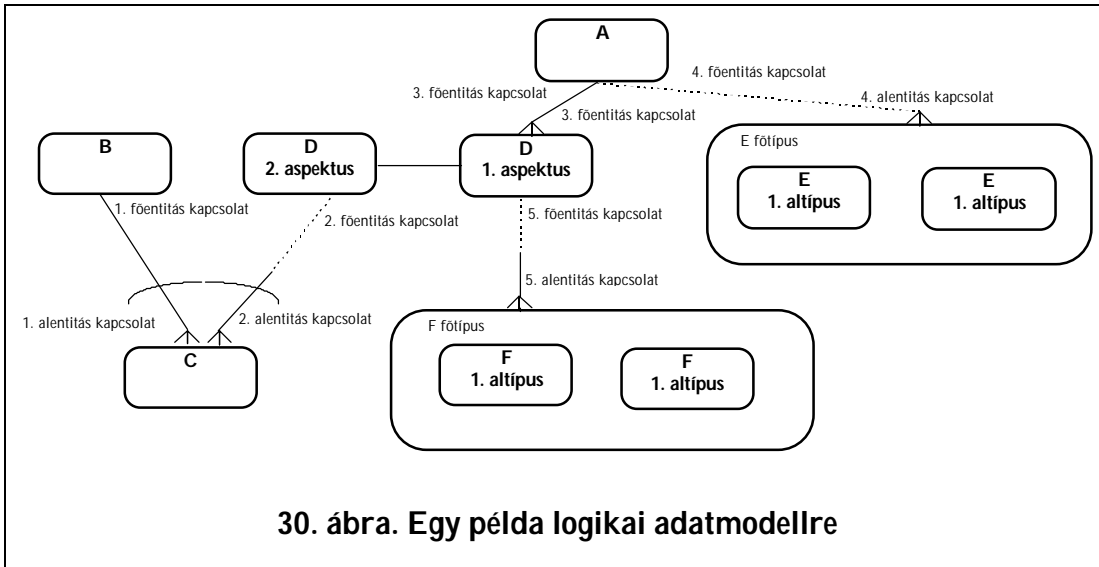
A fentebbiekből az következik, hogy a fizikai adatterv egyszerű kiinduló pontjaként az igényelt rendszer adatmodelljéből előállítható egy termék a következő lépések kivitelezésével:

- kapcsolat neveket eltávolítják;
- ugyanakkor az entitásnak a különböző megjelenéseit, nézeteit (aspektusait), egyetlen rekord típusba gyűjtjük⁴²;
- az altípus / főtípus jellegű hierarchiákat a következőképpen kell feloldani:
- ha az altípusoknak kevés egymástól különböző attribútumaik vannak, akkor célszerű a főtípus rekordjába becsomagolni; az eredmény rekordban természetesen létre kell hozni egy 'altípus' attribútumot, valamint lesznek olyan attribútumok, melyek egymást kölcsönösen kizáró módon kapnak értéket, az 'altípus' attribútum értékétől függően;
- ha az altípusoknak sok egymástól különböző attribútumaik vannak, akkor különböző rekord típusokat hozunk létre, ugyanazzal a kulccsal.
- a kapcsolatok opcionálítását csak az alentitás oldaláról nézve jelölik meg;
- a 'kizáró vagy' -ot jelölő íveket opcionális kapcsolatokkal kell helyettesíteni.

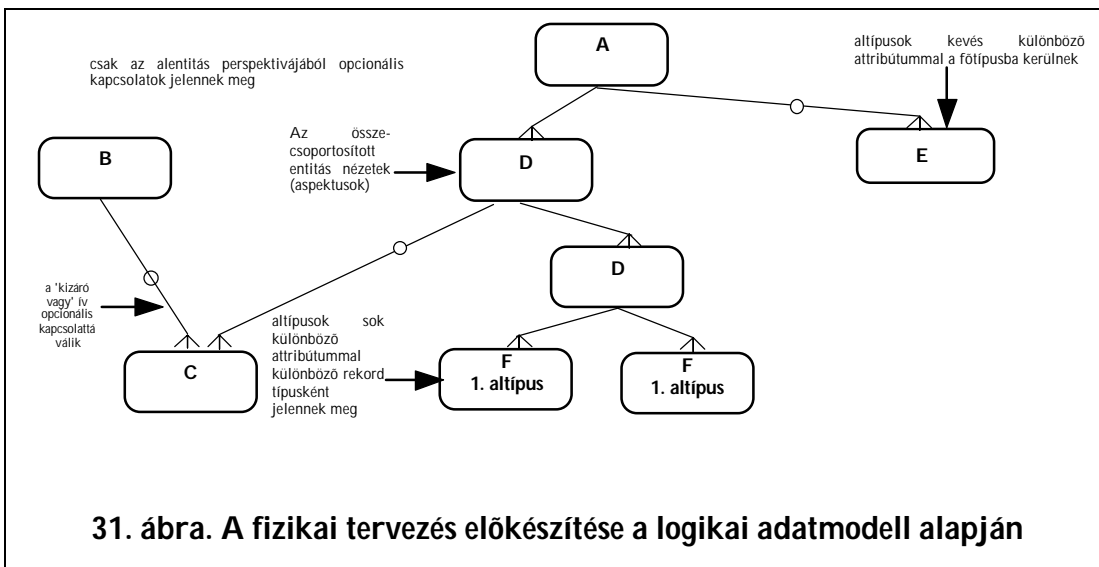
⁴⁰ Azonban előfordulhat az, hogy ún. fizikai kapcsolat neveket kell megadni, mert a megvalósításra szánt adatbázis-kezelőnek szüksége van erre.

⁴¹ Vagyis nem 'gyökér' entitása a fizikai adatcsoportnak.

⁴² Ez alól csak az a kivétel, ha ezek az aspektusok különböző telephelyeken, földrajzilag távol eső, elkülönült helyeken lesznek majd implementálva lsd. [ISE94], *Distributed Systems: Application Development*, ed. R. Duschl, Dr. J. Stewart, authors: Dr. M. Breu, A. Aue, J. Hall, K. Robinson, ISE (Information Systems Engineering) Library, HMSO (Her Majesty Stationary Office) / Siemens Nixdorf, 1994.



A fentebbiekben felsorolt elvek alkalmazásával a logikai adatmodellt (30. ábra.) a következőképpen lehet átalakítani (31. ábra.).



Az igényelt rendszer adatmodelljéből a mennyiségi adatokat át kell menteni a fizikai adattervbe. A mennyiségi adatok alatt azt értjük, hogy az egyes entitásokban hány entitás példány fordulhat elő (ezt tartalmazzák az entitást ábrázoló téglalapok), továbbá mennyi a "függő" előfordulások száma, azaz egy főentitás példányra hány alentitás példány jut (ezek az adatok a csirkelábaknál jelennek meg az ábrán).

A nem kulcs adatelemen keresztül történő belépési pontokat fel kell ismerni és meg kell jelölni. A belépési pontok az eseményhatás ábrákból (ECD) és a lekérdezési utakból ismerhetők fel. A nem kulcson keresztüli belépésekhez meg kell határozni:

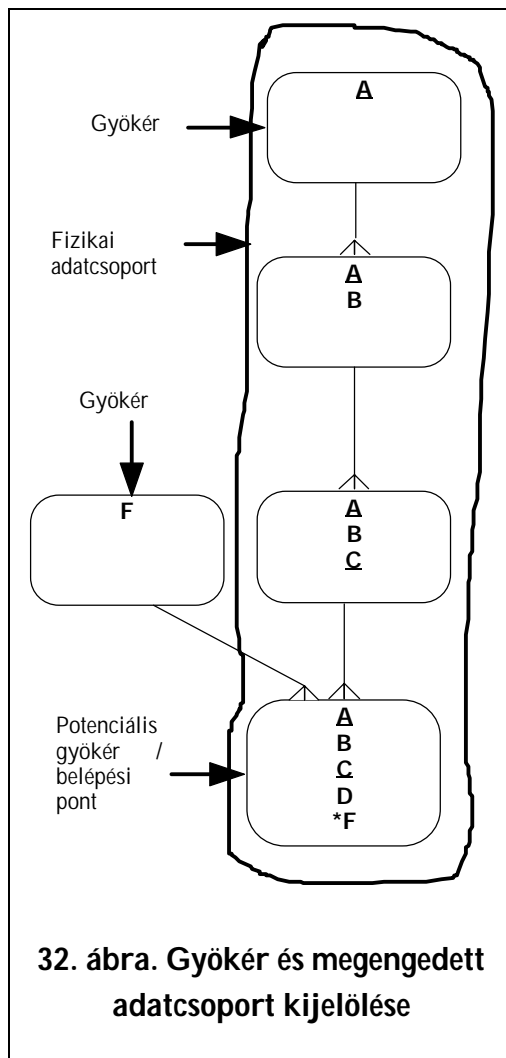
- az adatmodellben adatelérési igényként jelentkező belépési pontokat;
- mindegyik belépési pontnál az adatelemeket össze kell vetni az érintett entitások kulcsainak adatelemeivel azért, hogy a nem kulcs adatelemek egyértelműen kiderülhessenek;

- különböztessük meg az adatmodellbe történő kulcsokon keresztüli és a nem kulcs adat-
elemeken keresztül belépéseket.

9.1.4.3.2 A logikai adatterv implementáció független fizikai tervvé alakítása

Ez egy közbenső átalakítási lépés, amelyet vagy egy SSADM rendszerelemző vagy egy adatbázis tervező / adatbázis adminisztrátor hajt végre. Ezt a lépést a következő lépéssel párhuzamosan is el lehet végezni, amikor is a megvalósítás specifikus változata készül el. Az előző lépésben vázlatos terv azért hasznos, mert könnyen felismerhetők rajta a az ún. "gyökér" entitások, amelyek a fizikai adatcsoportok kristályosodási pontjai lesznek.

Definíció 9-3 Gyökér entitás



A fizikai adatcsoportok gyökér entitásai a következő két típusba tartoznak:

- azok az entitások, amelyeknek már nincsen fő-entitásuk, vagyis nem alentitásai egyetlen más entitásnak sem;
- közvetlen belépési pontként jelölték meg az adatmodellen; ez alól csak a következő eset a kivétel, nevezetesen ha:
- a szóban forgó entitás⁴³ kulcsa része egyik fő-entitás⁴⁴ kulcsának, és
- ez a bizonyos főentitás már a fizikai adatcsoport hierarchia tetején helyezkedik el mint gyökér.

A gyökér entitásokat valahogy meg kell jelölni a fizikai adatterven.

Definíció 9-4 Megengedett fizikai adatcsoportok (hierarchiák)

A nem gyökér entitások hozzárendelődnek bizonyos fizikai adatcsoportokhoz:

- egy nem gyökér entitás csak olyan fizikai adat-

⁴³ A szóban forgó entitás akár hierarchikus akár összetett kulcsa, amely a belépési pontot és adatelemeket jelenti a logikai adatmodellen, tartalmazza valamelyik főentitásának a kulcsát, amelyet más gyökéreként jelöltek meg és így egy fizikai adatcsoport tetején helyezkedik el.

⁴⁴ Ez nem feltétlenül a közvetlen főentitást jelenti, hanem egy entitás hierarchia is lehet!

csoportba kerülhet be, amelynek már tagja valamelyik főentitás, és amelyhez a szóban forgó (al)entitás az alentitás felől nézve kötelező kapcsolaton keresztül kötődik⁴⁵;

- ha egy nem gyökér entitás, közvetlen belépési pont és van választási lehetőség, hogy melyik adatcsoportba kerüljön (vagyis két vagy több olyan főentításhoz tartozhat kötelező kapcsolaton keresztül, amelyek viszont különböző csoportokba tartoznak), akkor abba a csoportba kell elhelyezni, ahol az a főentitás jelenik meg, amelynek kulcsa része az alentitás kulcsának⁴⁶.

Ha két vagy több fizikai adatcsoport is megengedett, akkor ahhoz a csoporthoz kell kapcsolni, amelyiket a "legalacsonyabb függő előfordulások" szabálya jelöli ki.

Definíció 9-5 A legalacsonyabb függő előfordulások szabálya

Ha egy entitást két vagy több megengedett fizikai adatcsoport hierarchiához lehetne csatolni, akkor ahhoz kell kapcsolni, amelyikben a szóban forgó entitásnak mint alentitásnak az előfordulásai alacsonyabbak, vagyis kevesebb példánnyal vesz részt az adatcsoportban a főentitásához vezető kötelező adatcsoporton keresztül.

9.1.4.3.3 A fizikai adatterv megvalósítás specifikus megoldássá alakítása

Ezt a lépést az adatbázis tervező / adatbázis adminisztrátor fogja végrehajtani. A tervezőnek dönteni kell a következő kérdésekben:

- részletesen meg kell határozni, hogy a megvalósítás eszközeként szolgáló adatbáziskezelő mely tulajdonságait, szolgáltatásait fogják felhasználni a fizikai adatterv elkészítésében;
- a terv első változatát, amely általános tervezési elveken alapul, eszköz függő változattá alakítja, támaszkodva az adatbázis kezelő jellegzetességeiből adódó szabályokra.

A háttértár igények felbecslése igen fontos tevékenység, ezért a kialakított fizikai adatcsoport hierarchiákat blokkokba kell osztani. A fizikai blokk hosszának megállapításához több tényezőt kell figyelembe venni:

- az adatbáziskezelő rendszer által támogatott blokk méretek;
- a leggyakrabban használt fizikai adatcsoportok méretét;
- az adatbáziskezelő belső adminisztrációjához szükséges járulékos költségeket (háttértár igény értelmében);
- a memóriába beolvasott blokkok tárigényét.

⁴⁵ Tömör vonal a fizikai adatterv alakítás első lépésének ábráján, "bütyök", kis köröcske nélkül.

⁴⁶ Ezzel a szabállyal korrigáljuk annak a választásnak a következményét; amikor az (al)entitást mégsem választottuk gyökérnek, ebben az esetben kerüljön az adatelérési szempontból hatékonyabb csoportba, vagyis ahol kulcson keresztül lehetne elérni a példányait, hiszen az egyik főentitás kulcsán keresztül lehet ezt megtenni.

A cél az, hogy a leggyakrabban használt legnagyobb fizikai adatcsoport beleférjen a választott blokk méretbe. Ezeket a leggyakrabban használt legnagyobb fizikai adatcsoportokat a funkció meghatározásokból, a hozzájuk kapcsolt lekérdezési utakból, eseményhatás ábrákból, a funkció komponensek (fragmentumok) használati gyakoriságából, események és lekérdezések gyakoriságából lehet megállapítani.

A választott blokk méretnek az adatbáziskezelő által felajánlott és megengedett méretek közé kell tartoznia, nem okozhat memória problémát akkor, amikor beolvassák a belső memóriába és egyidőben több tranzakciót kell kezelnie a rendszernek.

Ha a fizikai blokk méretet megállapították, le kell ellenőrizni, hogy vajon a fizikai adatcsoportok beleférnek-e. Mindegyik rekord méretet ki kell számolni az entitás leírásokban található adatok alapján, az adatelemek hosszának összeadásával. A kapcsolatok változása követésének költségét valamint a háttértáron elfoglalt hely kezelésének a költségét is meg kell becsülni.

Ha egy fizikai adatcsoport nem fér bele a választott fizikai blokk méretébe, akkor egy vagy több kisebb csoportra kell bontani.

9.1.5 A fizikai adatterv optimalizálása

Optimalizálás valójában egy meglehetősen költséges tevékenység, amihez ráadásul speciális szaktudásra van szükség az összetett feladatok megoldásához. Sok időt vesz igénybe és növeli a rendszer költségeit.

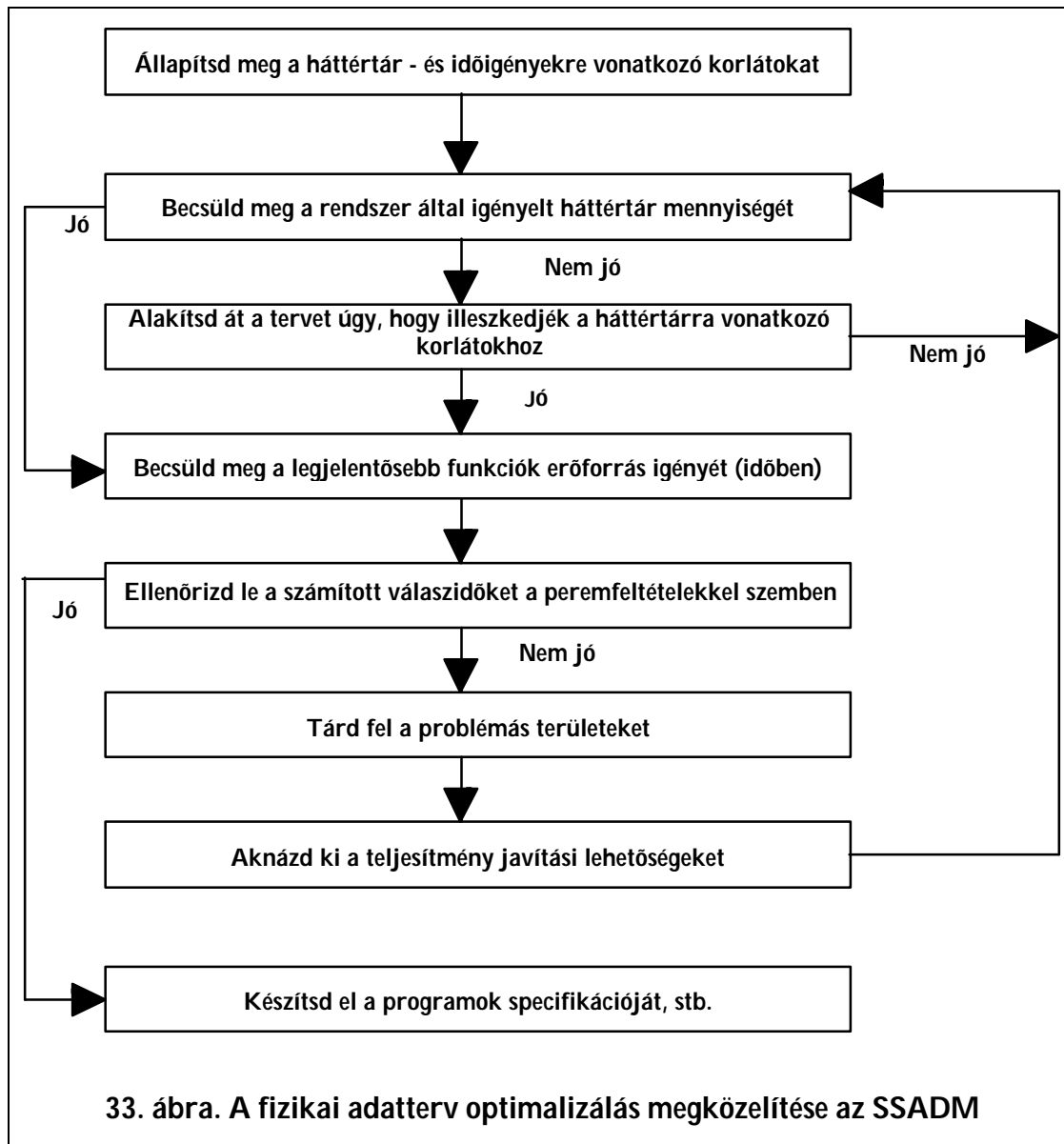
Az SSADM csak általános szabályokat határoz meg, két általános elv szem előtt tartásával:

- Optimalizálást csak akkor kell végezni, ha az előre rögzített teljesítmény célkitűzések csak ezen a módon érhetők el. Az optimalizálás segíthet a háttértár igény csökkentésében és / vagy az adatfeldolgozás idő igényének mérséklésben.
- A rendszeren belül meg kell őrizni fogalmi modellt. Minél közelebb van a megvalósított rendszer a fogalmi modellhez, annál könnyebb megérteni és karbantartani.

9.1.5.1 Az adatbázis optimalizálás lépései

Az eszköz függő fizikai tervezés nem ad semmilyen garanciát arra, hogy a rendszer elé állított teljesítmény követelmények teljesülnek-e, még akkor sem ha a tervet "elő-optimalizálták", heurisztikus (ököl szabályok) alkalmazásával; ilyen volt a 'legalacsonyabb függő előfordulások szabálya'.

Tehát egy szisztematikus optimalizálási folyamatra van szükség ahhoz, hogy az eszköz függő fizikai adattervet úgy módosítsák, hogy elérje vagy legalábbis megközelítse a rendszer elé állított teljesítmény követelményeket. ez természetesen egy iteratív eljárás (Isd. 33. ábra.).



A tervezőnek azonnal meg kell állnia, ha a terv eléri a kitűzött célokat. Fontos, hogy ne tervezzék túl a rendszert, ne pazarolják a munkaerőt olyan feladatokra, olyan peremfeltételek kielégítésére, amilyenek nem is léteznek. Ha az optimalizálási feladat túl bonyolult, akkor a felhasználóknak és a tervezőnek egyezkednie kell egyes feltételek, igények feladásáról vagy enyhítéséről.

9.1.5.2 Állapítsd meg a háttértár - és időigényekre vonatkozó korlátokat

A követelményjegyzékben rögzített igényekből levezethetők válaszüdő és háttértár paraméterek. Ha az eszközre készített fizikai adatterv az első lépésben kielégíti az eredeti peremfeltételeket, akkor közvetlenül is meg lehet valósítani, vagy lehet arra törekedni, hogy a megvalósítás során a terven további torzítások ne lépjenek fel. Ha azonban a terv nem találkozik az igényekkel, akkor további optimalizálási tevékenységekre lesz szükség, amelyekben esetleg a háttértár és / vagy válaszüdöket kell csökkenteni.

9.1.5.3 Becsüld meg a rendszer által igényelt háttértár mennyiségét

Ebben a becslésben támaszkodunk az egyes fizikai adatcsoportok elemeinek kiszámított méretére, amelyet megszorozunk az előfordulásuk gyakoriságával (hány példánnyal képviseltetik magukat tipikusan egy fizikai adatcsoporton belül). Az így kapott eredményeket összegezve egy mértéktartó becslést kapunk az adatbázis egészének méretére. Az egyes fizikai adatcsoportok méretét lényegesen megnövelheti azoknak a paramétereknek a figyelembe vétele, amelyek a memóriába betöltéshez szükséges adminisztrációhoz szükségesek, továbbá az adat hozzáférés biztonsági követelményeinek kielégítéséhez⁴⁷. Ezenkívül még számításba veendő tényezők: index tábla méretek, indexekhez szükséges adminisztráció költségei rekordonként, mutató láncok, stb. Ezeket a számításokat megkönnyítheti egy alkalmas kalkulációs lap használata, amelyek sok alkalmas szoftverben is létrehozhatók.

9.1.5.4 Alakítsd át a tervet úgy, hogy illeszkedjék a háttértárra vonatkozó korlátokhoz

Általában a háttértárra vonatkozó korlátozások kevés befolyásolják a végső tervet. Ha mégis a háttértárra vonatkozó korlátozások okoznak gondot, akkor is létezik több eljárás, ami segít a problémák megoldásában:

- Őrizzük meg az egy-egy értelmű megfeleltetést a fizika terv és a megvalósítás között:
 - osszuk el az adatokat jobban, háttértárak, hálózati csomópontok, stb. között;
 - használjunk adattömörítő eljárásokat.
- Ne őrizzük meg az egy-egy értelmű megfeleltetést a fizika terv és a megvalósítás között:
 - használjunk adattömörítő eljárásokat;
 - távolítsuk el a redundáns, fölöslegesen tárolt adatokat;
 - csökkentsük a visszamenőlegesen tárolt adatok mennyiségét.
- Ha szükséges akkor további módszerek is alkalmazhatók, nevezetesen, azokat az eljárásokat, amelyeket eddig teljesítmény (lényegében a válaszidők) javítására használtak, most "megfordítva" alkalmazzák a hely takarékoság érdekében (adatelérési mód, a blokk méret megváltoztatása, stb.).

A válaszidőt és a háttértár igényt meghatározó paraméterek között egy alku folyamat, kompromisszum keresési eljárás során lehet csak megnyugtató megoldást találni.

9.1.5.5 Becsüld meg a legjelentősebb funkciók erőforrás igényét (időben)

A legerőforrás-igényesebb funkciók meglehetősen könnyen felismerhetők. Ezek azok, amelyek azokkal az entitásokkal foglalkoznak, amelyeknek rekord populációja a legnagyobb - tipikusan a logikai adatmodell hierarchiáján belül a legalul elhelyezkedők.

Itt valóban csak a legjelentősebb, legfontosabb funkciókat kell használni, lehető legkisebb számban⁴⁸. A tervezőnek meg kell néznie, hogy vajon a fizikai adatterv első változatában mely fizikai rekordokat használja a kritikusnak tekintett funkció annak érdekében, hogy a logikai adatmodellben előírt adatelérést és navigációt meg tudja valósítani. Ezen a ponton a legprimitívebb eszköz, amelyet a tervező használhat, egy kalkulációs lap, amely a következő elemeket tartalmazhatja:

- a fizikai szintű adatelérési információk, azaz az elérendő rekordtípusok megnevezése, az elért rekord példányok száma, az adatelérés módja és az adat navigáció útvonala;
- a fizikai szintű adatelérésekhez, olvasásokhoz és írásokhoz, szükséges diszk hozzáfordulások számát is meg kell határozni. A tervezőnek meg kell állapítani, hogy az adatbáziskezelő egy rekord olvasási vagy írási művelete mennyi diszkhoz fordulást igényel⁴⁹.

9.1.5.6 Tárd fel a problémás területeket

A válaszidőre végzett becslések után az eredményeket össze kell vetni a célkitűzésekkel. A legnagyobb figyelmet a nagyszámú diszkhoz fordulásokra kell fordítani. Ha valamilyen probléma merül fel az összehasonlítás során akkor a rendelkezésre álló korlátozott eszköz készlettel meg kell próbálni megoldani.

9.1.5.7 Aknázd ki a teljesítmény javítási lehetőségeket

Elképzelhető, hogy a rendszer teljesítménye úgy növelhető, hogy a fogalmi modell szerkezetét megőrizték és az adatbáziskezelő szolgáltatásait alkalmazzák a teljesítmény javítására. Ebben az esetben az a célszerű, hogyha a kritikus funkciók egészére elvégzik az idő becsléseket és nem egyenként, külön-külön próbálkoznának meg a funkciók teljesítményének javításával.

A következő eljárások és technikák jöhetnek szóba:

- optimalizálják az entitások tárolási mechanizmusát, a következőképpen:
- az alentitásokat azoknak a főentitásoknak a közelébe helyezték, amelyekhez vezető kapcsolatot a leggyakrabban használják;
- változtassák meg az adatok elérési mechanizmusát (hashelésről indexeltre illetve megfordítva);

⁴⁷ Load, security parameters

⁴⁸ Tekintettel a Pareto-elvre (80 / 20) és a mini-max szabályra, vagyis a lehető legkevesebb elem (itt most funkció) felhasználásával a lehető legtöbb, legpontosabb információ kinyerése a cél. Ez ismét egy megfelelő kompromisszum megtalálását jelenti.

⁴⁹ Például az ún. 'cache' mechanizmus miatt az adatbáziskezelők nem olvassák be újra a memóriába azt a blokkot, amelyet röviddel azelőtt beolvastak és még mindig elérhető a memóriában. Az ehhez a feladathoz kapcsolódó egyre bonyolultabb algoritmusok gyakran fekete mágiává változtatják az erre vonatkozó becsléseket.

- a csupa kulcs entitásokat indexekként valósítsák meg vagy az indexek helyett csupa kulcs entitásokat vezessenek be;
- az alentitások rekordjai számára definiáljanak közvetlen adatelérési módot.
- optimalizálják a kapcsolatok tárolási mechanizmusát, a következőképpen:
- használják ki a rendelkezésre álló különböző indexelési mechanizmusok szolgáltatásait (parciális indexek a klasszterezett rekordokra, tömörített indexek, stb.);
- használjanak előre- és visszafelé utaló, valamint a rekord tulajdonos rekordját kijelölő mutatókat;
- rendezzék kulcs vagy más egyéb adatelem szerint a rekordokat.
- csökkentsék a diszkhez fordulások számát azzal, hogy egy hivatkozási táblát⁵⁰ közvetlenül a memóriában tartanak.

Ha a tervet egy ponton, egy kritikus funkcionál teljesítmény javítási okokból módosítják, akkor le kell ellenőrizni azt, hogy vajon a többi kritikus funkcionál nem következett-e be teljesítmény csökkenés.

Van néhány olyan eljárás is, ami nem rontja el a fizikai terv, a megvalósítás és a logikai terv összhangját:

- a fizikai adatbázis blokk méret módosítása;
- a rekord puffer befogadó kapacitásának változtatása;
- a tárolási sűrűség, az adattömörítési tényezők korrigálása;
- a leggyakrabban használt adatokat a leggyorsabb adatelérésű periférián tárolják;
- a leggyakrabban használt adatokat a rekord pufferben tartásuk, csökkentsék az adatbázis-kezeléshez szükséges adminisztrációs terheket (időben és helyben);
- optimalizálják az adatállományok fizikai elhelyezkedését a diszkeken;
- készítsenek csak olvasásra használt adatbázis példányt (ha másolat használata nem az interaktív terhelés idejére esik, vagyis kötegetelt feldolgozások támogatására, ahol a naprakészség nem olyan lényeges, mert ebben az időszakban az adatbázis nem módosul.)

Lehetőség van arra is, hogy a fogalmi modell szerkezetét nem tartják meg, hanem "elrontják" a megvalósítás érdekében vagy egyáltalán nem használják ki az adatbáziskezelők tulajdonságát. Ezek a megoldások azonban esetleg egy csak nehezen karbantartható rendszerhez vezethetnek, vagy egy olyanhoz, amelyben a felhasználók *ad-hoc* lekérdezéseit nem lehet egyszerűen megoldani, vagyis informatikus szakemberre van szükség a kérdések megfogalmazásához; a felhasználók maguk ezt

nem tudják megoldani, mert a megvalósított adatszerkezet olyannyira különbözik a logikai adatmodelltől és a felhasználók fejében létező mentális modelltől. A tervezőknek lehetőleg el kell kerülniük az ilyen megoldásokat, és csak vég szükségben érdemes ezeket alkalmazni, és alkalmazásuk előtt mindenképpen célszerű megvizsgálni, hogy a nyújtott teljesítmény nem kielégítő-e a felhasználók számára, és így ezeknek a technikáknak az alkalmazása nélkülözhető.

Tehát a további optimalizáláshoz használható technikák:

- az aktualizálási műveletek elhalasztása;
- az összes aktualizálási művelet elhalasztása;
- a kapcsolatokat módosító aktualizálási műveletek elhalasztása;
- a törlési műveletek elhalasztása.
- redundáns adatokkal bővítik az adatbázist;
- a főentitásokban a származtatott / számított adatmezőkkel bővítik a rekordot;
- a főentitás egyes adatmezőit helyezték el az alentitásban;
- változtassák meg az entitás rekord formában való megjelenítését;
- a főentitás rekordjába csomagolják be az alentitások rekordjait⁵¹;
- két vagy több rekordtípusra bontsák fel, amelyeket pl. a kulcson keresztül lehet elérni.
- bontsák fel a funkciókat úgy, hogy az fizikai adatcsoportok hierarchiáját vagy a rekordok tárolási sorrendjét ki tudják használni az egyes rész funkciók;
- készítsenek csak olvasásra használt adatbázis példány, amit interaktív módon lehet használni, a konzisztencia biztosítását megfelelő kompromisszumokkal oldják meg⁵².

9.2 Fizikai folyamatok specifikációja

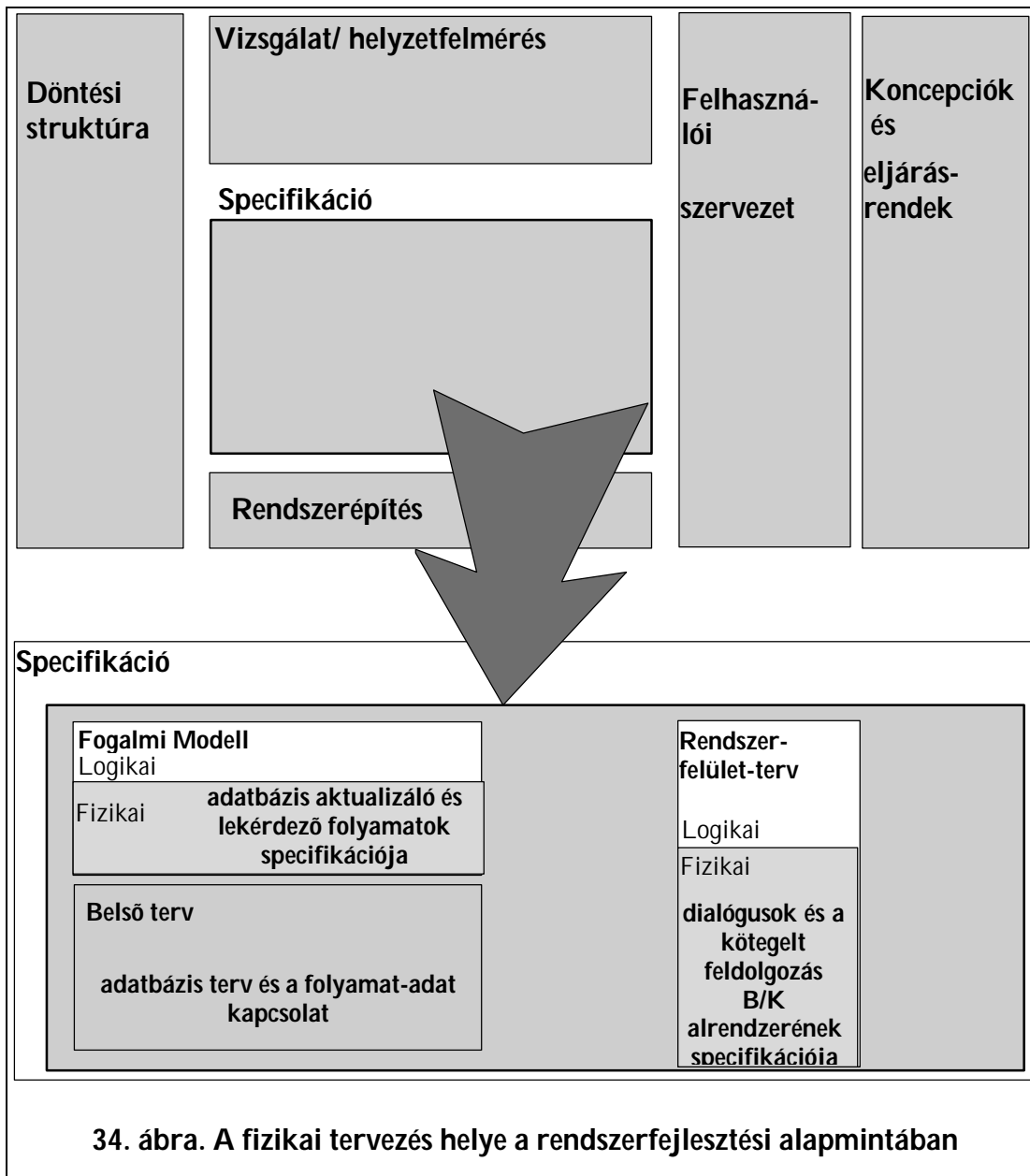
Az SSADM a néhány általános tervezési elvet tartalmaz ezen a területen, amelyek a fizikai tervezés részlet kérdéseivel csak korlátozott mélységben foglalkoznak. A legfontosabb elv, amit érdemes szem előtt tartani az az, hogy a strukturált terve kialakításra fordított erőfeszítéseket nem érdemes elpazarolni. Vannak olyan adatbázis és program tervezők, akik szeretnék teljesen átszervezni az SSADM-ben készült rendszer specifikációkat, amivel elvesztik azt az előnyt, amelyet az SSADM strukturált specifikáció nyújt a karbantartásban és jövődő további fejlesztéseknél.

⁵⁰ look-up table

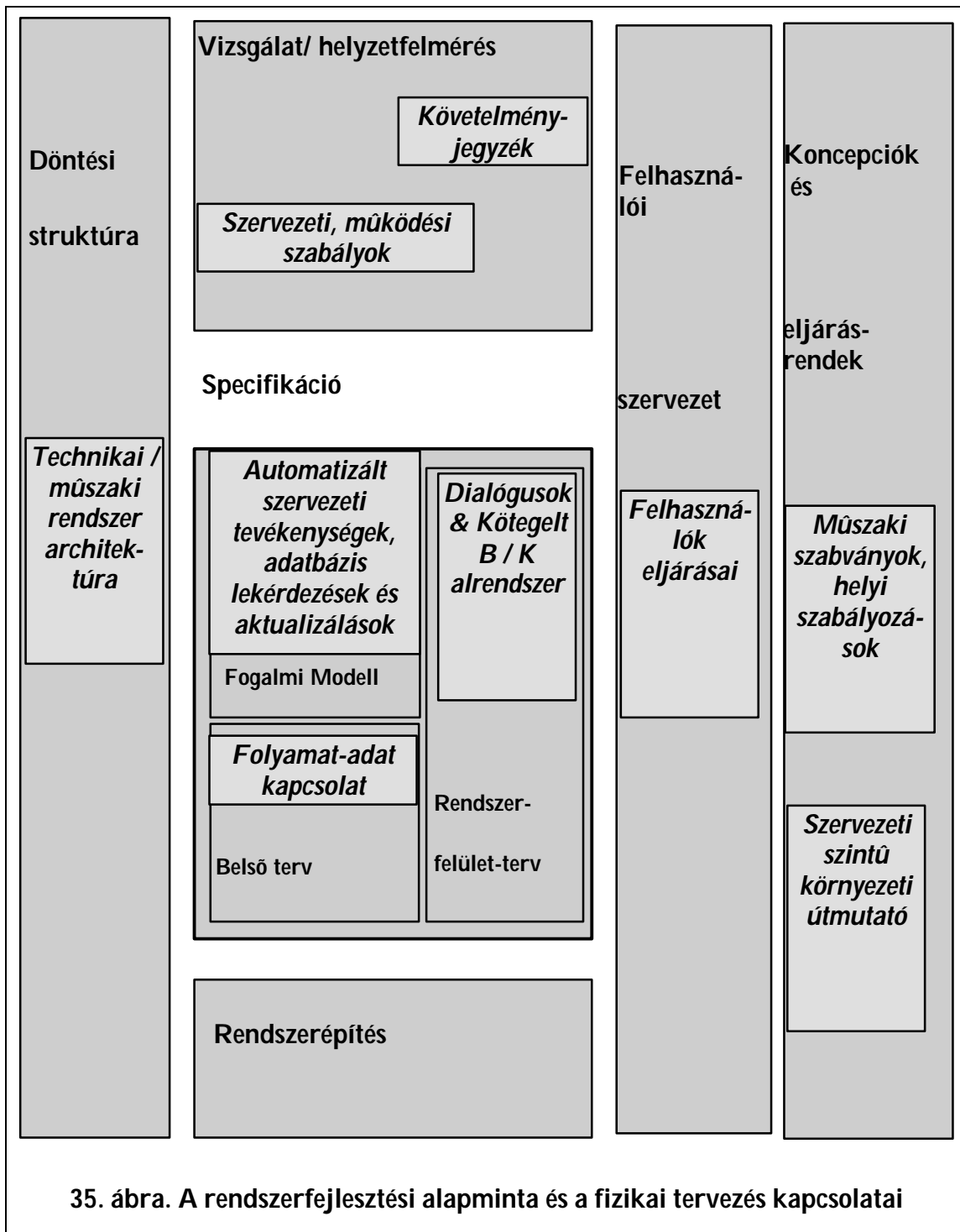
⁵¹ Ez a denormalizálás az 1. normálforma alak tulajdonság elrontása, a rekordtípusban ismét lesz ismétlődő csoport.

⁵² Snapshot adatbázis (pillanatfelvétel adatbázis), hátránya, hogy csak a pillanatfelvétel készítés konzisztens állapotát tükrözi, és kétszer annyi helyet igényel mintha csak egy példány volna, de lényegesen felgyorsítja a csak olvasási műveleteket, és nem terheli a költségesebb, aktualizálást elszoruló adatbázist.

A fizikai folyamatok specifikálása a rendszerfejlesztési alapminta három területére támaszkodik, a 3-séma specifikációs architektúrára. Kettő közülük egyaránt tartalmaz logikai és fizikai komponenseket is (Isd. 34. ábra.), a "Belső terv"-nek csak fizikai alkotórésze van.



A funkció-komponens megvalósítási terv pedig lefedi a 3-séma architektúra mind három részét. Természetesen a rendszerfejlesztési alapminta egyéb részeitől is kap bemenő dokumentumokat a fizikai tervezés, és kölcsönhatásban áll bizonyos részekkel (35. ábra.) . A követelményjegyzék tartalmazza azt, hogy valójában mit kell az új rendszernek nyújtania. A műszaki / technikai architektúra az új rendszer műszaki elemeit írja le. A "Koncepciók és az eljárásrendek" egyes döntési helyzetekre nyújtanak útmutatást és ezekkel együtt meghatározzák azokat a korlátokat és peremfeltételeket, amelyeket figyelembe kell venni. A felhasználók eljárásai pedig azt jelölik ki, hogy mit is kell tulajdonképpen kifejleszteni, és fizikai folyamatok specifikálása során megtervezni.



9.2.1 A fizikai folyamat specifikálás bemenetei

A technikai / műszaki architektúra az egyik legfontosabb bemenete, ez írja le az új rendszer különböző alkotó elemeit és a megvalósításukhoz alkalmazható technológiákat.

A másik fontos bemenet "Konceptiók és az eljárásrendek"-ből jön, ez a szervezeti elem a következőkről fog gondoskodni:

- **Az adatfeldolgozási rendszer jellemzése.** Ez a dokumentum a megvalósítási technológia sajátosságait fogja leírni, abban az értelemben, hogy az SSADM specifikáció, logikai terv egyes elemeit, hogyan lehet a technológia egyes elmeit használva, kiaknázva. Ide tartozó probléma körök: a procedurális és nem-procedurális nyelvek alkalmazási területei, vegyes felhasználásuk helyei, az adatbázis kezelő beépített eljárásainak, egyéb adatfeldolgozási szolgáltatásainak, valamint a dialógus szerkesztő eszközeinek javasolt alkalmazási módjai, területei, szabályai, szabványai.
- **Alkalmazás szintű környezeti útmutató.** Ez a dokumentum voltaképpen szabályok, belső és külső szabványok gyűjteménye, amely kritérium rendszert tartalmaz arra, hogy hogyan kell a funkciók alkotó részeinek megvalósítására a rendelkezésre álló technológiát alkalmazni, hogyan hajtsák végre a fizikai folyamat tervezést a program specifikáció szabályait milyen módon alkalmazzák. A legtöbb szervezetnél létezik általános belső szabályozás és szabvány, amelyet az egyes projektekre testre lehet szabni, a lokális igényekhez lehet igazítani. Ha azonban ilyen belső szabályozás nem létezik, akkor ezt a fizikai folyamat tervezés előtt ki kell dolgoztatni.

A fizikai folyamat tervezés leginkább a következő SSADM stílusban előállított termékekre támaszkodik: A funkció meghatározásokra; az ehhez szorosan kapcsolódó fogalmi folyamat modellezésre; a dialógus tervezés, felhasználói rendszer felület dokumentumaira. Ezeknek a termékeknek az összessége nagyon részletesen leírást ad a fizikai folyamatok specifikálásához.

A fizikai folyamat specifikáció **célja** az, hogy az SSADM termékeket bizonyos finomítási és transzformációs lépések után alakítsák a fizikai megfelelőiké, azonban ez nem azt jelenti, hogy egyszerűen csak átmásolják az eredeti termékek tartalmát a program specifikációkba, vagy a semmiből alakítsák ki a program specifikációkat tekintet nélkül az eddig elkészített SSADM termékekre⁵³.

9.2.2 A fizikai folyamat specifikálás módszere

A fizikai folyamatok specifikálása a 3-séma architektúrára alapul, amely a követelmény specifikáció és a logikai specifikáció moduljaiban végzett munka elméleti támogatását adta. A fogalmi modell, rendszerfelület-terv és a belső terv szétválasztásának koncepciója megőrződik a fizikai tervezésben is.

Ennek az elválasztásnak a megőrzése több előnyt is nyújt:

- az áttérés a logikai tervről a fizikai tervre sokkal könnyebb;
- az SSADM logikai szintű termékei sokkal könnyebben képezhetők le a fizikai megfelelőikre, ezzel a karbantartás és a további fejlesztések számára jelentős segítséget tudnak nyújtani;

⁵³ Ideális esetben ezt az átalakítást egy magas technológiai színvonalon álló CASE eszköz nagy mértékben segítheti, esetleg az emberi munkaerő igényt minimálisra csökkentheti.

- a dekompozíció elve itt is érvényesül, azaz nem egy hatalmas feladatot kell egyszerre megoldani, hanem három viszonylag kisebb részterületen kell a megoldásokat kidolgozni;
- az elosztott és az ügyfél-kiszolgáló⁵⁴ architektúra szellemében készíthető rendszerek építőelemeire sokkal könnyebben képezhetők le a fizikai terve egyes alkotó elemei. A séma egyes elemeinek megvalósítására különböző technológiai megoldások alkalmazhatók, miközben a közöttük fennálló adat és funkcionális kapcsolat nagyon részletesen és pontosan szabályozott⁵⁵;
- az ősrégi kövületként⁵⁶ megmaradt rendszerek, valamint további közösen használt, megosztott használatú adatbázisok elrejtethetők az adat-folyamat kapcsolat⁵⁷ mögött.

9.2.2.1 A 3-séma specifikációs architektúra

Tehát a fentebb mondottaknak megfelelően a fogalmi modell, a rendszerfelület-terv és a belső tervek szétválasztásának koncepciója megőrződik a fizikai tervezésben is. A 3-séma architektúra leképezését a fizikai tervek elemeire a következő ábrán lehet látni (36. ábra.). Ebben az ábrában az adatbázis a belső tervek részeként látható, ennek az az indoka, hogy az adatbázis tervek a fizikai adattervek része.

A 3-séma architektúra különösen a karbantartási feladatoknál hasznos, mivel a változtatások hatásának terjedését, a nem szándékolt melléhatásokat segíti kézben tartani:

- a munka szervezésben, munka folyamatokban bekövetkező változások (vagyis ki mit csinál) csak a rendszerfelület tervére gyakorol hatást, feltéve, hogy a kapcsolódó szervezeti tevékenységekben nem történik változás;
- az automatizált szervezeti / működési szabályokban fellépő változások csak a fogalmi modell adatfeldolgozó folyamatait befolyásolják, hacsak az igényelt információ szolgáltatással és kapcsolódó munkafolyamattal szemben támasztott igények nem változnak;
- a felhasználói felület létrehozásához alkalmazott technológia megváltozása, az adatbázis átszervezése (teljesítmény javítási okokból), az adatok átmentése egy régebbi rendszerből, mind jól elhatárolható hatásokat és következményeket jelent, így a változtatási igények elszigetelhetők és elkerülhető az, hogy a rendszer összes komponensét módosítani kelljen.

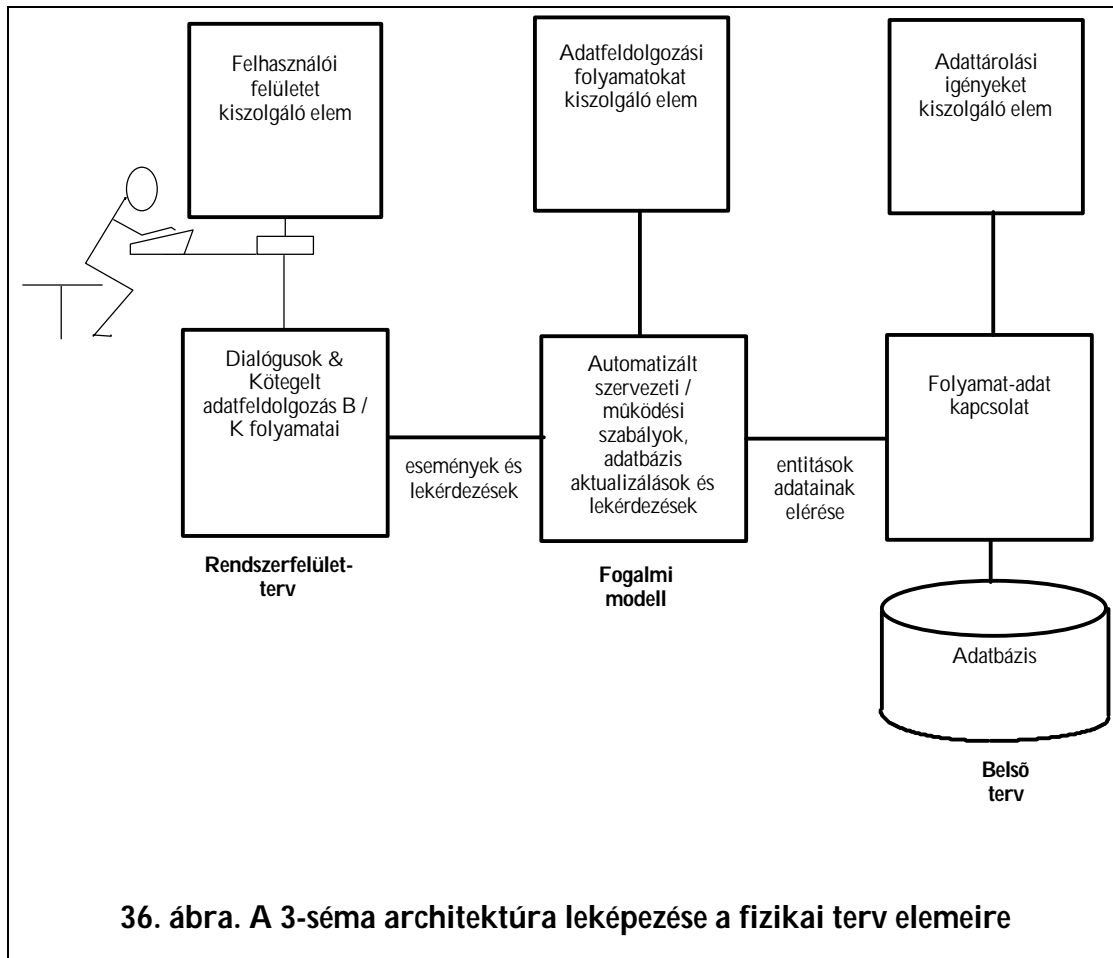
A fizikai folyamat tervezés első lépése annak meghatározása, hogy vajon a 3-séma architektúra egyes részeiben mit kell megtervezni. A 3-séma architektúra nem írja elő,

⁵⁴ Client-server

⁵⁵ A rendszerfelület-tervét esetleg több különböző platformon (eltérő hardver és szoftver eszközök kombinációján) valósítják meg, míg a belső tervet pedig adatbáziskezelők alkalmas kombinációján valósítják meg.

⁵⁶ legacy systems

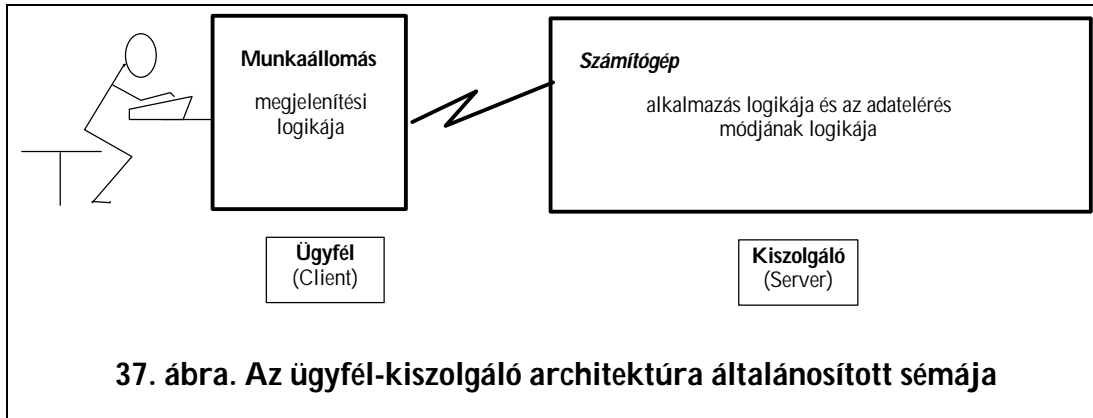
hogyan az egyes részeket hogyan kellene megvalósítani. Sok választási lehetőség áll fenn: mindegyik sémát lehet akár ugyanabban a technológiában megvalósítani, de akár mindegyik séma esetében az alkalmazott technológia lényegesen különbözhet.



9.2.2.2. Az ügyfél-kiszolgáló architektúra és a 3-séma architektúra

Az ügyfél-kiszolgáló architektúra azok egyike azok közül, amelyek különösen jól illeszkednek a 3-séma architektúrához. Az ügyfél-kiszolgáló architektúrában az 'ügyfél' által kezdeményezett folyamat kérését intézi a 'kiszolgáló' oldal egyik folyamatához, amelyik végrehajtja a kérést és visszaküldi a választ. A gyakorlatban az 'ügyfél' folyamatait munkaállomásokon vagy személyi számítógépeken valósítják meg, míg a 'kiszolgáló' folyamatait egy olyan számítógépre telepítik, melyet több 'ügyfél' is használhat. Az ügyfél-kiszolgáló architektúra általános szerkezetét a következő ábra tartalmazza (37. ábra.):

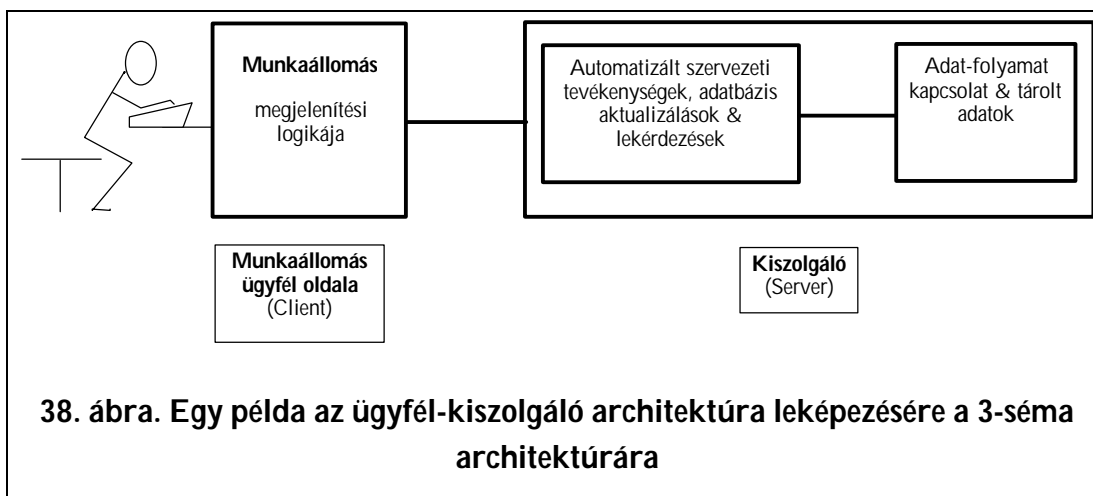
⁵⁷ PDI, Process Data Interface.



A 3-séma architektúra legegyszerűbb leképezése az ügyfél-kiszolgáló architektúrára a következő:

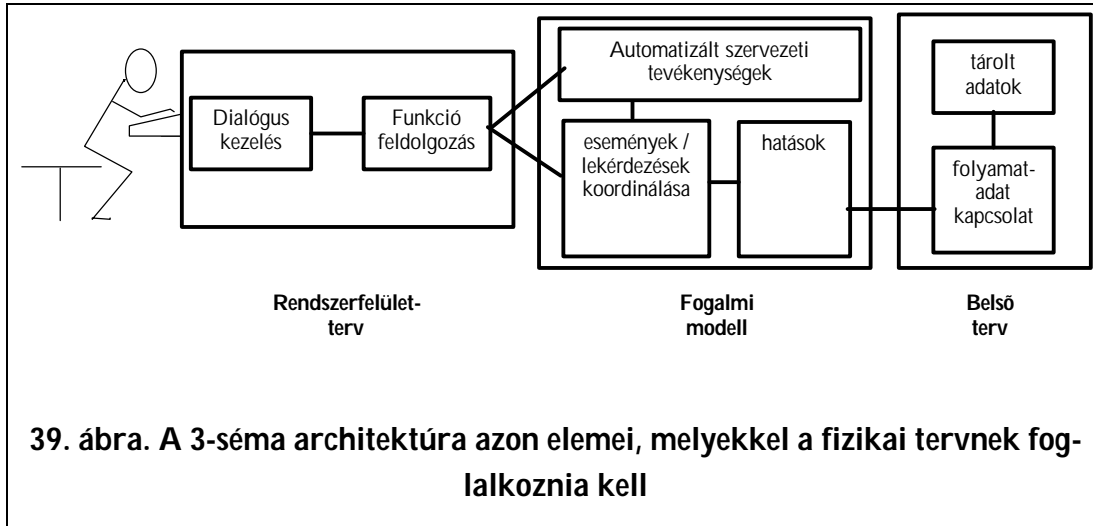
- A rendszerfelület tervének megvalósítását az 'ügyfél' oldal elkészítésével lehet azonosítani;
- a fogalmi modell és a belső terv megvalósítását pedig a 'kiszolgáló' oldal felépítésével lehet azonosítani.

Ennek megfelelő leképezést az ábra mutatja (38. ábra.), a gyakorlatban persze gyakran teljesen máshol húzódhat a határvonal, sőt funkcióról funkcióra is különbözhet⁵⁸.



Ha a rendszerfelület tervét és a fogalmi modellt közelebbről vizsgáljuk, akkor további olyan alkotórészeket találunk, amelyekkel a fizikai tervnek foglalkoznia kell.

⁵⁸ [ISE94A], *SSADM and Client Server Applications*, ISE Library, HMSO, 1994.



A fentebbi ábrán bemutatott alkotó elemek a következők (39. ábra.):

- A rendszerfelület terve logikai szinten dialógusokat és funkciókat tartalmaz, a fizikai tervben a dialógusoknak és a funkcióknak megfelelő fizikai alkotó elemeket kell leírni:
- a dialógusok kezelése tartalmazni fogja a funkciók megjelenítését, a felhasználók felé mutatott képét, a rendszer szolgáltatásai közötti navigációt, menük segítségével illetve közvetlen utasítások (pl. funkció gombok segítségével), továbbá a tájékoztató és segítő információkat;
- a funkció feldolgozás a következőt jelenti: az egyes eseményeket illetve a lekérdezések kezdeményezését fel kell ismerni és ki kell válogatni a képernyőn keresztül kapott információk közül, majd a megfelelő, a fogalmi modellben leírt folyamatok közül azokat kell meghívni, amelyeket ezeknek a lekezelésére készítettek fel. Ide tartozna a fogalmi modell folyamatai kimenetének kezelése is, nevezetesen az adatok esetleges sorba rendezése, válogatása, összegzése annak érdekében, hogy ezek az adatok vissza-kerüljenek a felhasználók képernyőire.
- A fogalmi modell logikai szintje azokat a folyamatokat tartalmazza, amelyeket az események és lekérdezések kezdeményeznek. A fizikai tervben ezeknek a folyamatoknak a leírásához a következő fizikai komponensekre van szükség:
- az automatizált szervezeti tevékenységek. A munkafolyamat modellben és a követelmény meghatározás során bizonyos szervezeti tevékenységeket a számítástechnikai rendszerre bízta a felhasználók helyett. A szervezeti / működési szabályok, amelyeket a szervezeti tevékenység modell ír le, megadják a rendszer üzemeltetés és működtetés kívánt módját, ezt a leírást kell megvalósítani;
- események és a lekérdezések koordinálása. Olyan fizikai folyamatokra van szükség, amelyek az adott esemény vagy lekérdezés végrehajtásához szükséges hatásokat és adat-visszakereséseket megvalósító fizikai folyamatokat meghívják, majd a kapott

eredményeket összegyűjtik és előállítják az adott eseményre vagy lekérdezésre adott válasz formájában;

- 'hatásokat kiváltó folyamatok' — a fogalmi folyamat modellezésben létrehozott folyamatokból állíthatók elő ezek a folyamatok többek között az esemény hatás diagrammok és a lekérdezési utak felhasználásával. Olyan fizikai folyamatokra van szükség, amelyek ezeket a folyamatokat valósítják meg és amelyek tartalmazni fogják az ide tartozó hatások és adatelérések csoportjait, valamint a feltételes választásokat (szelekciók) és iterációkat, a ciklus és választási feltételek megfogalmazását, továbbá a műveleteket.

Az események és hatások koordinációját végző folyamatok - egyes technológiai környezetekben - beolvaszthatók az aktualizálási és lekérdezési folyamat modellekbe.

9.2.2.3 A 3-séma architektúra leképezése a hardver konfigurációra

Természetesen több lehetőség is van a 3-séma architektúra leképezésére az aktuális hardver környezetre, néhány példa:

- a dialógus kezelést a felhasználói munkaállomásokon lehet megvalósítani, míg a többi adatfeldolgozó elemet pedig egy számítóközpontban (mainframe);
- a rendszerfelület tervét és az automatizált szervezeti folyamatokat a felhasználói munkaállomásokon lehet megvalósítani, az események és lekérdezések folyamatait pedig az adatbázist kiszolgáló oldalon tárolt adatbázis eljárások⁵⁹ formájában. Ez a megoldás legjobban egy nem elosztott, ügyfél-kiszolgáló architektúrához illeszkedik;
- a rendszerfelület tervét, az automatizált szervezeti folyamatokat, és az események / lekérdezések koordinálást végző folyamatokat a felhasználói munkaállomásokon lehet megvalósítani, a hatásokat kiváltó és az adatelérések végrehajtó folyamatokat pedig az adatbázist kiszolgáló oldalon tárolt adatbázis eljárások formájában. Ez a megoldás jól illeszkedik egy elosztott ügyfél-kiszolgáló architektúrához;
- a rendszerfelület tervét és az automatizált szervezeti folyamatokat, az események és lekérdezések folyamatait, a folyamat-adat kapcsolatot egy olyan alkalmazást kiszolgáló elemen⁶⁰ valósítják meg, amely viszont egy elosztott adatbázist érhet el. Ez a megoldás olyan elosztott rendszereknél jön szóba, amelyeknél alkalmazzák az adatok replikálását, azaz másolatot tartanak bizonyos adatokról földrajzilag különböző telephelyeken és valamilyen mechanizmussal gondoskodnak a konzisztenciáról; illetve akkor jó megoldás ez, ha az adatok földrajzi elhelyezkedése változik.

⁵⁹ stored procedures

⁶⁰ local application server

9.2.2.4 A probléma területek szétválasztása nem mindig őrizhető meg

Egyes gyártók által nyújtott technológiai megoldások összeolvasztják a 3-séma architektúra egyes területeit azért, hogy gyorsabb fejlesztést és hatékonyabb megvalósítást tegyenek lehetővé. Gyakran előforduló esetek:

- forma lap alapú felhasználói felületek, amelyek közvetlenül a relációs adatbáziskezelő tábláin manipulálnak anélkül, hogy a fogalmi modellt elkészítették volna;
- gyártó specifikus SQL dialektus, amelynek lehetnek olyan SQL kiterjesztései, amelyek olyan szolgáltatásokat nyújtanak, amivel ki lehet használni az adattárolás fizikai tulajdonságait és így magasabb teljesítmények érhetők ezzel el.

Ezeknek a lehetőségeknek a kihasználása általában a lehető legjobb választás a projekt részéről. Az egyetlen pont, amelyet világosan látni kell az az, hogy ez egy kompromisszumot jelent, egyik oldalról egy gyártó specifikus technológiától való kizárólagos függést, a másik oldalról pedig gyorsabb fejlesztést, esetleg egyszerűbb karbantartási feladatokat, és beláthatatlan könnyebbséget vagy nehézséget a továbbfejlesztéseknél, mivel a 3-séma architektúra nem őrződik meg a fizikai tervezés során.

9.2.2.5 A fizikai folyamat specifikáció termékei

Az SSADM a következő termékeket elengedhetetlenül fontosnak tartja a specifikáció során, de a helyi szabályok előírhatnak természetesen további elkészítendő és esetleg leszállítandó termékeket:

- funkció-komponens megvalósítási terv;
- folyamat-adat kapcsolat;
- program specifikáció.

9.2.2.5.1 A funkció-komponens megvalósítási terv

A funkció-komponens megvalósítási terv teremti meg a fizikai folyamat specifikáció keretét:

- meghatározza azt, hogy hogyan kell az logikai rendszerterv / specifikáció egyes elemeit csoportosítani azért, hogy azok tartalma kiadja a megfelelő fizikai komponensek tartalmát;
- hogyan illeszkednek össze a fizikai specifikáció elemei, melyeket lehet többször felhasználni.

A funkció-komponens megvalósítási terv valódi célja az ún. procedurális alkotórészek leírása, ugyanis a nem-procedurális részeket a megvalósítási környezet (általában 4 GL nyelvben) le lehet írni.

A funkció-komponens megvalósítási terv megjelenési formája projektről projektre változhat, az alkalmazott fejlesztési környezet függvényében: adatszótár, repozitórium, adatraktár, CASE eszköz:

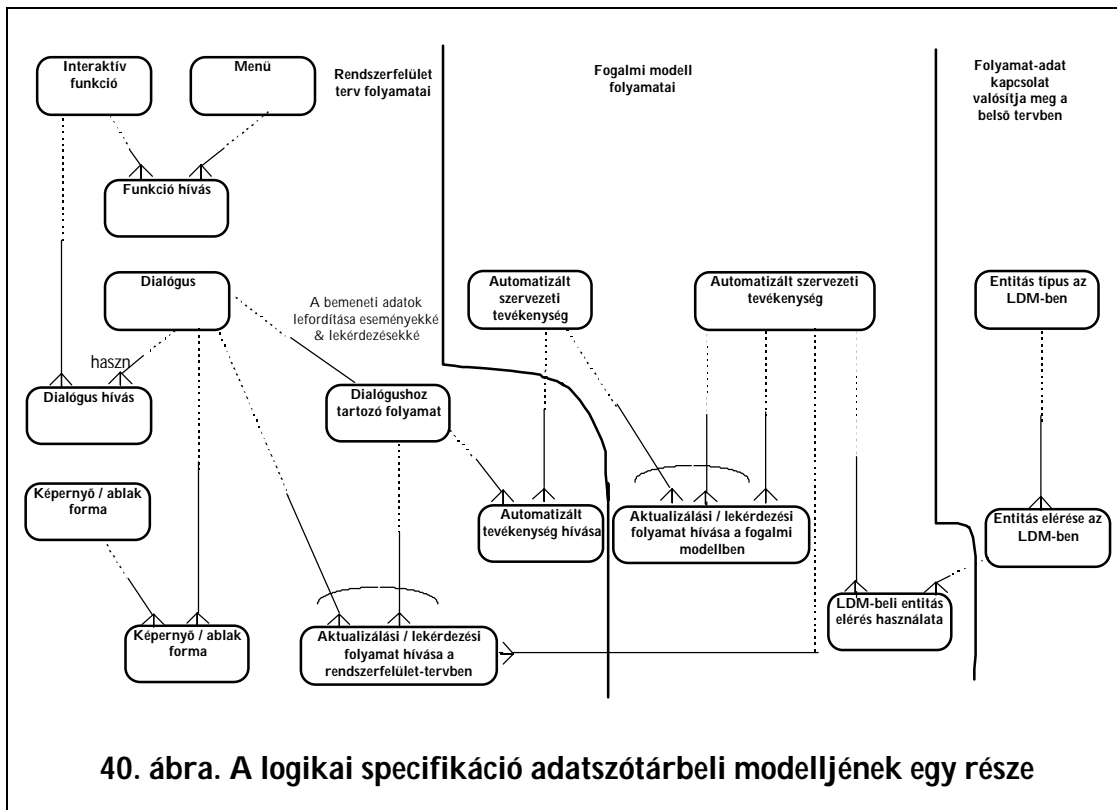
- Ha egy adatszótárt alkalmaznak a fizikai és logikai komponensek tárolására, akkor az alkotó elemek közötti kapcsolatok kijelölésével megoldható ez a feladat;
- ha különböznek az adatszótárak, akkor valamilyen egyéb adatszótárt kiegészítő eszközzel lehet összekötni a fizikai és a logikai komponenseket, pl. táblázatokkal, kiadatbázissal, stb.

A fizikai terv alkotó elemeit a következők figyelembe vételével lehet meghatározni:

- a 3-séma architektúra szerinti felosztás számításba vételével, a rendszerfelület tervének folyamatai és a fogalmi modell folyamatai valószínűleg különböző fizikai komponensbe kerülnek;
- "hívási" kapcsolat, vagyis ha egy lekérdezést egy automatizált szervezeti tevékenység használni kíván, akkor a két egységet érdemes ugyanabba a fizikai alkotó részbe helyezni;
- A műszaki architektúra alkotó elemeihez rendelés. Ha még nem eldöntött, hogy egy valamelyik dialógushoz tartozó folyamat megvalósítása az 'ügyfél' vagy a 'kiszolgáló' oldalon lesz-e, akkor érdemes a dialógust és hozzátartozó folyamatot külön álló komponensként megfogalmazni;
- a megvalósítás helyi környezetének (platformnak) az igényei. Ha egy dialógust eltérő technikai tulajdonságú munkaállomásokon fogják megvalósítani, akkor ezeket a változatokat külön-külön kell specifikálni.

A funkció-komponens megvalósítási terv létrehozásakor a tervezőnek két különböző szinten kell dolgoznia:

- egyrészt keresni kell az újra vagy többször felhasználható elemeket a funkciók komponensei között;
- további részletekkel egészíti ki a funkció specifikációkat, funkció-komponens megvalósítási tervet lehetőleg teljes egészévé téve.



9.2.2.5.2 A folyamat-adat kapcsolat

A folyamat-adat kapcsolat tulajdonképpen egy olyan szoftver réteg, amely a fizikai adatbázis megvalósítási sajátosságait rejti el a fogalmi modell folyamatainak szeme elől. Az adatbázis adatfeldolgozó folyamatai úgy írhatók meg mintha a logikai adatmodellen dolgoznának. Az egyik legegyszerűbb megoldás az, ha az adatbázis bizonyos törzstábláinak nézeteit (views) és az ezekhez tartozó SQL hívásokat valamilyen programozási nyelvben írt adatfeldolgozó folyamatokba ágyazzák be (pl. C, COBOL). A másik végtel pedig az, ha adatbázisok elosztott hálózatát és ősrégi rendszereket bújtatnak el a folyamat-adat kapcsolat mögé, ezzel elrejtve az adatok tényleges helyét illetve esetleg hatékonysági okokból az adatbázisról készült másolatokat (replikációk).

A folyamat-adat kapcsolat lehet:

- egy általánosított szoftver elem, modul, amely kezeli a logikai adatmodell, a fogalmi folyamat modell és a fizikai megvalósítás között fennálló különbségeket és melyet az összes aktualizálást vagy lekérdezést végző folyamat meghív;
- szoftver elemek halmaza, mint pl. rutin könyvtár, vagy egyéb előre elkészített részben általánosított elemek, amelyeket bizonyos aktualizálási és lekérdezési folyamatokhoz kötnék.

A fentebb felvázolt kép persze csak egy ideális képet tükröz, a valóságban a feladat gyakran nem oldható meg ilyenképpen, azaz az adatfeldolgozási folyamatok belső szerkezete visszafogja tükrözni az adatbázis fizikai tervezése során hozott döntéseket,

pl. hatékonysági okokból végrehajtott változtatásokat, vagy a megvalósítási eszköz bizonyos szolgáltatásainak kiaknázása miatt, pl. a programozás vagy a tesztelés könnyítése miatt.

Néhány fogalmi modellbeli folyamatot hatékonysági okokból lehet, hogy át kell tervezni, azért, hogy az adatbázis által szolgáltat választandó kielégítő legyen, ilyen lehet például a köteget feldolgozásoknál a bemeneti adatok két menetben való sorba rendezése, különböző rendezési kritériumok alapján, vagy az aktualizálások tényleges végrehajtását olyan periódusokra halasztják, amikor a rendszer terhelése alacsonyabb.

9.2.2.5.3 Program specifikációk

- Program generálás nem-procedurális kódból;
- Jackson rendszer programozási módszer (Jackson Systems Programming) szerinti specifikáció⁶¹;
- Strukturált természetes nyelv
- Angol, magyar,
- Formális specifikációs nyelv
- Z⁶², VDM⁶³

9.2.3 Az univerzális funkció modell

A fizikai folyamat tervezés az SSADM során létrehozott funkció specifikációk megvalósításával foglalkozik. Az univerzális vagy általános funkció modellt a funkciók meghatározásával foglalkozó fejezetben már megtárgyaltuk. Ezt a specifikációt fejlesztik tovább a fizikai folyamat tervezésben:

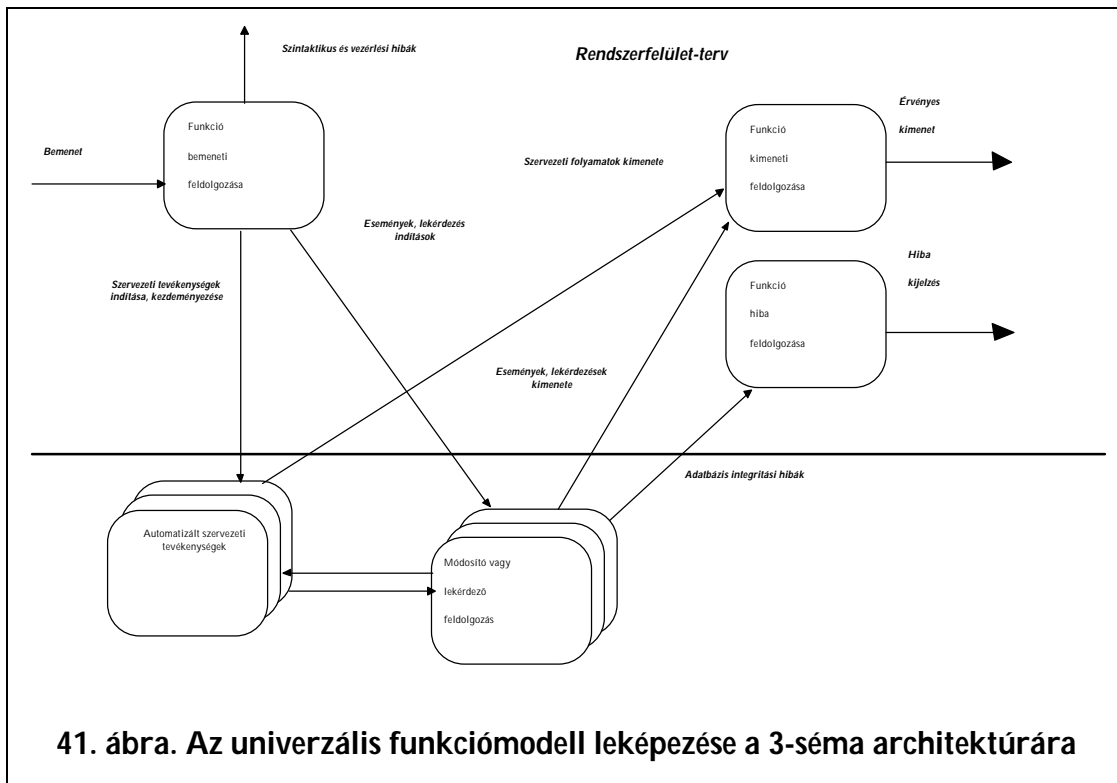
- az univerzális funkció modellt leképezik a 3-séma specifikációs architektúrára;
- a folyamat-adat kapcsolat résszel bővítik;
- az automatizált szervezeti tevékenységeket elkülönítik;
- a közbenső adatcserék, áramok részletes leírását is meghatározzák.

Ennek alapján az általánosított funkció modellt a következőképpen lehet átalakítani, tovább bővíteni (41. ábra.):

⁶¹ [Jackson82], [Cameron83], [Burgess90]

⁶² [Spivey88], [Spivey89]

⁶³ [Jones86]



A fizikai folyamatok specifikációja előtt teljesen érdektelen, hogy a tervező milyen módon képzelte el a fentebbi funkció modell részeinek megvalósítását, akár különálló programonként akár egy program moduljaiként vagy fragmentumaiként. A program specifikáció során rendelik az általános modell egyes elemeit a konkrét megvalósítási környezethez.

9.2.3.1 Specifikus vagy egyedi funkció modell

Egy specifikus funkció modell bizonyos funkciók esetében az egyes folyamat komponensek és adatfolyamok közötti egyedi kapcsolatokat jeleníti meg. Általában ezt jeleníti meg az univerzális funkció modell, de lehetnek olyan procedurális funkciók, amelyeknél szükség lehet egy eseti ábrázolásra, ez különösen akkor hasznos, amikor a funkció logikai felfogása és fizikai megvalósítása jelentősen eltér.

9.2.3.2 Fragmentumok és adatkapcsolatok

Az általános funkció modell mutatja azokat az alkotó részeket, amelyeket az egyes funkcióknál specifikálni és megvalósítani kell, valamint a köztük áramló adatokat. Ezeket a logikai alkotó elemeket a funkció-komponens megvalósítási tervben mint részelemeket, fragmentumokat kell megjeleníteni, két féle szempontból osztályozva:

- a fizika megvalósítási környezetben rendelkezésre álló objektum típusok (eszközök, szolgáltatások, sajtóságok, stb.) alapján, a leendő megvalósítás szerint;

- valamint a bemeneti és kimeneti adatokhoz tartozó feltételek, az eseményhatás ábrából, a lekérdezési útból és a fogalmi folyamat modellből származó logikai adatbázis műveletek⁶⁴.

A funkció-komponens megvalósítási terv egyes részei, fragmentumai közösen használt, többször felhasználható, újra hasznosítható folyamatokká válhatnak.

⁶⁴ Az SSADM által nyújtott pszeudó kód.

10. Áttekintés a megvalósíthatósági tanulmányról az SSADM 4+ban⁶⁵

A megvalósíthatósági tanulmány a leendő információrendszer rövid elemzése, felmérése és kiértékelése annak eldöntésére, hogy vajon a szervezet rendszerrel szemben támasztott igényei ténylegesen kielégíthetők-e, valamint továbbá létezik-e a tervezett projektekre vonatkozó üzleti, befektetési és kockázati elemzés⁶⁶.

A megvalósíthatósági elemzés elvégzését az SSADM kifejezetten ajánlja de nem teszi kötelezővé a teljes SSADM szerinti elemzés elvégzése előtt, általában egy teljes informatikai stratégiai elemzés⁶⁷ után következik.

Az SSADM módszertan alkalmasan szabható a megvalósíthatósági tanulmány elvégzésére, de a módszertan illesztésének súlypontjait a megvalósíthatósági tanulmány célkitűzése határozzák meg. A megvalósíthatósági elemzés általában nem tartalmazza viszont az entitás-viselkedés modellezés és a fogalmi folyamat modellezés részletes és sok erőforrást igénylő technikáit. A tipikus technika halmaz:

- adatfolyam modellezés;
- logikai adatmodellezés;
- követelményelemzés;
- (funkció meghatározás).

A megvalósíthatósági tanulmány célja azoknak az alternatíváknak a feltárása, amelyek az adott működési terület, szervezeti egységek informatikai támogatásaként szóba jönnek, valamint ezeknek a lehetőségnek egy kezelhető halmazra való szűkítése a teljes elemzés megkezdése előtt. Három tipikus, gyakran előforduló változata ezeknek a követelményeknek a következők:

- a szervezeti tevékenységek stabilak, viszont a meglévő informatikai rendszereket ki kell cserélni akár amiatt, hogy a rendszer technikailag elavult akár amiatt, hogy a karbantartása egyre nagyobb költségeket jelent;
- a szervezeti tevékenységek lassan megváltoztak azóta, amióta a jelenleg működő információrendszert üzembe helyezték, és ezért egy sokkal jobban illeszkedő rendszerre van szükség;

⁶⁵ [CCTA95], [CCTA95A], User Guide Part 1: Customising SSADM Appendix A: (Feasibility Study), 1-19—1-28. Továbbá [CCAT90].

⁶⁶ Ezt az angol szakirodalomban 'Busines Case'-nek hívják és két főrészből áll: (1) a költség / haszon elemzésből [Cost / Benefit Analysis], (2) az üzleti / szervezeti kockázat elemzésből (Business Risks Analysis / Risk Management).

⁶⁷ IT / IS Strategy Study, amelynek a végeredménye egy projekt-portfólió, a lehetséges / leendő projektek felsorolásával, a legfontosabb jellemzőikkel együtt.

- egy új működési területet alakítanak ki, üzletágot indítanak el, amelynek szüksége van egy új információrendszerre.

Egy másik gyakran felbukkanó igény az új technológiai lehetőségek kiaknázása, ha nyújtanak valami előnyt a szervezet, az üzleti tevékenység számára.

10.1 A megvalósíthatósági elemzés jellemzői SSADM-ben

10.1.1 Az elemzés kiterjedése

Egy megvalósíthatósági elemzést a következő esetekben lehet végrehajtani:

- egy informatikai stratégiai tervezés részeként;
- egy tender felhívásra benyújtott pályázat részeként, a kockázatok csökkentése érdekében (pl. Euromethodnak megfelelő tender válasz készítés keretében a rendszer kezdő és végállapotának pontosabb specifikálása érdekében);
- az információrendszer adaptációt előkészítő szerződés kötés előtt, során, a szerződés műszaki mellékletének részeként;
- önálló megvalósíthatósági elemzés — a szervezet egy bizonyos része lehetőségeinek vagy felismert problémáinak mérlegelése után.

A megvalósíthatósági elemzés egy vagy több teljes SSADM szerinti rendszerelemzési projekthez vezethet. Az informatikai igények elrendezésére több egymással összefüggő projektbe, és a végeredmények integrálására van egy ISE Library kötet: "Application Partitioning and Integration with SSADM".

A megvalósíthatósági elemzés során az informatikai lehetőségeket kell kiértékelni a következő értelemben:

- a szervezeti igények és célkitűzések támogatásának mértéke;
- szervezetre gyakorolt hatások (személyekre és feladatokra);
- az információrendszerrel szemben támasztott igények kielégítése;
- a fejlesztés és a rendszer elkészítésének műszaki megvalósíthatósága;
- költségek, előnyök, hasznok és kockázatok.

Az információrendszer elemzése során az információ-technológián alapuló és az információ-technológia-mentes rendszerek elemzésre is ki kell térni.

10.1.2 Tevékenységek

Bármilyen legyen is végül a megvalósíthatósági elemzés célja, a tanulmányt készítő csoportnak a következő lépéseket kell végrehajtania:

- az igényelt szervezeti / informatikai környezet leírása;

- a szervezeti, üzleti környezet kiértékelése;
- a követelmények és igények meghatározás és egyetértés kialakítása elfogadásukra;
- az információrendszerekre vonatkozó lehetséges fejlesztési és megvalósítási alternatívák azonosítása és kiértékelése;
- a rendszerszervezési és - technikai javaslat hatásának a megállapítása a szervezetre és alkalmazottakra vonatkozóan;
- a költségekre és a hasznokra egy becslés kialakítása;
- az alternatívák ismertetése a projektvezetőség és a szervezeti egység vezetők előtt.

A megvalósíthatósági tanulmány a megvalósíthatósági elemzés végeredménye. A projektvezetőség ennek alapján fogja el dönteni:

- a teljes SSADM elemzés végrehajtását engedélyezze és kezdeményezze;
- vagy a megvalósíthatósági tanulmányban részeként létrejött 'Projekt alapító okiratban' elképzelt fejlesztési és informatikai elképzelésektől teljesen eltérő irányba folytatódjék.

10.1.3 Bemenetek

Projekt alapító okirat

Háttér információkat nyújtó anyagok:

- Szervezeti, üzleti tervek;
- Szervezeti / üzleti célkitűzések;
- Szervezet felépítése (ábra);
- az informatikai taktikai tervből:
 - projekt portfólió;
- az informatikai stratégiai tervből:
 - informatikai stratégia kifejtése;
 - az irányítási és műszaki koncepciók és célkitűzések;
 - az informatikai stratégiai terv munkaanyagai.

10.1.4 Kimenet

Megvalósíthatósági tanulmány:

- Bevezetés;
- Vezetői összefoglaló;
- A tanulmány készítés megközelítés módja (SSADM testre szabása);

- A szervezeti tevékenységek támogatás a szervezet és az informatika oldaláról;
- A szervezet várható informatikai igényei;
- A javasolt rendszer, mennyiségi adatokkal, válaszidővel és áteresztőképességgel;
- A megvizsgált de elutasított alternatívák;
- Pénzügyi gazdasági elemzés;
- Projekt tervek;
- Következtetések és ajánlások;
- Műszaki mellékletek.

10.2 Megvalósíthatóság-elemzési modul (FS)

Ez a modul egyetlen szakaszból áll: 0. szakasz, Megvalósíthatóság.

10.3 0. szakasz: Megvalósíthatóság

A szakasz célja:

- megállapítani, hogy a javasolt információrendszer kielégítheti-e a szervezet által támasztott követelményeket;
- elkészíteni a javasolt információrendszer üzleti / befektetési indoklását, lehetővé téve a projektvezetőség számára a megalapozott döntést, hogy kössenek-e le további erőforrásokat a részletes tanulmány elvégzésére;
- megállapítani, hogy az informatikai stratégiában előírt irányoktól el kell-e térni;
- lehetővé tenni a projektvezetőség részére a megalapozott választást egy sor rendszer-szervezési és technikai alternatíva között, illetve segíteni a kiválasztott alternatívát megvalósító projektek kijelölését.

Leírás

A megvalósíthatósági elemzés röviden felméri, hogy a javasolt információrendszer ténylegesen kielégítheti-e a szervezet által támasztott szervezeti / működési követelményeket és gazdaságilag, pénzügyileg megindokolható-e egy ilyen rendszer létrehozása.

Minden projekt esetében a megvalósíthatósági elemzést a teljes tanulmány (követelményelemzés, követelmény-specifikáció, logikai rendszerspecifikáció) előtt ajánlott elvégezni, kivéve azokat, melyeknél a kockázat alacsony. Gyakran, de nem szükségszerűen, egy informatikai stratégiai tervezés után következik. Az elemzés határai sokszor túlmutatnak az SSADM-technikák és tevékenységek által kijelölt körön. Az SSADM-technikák elsősorban az információrendszer követelményeinek a meghatározásában és a technikai megvalósíthatóság kiértékelésében segítenek.

A jelenlegi és az igényelt környezetet csak olyan mértékben kell vizsgálni és leírni, hogy lehetővé váljon a *probléma megfogalmazása* és elfogadtatása, illetve a *rendszer szervezési és rendszertechnikai alternatívák* megjelölése.

Az elemzésben az elemző csoport tagjai, akik között projektirányítási és rendszer elemzési gyakorlattal rendelkezők is vannak, a felhasználók képviselői és bizonyos területekre szakosodott tanácsadók vesznek részt.

A modul tevékenységeinek előfeltételei

Vezetői döntések:

- Megegyezés a vizsgálat határaitól
- Megegyezés a probléma-megfogalmazásról
- a szóba jövő megvalósítási alternatívák megfogalmazása

Kiinduló anyagok:

Projektalapító okirat

Hivatkozott anyagok:

Szervezeti / működési célkitűzések

Szervezeti / üzleti tervek

Informatikai stratégia megfogalmazása

Informatikai stratégiai terv munkaanyagai

Irányítási és technikai politika

Szervezeti felépítés leírása

Projekt portfólió

Termékek

Megvalósíthatósági tanulmány

Technikák

Rendszerszervezési alternatívák kialakítása

Adatfolyam-modellezés

Dialógustervezés

Logikai adatmodellezés

Követelmény meghatározás

Rendszertechnikai alternatívák kialakítása

Tevékenységek

020 lépés: A probléma megfogalmazása

030 lépés: Megvalósíthatósági alternatívák kialakítása

11. Áttekintés a specifikációs prototípus-készítésről az SSADM 4+-ban⁶⁸

A prototípus annak a tervezési objektumnak a modellje, példaszerű megjelenítése, amely segít abban, hogy a fejlesztési folyamat végeredményeként előálló termék sajátosságait el lehessen képzelni. Egy számítógépes, automatizált rendszer fejlesztése során a prototípus általában képernyőket, menüket jelent, és csak korlátozott mértékben megvalósított funkcionális szolgáltatásokat, és azért hozzák létre, hogy a fejlesztő és a felhasználó között a fejlesztés különböző lépései során ennek alapján jöjjön létre egy megegyezés.

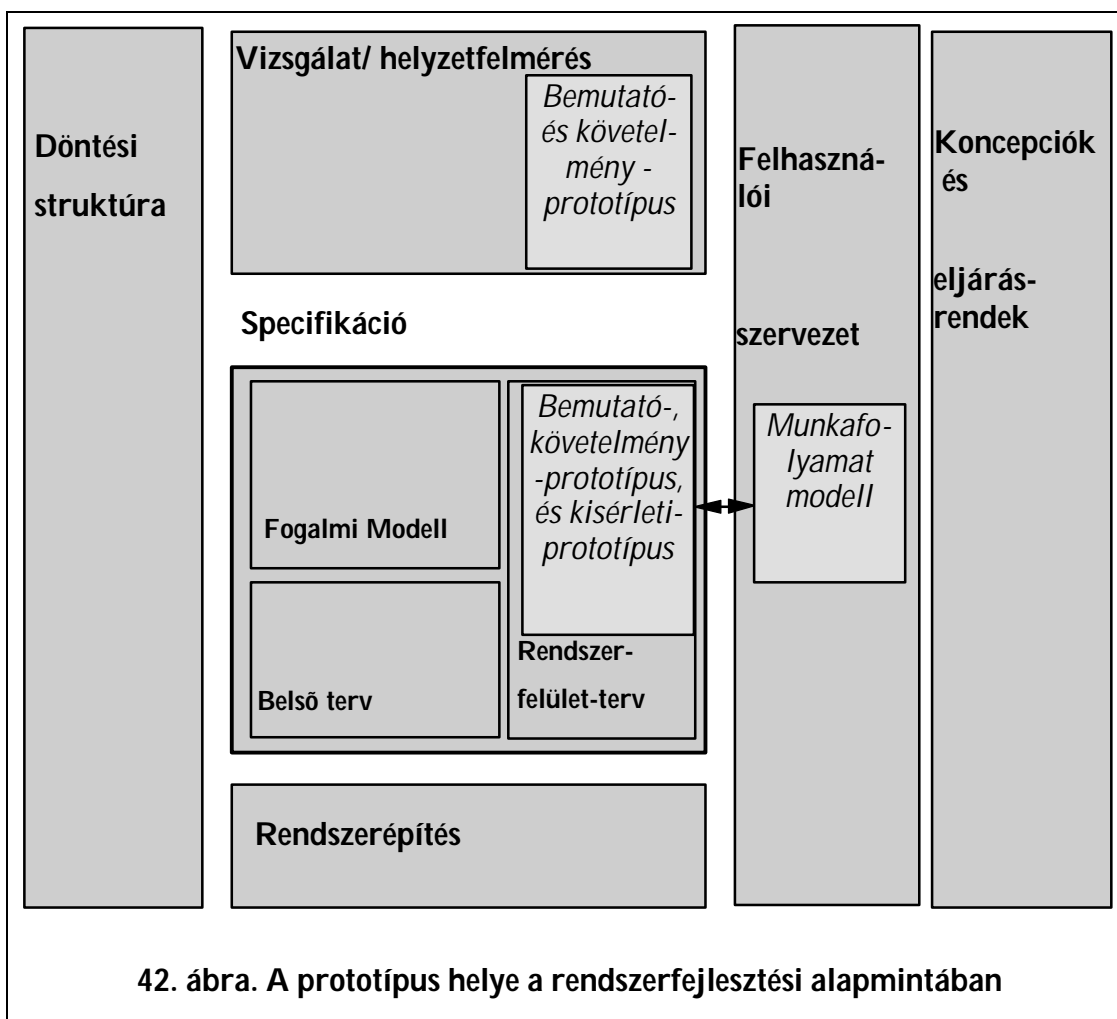
A prototípus a legkülönbözőbb formákban jelenhet meg, és különböző célokat szolgálhat. Vannak olyan prototípusok, amelyek a leendő rendszer első változataként jelennek meg, majd iteratív módon addig finomítják és bővítik a rendszert, amíg el nem érik a leszállítandó rendszerrel támasztott követelmények kielégítését. Más jellegű prototípusokat csak egy meghatározott célra használnak, majd utána el dobják.

Az SSADM-ben három fő célra használják a prototípus készítést:

- a leendő rendszerrel szemben támasztott követelmények rögzítésére;
- a rendszerfelület tervének kifejlesztésére;
- a fogalmi modell vagy a rendszertechnikai megoldások használhatóságának, a választott megközelítések életképességének bizonyítására.

A prototípus készítés helyét az ábra (42. ábra.) mutatja a rendszerfejlesztési alapmintában.

⁶⁸ [CCTA95], [CCTA95A], Reference Manual Part 7: Human Factors, Prototyping, 7-21—7-32, User Guide Part 3: Investigation, Prototyping in Investigation, 2-35—2-40. User Guide Part 3: Specification, Prototyping in Specification, 3-163—3-171, 'Prototyping in an SSADM Environment' ISE Library, HMSO, 1993. Továbbá [CCAT90].



A prototípus készítés nem alternatív módszere az SSADM-nek, és önmagában nem is egy 'Gyorsfejlesztési módszertan'⁶⁹, noha az ilyen jellegű módszertanoknak része a prototípus készítés.

Prototípus készítésnél a leendő felhasználók összes munkaköri feladatára figyelni kell, és nemcsak azokra, amelyek az automatizált rendszerrel kapcsolatba kerülnek. A prototípus készítés a munkafolyamat modellre fogja gyakorolni a legnagyobb befolyást.

11.1 A prototípus készítés

Egy SSADM-ben projektben négy fajta prototípus játszhat szerepet:

- Bemutatósi (demonstrációs). Ez a prototípus fajta a rendszer külső felületét mutatja be, hogyan fog kinézni, milyen érzés lesz használni. Általában egy önmagában álló feladatot jelent az elkészítése, korlátozott számú iteráció jöhet csak szóba. Ennek a prototípus készítési gyakorlatnak az eredményére alapozva lehet kifejleszteni az alkalmazás szintű környezeti útmutatót. Ezt a prototípust a vizsgálat / helyzetfelmérés szakaszában használják.

⁶⁹ Rapid Application Development (RAD), Isd. még DSDM (Dynamic System Development Method) [DSDM95]

- Követelmény. Ez a prototípus a követelmények tisztázására és a további követelmények feltárására szolgál. Ez az eljárás iteratív, azaz a prototípus készítés eredménye, a reagálások visszacsatolódnak a prototípus készítési folyamatba és újra bemutatják a felhasználóknak. A fizikai tervezés előtt bármikor sor kerülhet ilyen prototípus készítésre.
- Specifikációs. Ennek a prototípusnak a célja az, hogy azt érzékeltesse, hogy a rendszer mely része működőképes és használható és melyik nem. Célja az, hogy a követelményspecifikáció során előállított termékek hibáit felismerjék és azonosítsák, és ezen keresztül javítsák a minőségét;
- Kísérleti (K+F, kutatás + fejlesztési). Ez a prototípus arra szolgál, hogy bonyolult, vagy nehéz, szokatlan területeken megvizsgálják az informatikai rendszer alkalmazhatóságát, vagy a választott megközelítés életképességét. Voltaképpen a projekt során bármikor bevethető, de a legnagyobb szolgálatot akkor teszi, amikor a követelményeket már jól megértették és még a fizikai tervezés előtt. Azért készítik, hogy demonstrálják:
 - a választott megközelítés életképes;
 - összehasonlítsanak és kiértékeljenek különböző megközelítéseket.

A fenti négy prototípus fajta közül leginkább a specifikációs prototípus az, amelyik illeszkedik az SSADM módszertani megközelítéséhez.

11.2 A specifikációs prototípus áttekintése

A specifikációs prototípus készítéséhez kapcsolódó eljárások a követelményspecifikáció során létrejött termékek helyességét kívánják ellenőrizni, validálni.

Az idetartozó tevékenységek:

- a prototípusban megjelenítendő dialógusok kiválasztása;
- készítsék el a menük és a parancsok szerkezetét;
- tervezzék meg és készítsék el a prototípusát a menük és képernyők között lezajló információcserének és kölcsönhatásnak;
- mutassák be a felhasználóknak, akik vizsgálják meg, és véleményezzék;
- a szükséges változtatásokat hajtsák végre a prototípus tartalmában;
- aktualizálják a csatlakozó SSADM dokumentumokat;
- véglegesítsék a (követelmény)specifikáció tartalmát.

11.3 A specifikációs prototípus termékei

A prototípus készítés termékei — mivel jellegükből adódóan csak átmeneti, közbenső termékek az SSADM-en belül — nem jelennek meg az SSADM termék struktúrájában.

A specifikációs prototípus készítés során két fő termék típus jelenik meg:

- a prototípus készítés folyamatának dokumentumai: az ütemterv, a prototípus informatikai terve, a végrehajtás / bemutatás terve és az eredmények, vélemények rögzítése;
- az SSADM termékek munka példányai.

Továbbá az SSADM termékekben észlelt hibák és javításuk (illetve a kijavításukra hozott intézkedések, mint utómunkálatok).

A prototípus készítés folyamán használt dokumentumok:

- prototípus-bejárési út;
- a prototípus bemutatás céljai (prototípus-bemutatói célkitűzés);
- az eredmény rögzítő napló (prototípus-bemutatói eredménynapló);
- a prototípus kiértékelő jelentés (prototípus-kiértékelés).

A prototípus készítés során használt SSADM termékek munkapéldányai:

- menüszerkezet / hierarchia;
- parancsszerkezet.

A valószínűleg aktualizálandó SSADM termékek:

- követelményjegyzék;
- adatjegyzék;
- az igényelt rendszer logikai adatmodellje;
- funkció meghatározások;
- B/K struktúra;
- szerepkör/funkció táblázat.

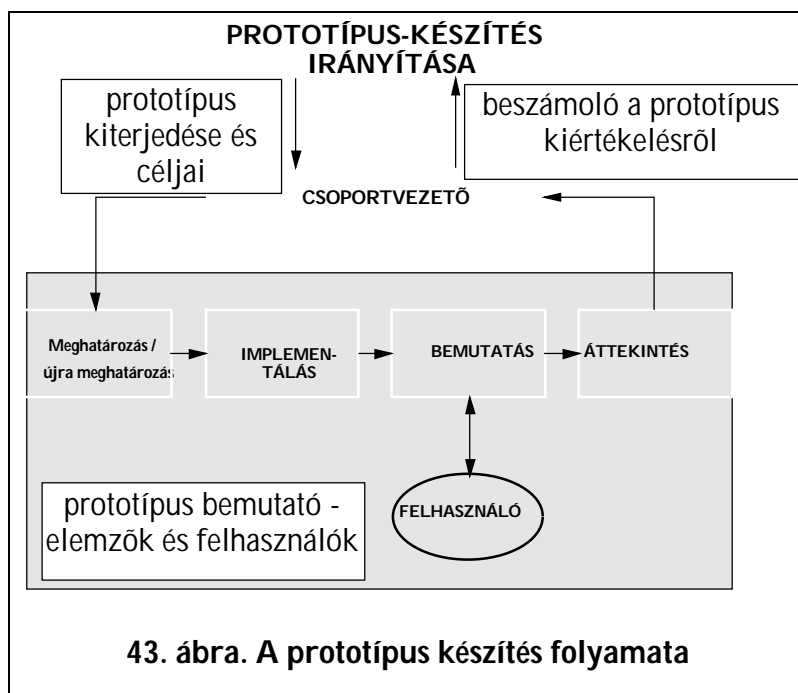
12. Specifikációs prototípus-készítés

12.1 A specifikációs prototípus készítésének kérdései

12.1.1 Eszközháttér kiválasztása

A prototípusokat a munkához legjobban illeszkedő eszközzel kell megvalósítani, amely nem feltétlenül azonos a rendszer esetleg szóba jövő megvalósítási környezete (hardver és szoftver), mivel az ezen a ponton valószínűleg még nem ismert (ideál tipikus projekt lefutásnál — a valóságban gyakran a projekt elején, vagy a rendszerszervezési alternatívák kiválasztásánál már megtörténik a rendszer technikai feltételeinek tisztázása).

A specifikációs prototípus elkészítéséhez ajánlott eszköznek rendelkeznie kell képernyők, menük és jelentések formáját előállító, megjelenítő lehetőségekkel, valamint interaktív, párbeszédés menük és dialógusok közötti közlekedést lehetővé tevő szolgáltatásokkal és aktív adatszótárral. További hasznos sajátosságokat jelent az alkalmazás logikájának szimulálása, az alkalmazás adattárolásának szimulálása és a prototípus verziók követését segítő szolgáltatások.



Mivel a specifikációs prototípus néhány ábraserkezetet tartalmazó termékre alapul (pl. igényelt rendszer logikai adatmodellje), ha ezeket a termékeket egy CASE eszköz segítségével állították elő, akkor célszerű, hogy a prototípus készítésének eszköze legalábbis össze tudjon kapcsolódni ehhez a CASE eszközhöz, azaz az eredményeiket valamilyen formában át tudják venni nem túl nagy erőfeszítéssel.

A támogató eszköz kiválasztásának a projekt életében lehetőleg korán meg kell történnie, azaz amint a prototípus készítésének kérdése eldőlt, a vezetésnek el kell kezdenie vizsgálni a prototípus készítésének kiterjedését és a megvalósítás eszközét.

12.1.2 A prototípuskészítés szükségességének megállapítása

Egy projekt kezdetén el kell dönteni azt, hogy szükség van-e a fejlesztési tevékenységen belül prototípust készíteni. Egy sor feltételt meg kell vizsgálni ahhoz, hogy eldöntsék, van-e valamilyen haszna a prototípus készítésének. Ezt a döntést — PRINCE értelmében — a projekt vezetőségnek kell meghoznia.

12.1.2.1 Prototípuskészítésre alkalmas projektek

Minden projektben meg kell vizsgálni, hogy mennyire igazak a következő kijelentések, vagyis vajon:

- a felhasználó követelményeit megfelelően értelmezték-e, konkrétan, a logikai adatmodell és a funkcióleírások hitelesen tükrözik a rendszer által kiszolgált felhasználók közösségének jelenlegi üzleti/működési igényeit,
- a rendszernek sokféle adatot kell-e kezelni,
- a felhasználói telephelyek földrajzilag szétszórta-e, ami helytől függő egyedi követelményeket is jelenthet,
- a nem megfelelő tervezésből eredő szervezeti / működési vagy technikai kockázatok magasak-e,
- a rendszer kifejlesztése és megvalósítása nagy pénzügyi befektetést igényel-e,
- a felhasználói szervezetek struktúrájának jelentős átalakítása várható-e,
- a felhasználók munkamódszereiben nagy változások várhatók-e,
- sok lehetséges változata van-e a tervezett megoldásoknak,
- a felhasználóknak vannak-e a számítógépes rendszerekkel kapcsolatos tapasztalataik, ami miatt az informatikai igényekhez nehezen tudnak alkalmazkodni, a követelmények specifikálását nehéznek találhatják,
- a projekt elemzői valójában járatosak-e az adott szervezeti működési területen, nincsenek komoly tapasztalataik róla.

Ha egy projekt megfelel a fentiek valamely kombinációjának, akkor a specifikációs prototípus készítése hasznos lehet.

Képernyő prototípusok

A képernyők prototípusainál meg kell vizsgálni a következőket:

- a rendszeren belül azoknak az interaktív tevékenységeknek a súlyát, gyakoriságát, melyek valószínűleg bekerülnek a rendszerbe. Ha sok képernyőn keresztül, párbeszédes információcsere várható, akkor a követelmények specifikációjának ellenőrzését célszerű prototípus készítésével leellenőrizni.

- a képernyőn belül használt adatelem mennyiségét. Ahol egy adott funkció interaktív tevékenységeinek mennyisége nagy, a prototípus elkészítése segít ellenőrizni, hogy az összes felhasználó igényt sikerült-e kielégíteni.
- a gyenge minőségű szolgáltatást nyújtó interaktív tevékenységből származó szervezeti / működési kockázatokat. Ha egy funkció hibás vagy lassan hozza létre a szükséges választ, akkor van-e ennek befolyása a szervezet (egészének) zavartalan működésére? Azokban az esetekben, amikor a szolgáltatás kritikus a szervezet működése szempontjából, akkor ajánlatos a funkcióhoz vagy funkciókhoz prototípust készíteni azért, hogy a követelményeket a lehető legjobban ellenőrizni lehessen.

Jelentések kimenetének prototípusai

A jelentések kimenetének prototípusaihoz meg kell vizsgálni a következőket:

- Jelentések, beszámolók kimeneti adatszerkezetének, formájának, adatelem halmazának ellenőrzése, ha a kimenetet egy másik rendszer fogja felhasználni. Ha egy adott kimenet formátumának meg kell felelnie egy másik rendszer bemeneti igényeinek, akkor célszerű, hogy a prototípus segítségével leellenőrizzék a két specifikáció összhangját.
- Jelentés kimeneti formátumának ellenőrzése a külső előírások, szabályok betartása miatt. Bizonyos kimeneteknek, például adózással kapcsolatos információknak (ÁFA bevallás, stb.), meg kell felelniük bizonyos külső előírásoknak és egy prototípus segíthet ennek az ellenőrzésében.
- A felhasználói követelmények meghatározásának helyessége. Ha a követelmények homályosak, nem érthetőek, kevésbé specifikáltak vagy nehezen megvalósítható igényeket tükröznek, a prototípus készítése segíthet.

12.1.2.2 Prototípuskészítésre nem alkalmas projektek

Az SSADM-en belül általában nem használható a prototípuskészítés a következő projektekben:

- az igényelt rendszer a meglévő rendszer egyszerű és közvetlen átalakítása az új rendszerré. Ez általában akkor fordulhat elő, ha a jelenlegi rendszer támogatja a szervezet működését, és a rendszer újrafelhasználása a jelenlegi szoftver és / vagy hardver változása miatt szükséges, pl. újabb technológiára való áttérés következtében.
- a projekt nem viseli el a prototípus készítésével kapcsolatos további fejlesztési kiadásokat.

12.1.2.3 Projektek, amelyeknél nincs jelenlegi rendszer

Nem kell, hogy létezzen jelenlegi rendszer ahhoz, hogy prototípust készítsenek. Sőt, ahol nincs jelenleg működő rendszer, ott a prototípus fontosabb szerepet játszik, segít a felhasz-

nálóknak a követelményeik megfogalmazásában és az elemzőknek az igények és a lényeg megragadásában.

12.1.3 Prototípuskészítés irányítása

A prototípus készítésének egyik nagy kockázata, hogy tervezés és kemény kezű irányítás hiányában a prototípusok készítésének eljárása végtelen ismétlésekbe torkollhat, és így elfelejtődik az, hogy milyen előnyök, elérhető haszon érdekében kezdtek el. Bár a prototípusok készítése kevésbé szigorú ellenőrzést igényel, mint más SSADM technikák, fontos néhány alapvető ajánlást figyelembe venni.

A vezetőségnek előre meg kell határoznia a prototípus készítésének kiterjedését. A kiterjedésnek nemcsak a prototípus készítésének terjedelmét kell meghatároznia (azaz azt, hogy a specifikáció mely részeit kell bevonni, SSADM termékek értelmében), hanem ütemezést kell készítenie az elvégzendő tevékenységekre, pontos célokat kell megfogalmaznia és meg kell határozni az erőforrás felhasználást, emberi / szakmai és hardver / szoftver értelemben.

A prototípust készítő csoport

A csoportnak egy vezetőből és legalább két elemzőből kell állnia. A vezető felelős a következőkért:

- a prototípus kezdeti kiterjedése meghatározásának átvétele a projekt vezetőségtől;
- a választott dialógusok / jelentések jóváhagyása;
- a prototípus bemutatók visszajelzéseinek átvétele és jóváhagyása;
- a visszajelzéseken alapuló döntések meghozatala, azaz további bemutatók engedélyezése vagy a prototípus készítésének lezárása;
- biztosítani a vonatkozó SSADM termékekre vonatkozó változtatási igények eljuttatását a megfelelő személyekhez;
- jelentés készítése a projekt vezetőségnek, jelezve az elért és a kihagyott célkitűzéseket és összefoglalva a munka eredményét.

Az elemzők a megvalósító/bemutató szerepköröket töltik be.

A prototípuskészítési ciklus

A prototípusok elkészítése a következő tevékenységek ismétléséből áll:

- Meghatározás /újra meghatározás
- megvalósítás
- bemutatás
- felülvizsgálat.

12.1.4 Prototípus készítésének előnyei és hátrányai

A prototípus készítésének lehetnek előnyei és hátrányai. A legnyilvánvalóbb hatása az, hogy a felhasználók tevékenyebb szerepet vállalnak mint egyébként, nélkülük a prototípusok készítésének folyamata értelmetlen.

12.1.4.1 Prototípus készítésének előnyei:

- felhasználói követelmények megerősítése,
- a lehetőségek fokozottabb megértése a felhasználók részéről,
- felhasználói elkötelezettség növekedése,
- bizalom növekedése.

12.1.4.2 Prototípus készítésének hátrányai:

- felfokozott felhasználói elvárások,
- a prototípuskészítési eszközökből eredő hiányosságok megjelenése,
- a projekt határainak megváltoztatása,
- túl sok prototípus-készítési ciklus,
- nem szabványos tervezés,
- dokumentáció hiánya.

12.2 A követelmény-specifikációs prototípus

A következő tevékenységek írják le a prototípusok készítését a követelmény-specifikáció kiválasztott részeihez:

- Környezeti útmutató beszerzése és alkalmazásának rögzítése;
- A prototípuskészítés kiterjedésének meghatározása;
- Kezdeti, kiinduló menüszerkezetek prototípusainak elkészítése;
- Menük, képernyők és jelentések prototípusainak elkészítése;
- Prototípusok bemutatása és véleményezése;
- Az SSADM dokumentáció megváltoztatása és beszámoló készítése a prototípus készítéséről.

12.2.1 Környezeti útmutató beszerzése

Sok szervezetben létezik 'Szervezeti szintű környezeti útmutató'. Ennek a kézikönyvnek a létezéséről meg kell bizonyosodni, és ha létezik egy példányt a projekt számára be kell szerezni azért, hogy hivatkozás kézikönyvként lehessen használni a

projektben. Ahol a 'Szervezeti szintű környezeti útmutató' meglehetősen általános, ott szükséges lehet egy testre szabott, a fennálló feltételekhez illesztett útmutató kialakítása, amelyet a prototípus készítés folyamán fognak használni. Természetesen az a célszerű, ha az itt alkalmazott környezeti útmutató és a megvalósított rendszerre alkalmazott környezeti útmutató megegyezik.

Ha a megvalósítás és / vagy a prototípus készítés eszköze⁷⁰ egy jelenlegi *de facto* szabványnak megfelel, akkor elképzelhető, hogy gyártók által kibocsátott útmutató egyszerűen megvásárolható, vagy megrendelhető, sőt ezeknek az eszközöknek lehet olyan része, amely kifejezetten a prototípus készítésre alkalmas alkönyvtárt is tartalmaz.

12.2.2 A prototípuskészítés kiterjedésének meghatározása

A prototípus fejlesztő csoport első feladata a modellezésre kijelölt részrendszeren kiterjedésének és határainak a megállapítása.

A munkafolyamat modellt meg kell vizsgálni azért, hogy azokat a feladatokat megjelöljék, amelyek végrehajtásáért bizonyos szerepkörök felelősek, és amelyek informatikai támogatásra tartanak igényt. A feladatok és a prototípus által lefedett területeknek meg kell egyezniük —az automatizált rendszer nem vizsgálható fekete lyukként, csak önállóan, elszigetelten a felhasználók teljes munkaköri feladatainak figyelembe vétele nélkül.

A "Funkciók meghatározása" során (330. lépésben) minden funkcióhoz létrehoztak egy B/K adatszerkezetet, valamint egy 'Felhasználói szerepkör / funkció mátrixot', amelyből a kritikus dialógusok azonosíthatók. A kritikusként megjelölt dialógusokra meg kell vizsgálni azt, hogy kell-e rájuk prototípust készíteni. A felhasználók további dialógusokat jelölhetnek meg mint olyanokat, amelyek tartalmának az egyeztetése szükségessé teheti a prototípus készítést. Ezeket általában el kell fogadni, ha az idő- és költségkeretek ezt megengedik.

Ezen felül hasznos lehet a jelentések kimenetének modellezése is, ha léteznek olyan külső vagy belső előírások, amelyeknek meg kell felelni.

12.2.3 Kezdeti menüszerkezetek prototípusainak elkészítése

A megállapított kiterjedésnek megfelelően ki kell alakítani a menü- és parancsszerkezeteket és meg kell valósítani őket a támogató eszközben. Ezeket a dialógus tervezésben leírt módon kell elkészíteni. Mindegyik felhasználói szerepkörre el kell készíteni a menü szerkezetet.

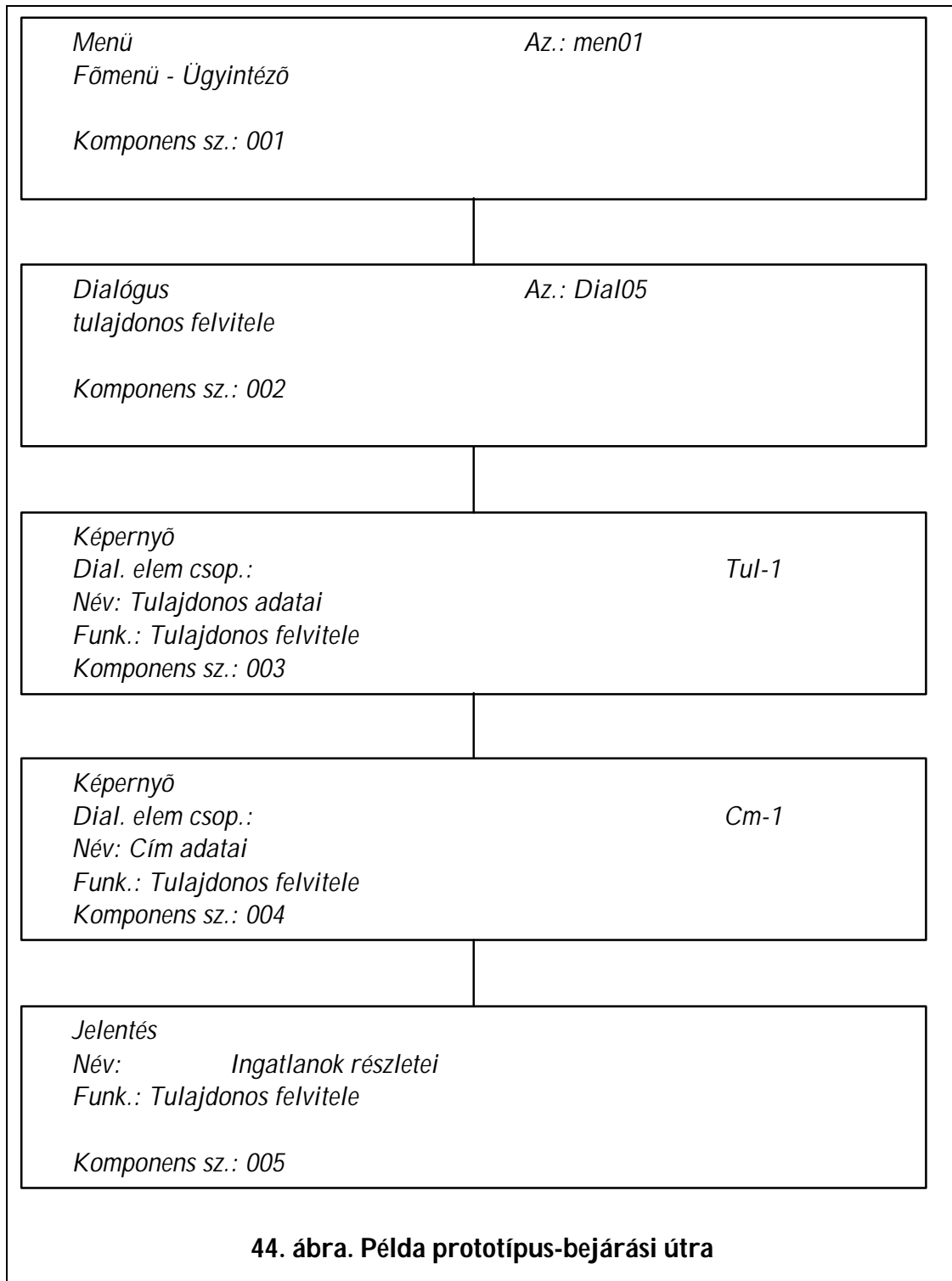
⁷⁰ Microsoft Windows, OSF / Motif, X-Windows: "The Windows Interface, an Application Design Guide", Microsoft Press (1992); "OSF / Motif Style Guide, Revision 1.2", Prentice Hall (1993), stb.

A menüket úgy kell megtervezni, hogy:

- egy adott felhasználói szerepkörre a teljes részrendszert ki kell fejleszteni;
- mindegyik kritikusnak megjelölt dialógust be kell venni a fejlesztésbe, illetve még a felhasználók által külön megjelölt dialógusokat.

A menüket közvetlenül a választott eszközben kell megvalósítani. A prototípusnak összhangban kell lenni a környezeti útmutatóban leírtakkal. Ha egy gyártó által készített útmutatót használnak illetve annak a prototípus készítési alkönyvtárát, akkor sokszor rendelkezésre áll egy olyan fejlesztő környezet, vagy szoftver sablon, amelyből a menüket nagyon gyorsan elő lehet állítani.

A létrejött menü prototípusokat be kell mutatni az illetékes felhasználóknak, jelezve a csoport vezetőjének a felmerülő változtatási igényeket. Ilyenkor meg kell vizsgálni, hogy van-e esetleg a háttérben meghúzódó probléma (pl. a felhasználói szerepkör/funkció mátrix kialakításában) vagy egyszerű felhasználói igényről van csak szó. Az elfogadott változtatásokat a kísérő dokumentációba (menüszerkezetekbe) is át kell vezetni, módosítva szükség esetén a követelményjegyzéket és a felhasználói szerepkör/funkció mátrixot is. A prototípus módosítása után újra be kell mutatni az illetékeseknek.



12.2.4 Menük, képernyők és jelentések prototípusainak elkészítése

Minden kijelölt dialógust vagy jelentést egy prototípus-bejárési út formájában kell meghatározni, ami felhasználói szerepkörönként mutatja a dialógus vagy jelentés útját a prototípuson belül. Az út tartalmazza az összes felhasználói felületre vonatkozó követelményt, a rendszer kiindulópontjától (főmenütől) a dialógus vagy jelentés végrehajtásának befejezéséig. Ez egy olyan munkaanyag, amely magas szinten írja le azt,

hogyan a menük, dialógusok és jelentések hogyan kapcsolódnak egymáshoz a prototípuson belül.

Ott ahol szervezeti tevékenység modell készült, ez a modell tartalmazza a szervezeti tevékenységek koherens vonalait, csoportjait és erre alapozva lehet elkészíteni a prototípus-bejárési utat.

A képernyők komponenseinek azonosításához a dialóguselemek logikai csoportjait kell felhasználni, amelyeket a B/K adatszerkezetek alapján lehet kialakítani. Ennek a technikáját a dialógus tervezés tárgyalja részleteiben. A jelentések kimenetét alkotó komponenseket a B/K adatszerkezetek kimenő adatelemei alapján lehet azonosítani.

Megjegyzés: A prototípus készítés folyamán használt képernyő, illetve képernyő alkotórészek fogalma logikai képernyőket jelentenek és nem a fizikai megvalósításukat, azaz logikailag összetartozó dialógus elemeket jelölnek. A megvalósításkor, pl. a prototípus készítés során, egy logikai képernyő több fizikai képernyőn jelenhet és ez megfordítva is fennáll.

12.2.4.1 Prototípus-bejárési utak létrehozása

A képernyő és jelentés komponenseinek azonosítása után ezeket a komponenseket össze kell illeszteni a meglévő menükkel, létrehozva a prototípus-bejárési utakat. Egy ilyen út leírása szögletes dobozokból és a dobozokat összekötő függőleges vonalakból áll. Minden doboz egy menüt, képernyőt vagy jelentést jelöl. Egy befejezett Prototípus-bejárési út megmutatja azt, hogy a felhasználónak hogyan kell bejárnia a dialógusokat, a neki szóló menü pontok kiválasztásával, és az egyes dialógusok lezárásával.

12.2.4.2 Prototípus-bejárési utak megvalósítása

Menük prototípusai

A menük prototípusai már készen vannak ezen a ponton, így keretként használhatók a képernyők és jelentések komponenseinek megvalósításánál.

Képernyők és jelentések prototípusai

A képernyők terveinek a szervezet szintű környezeti útmutató szerint kell készülniük és olyan könnyen követhetőknek kell lennie amennyire az csak lehetséges. A feladat az, hogy a lehető legkönnyebbé tegyék a képernyők kezelését azért, hogy a felhasználók a tartalom helyességére tudjanak koncentrálni és ne kelljen azok formájával bajlódniuk. Természetesen egy egyszerű képernyő tervet egyszerűbb is elkészíteni. Általános elvek:

- áttekinthető és jól strukturált legyen;
- az adatok bevitelét felülről lefelé és balról jobbra haladó sorrendben engedje;

- tartalmazza az összes adatot az adatbevitel után következő adatfeldolgozáshoz.
A jelentések prototípusainál az azonosított kimenő adatelemeket meg kell feleltetni a prototípus kimeneteinek.

Képernyők és jelentések prototípusainak ellenőrzése

A képernyők és jelentések prototípusainak adatelemeit össze kell hasonlítani az igényelt rendszer logikai adatmodelljével és az adatelemek leírásaival. A származtatott vagy számolt adatelemeket külön adatelemként le kell írni. A kiszámítás módját le lehet írni az adatelemek leírásában, vagy le lehet írni mint közös használatú feldolgozási folyamatot a funkciók leírásához. Az elemi folyamatok leírásaira szükség esetén hivatkozni lehet.

12.2.4.3 Felkészülés a prototípus bemutatására

A bemutatás előtt minden prototípus-bejárési úthoz el kell készíteni egy prototípus-bemutatói célkitűzéseket tartalmazó dokumentumot, amely felsorolja minden menühöz, képernyőhöz és / vagy jelentéshez a feltételezéseket és a feltevésre váró kérdéseket (a bejárési út leírásában azonosított komponensekhez).

A bemutatáshoz szükséges adatokat is mindegyik dialógusra elő kell készíteni, és a prototípus ellenőrizni fogja az adatok teljességét.

Bármilyen észlelt hibát, hiányosságot ki kell küszöbölni és a dokumentációt aktualizálni ennek megfelelően.

12.2.5 Prototípusok bemutatása és véleményezése

A prototípusok modelljeit egy vagy több olyan felelős felhasználónak kell bemutatni, aki az adott prototípus-bejárési útban meghatározott felhasználói szerepkört tölti be. A bemutató során két dokumentumot kell használni:

- Prototípus-bemutatói célkitűzéseket, amely minden komponenshez tartalmazza a megbeszélendő kérdéseket.
- Prototípus-bemutatói eredménynapló, amelyben a bemutató eredményeit rögzítik.

Az eredménynaplóba rögzíteni kell a felhasználó által felvetett igényeket, a prototípus verzióján belül menüként és képernyő komponensenként csoportosítva. A bemutató után az eredménynaplót ki kell egészíteni az eredményekhez tartozó változtatási igényekkel, és meg kell hozni a döntéseket a végrehajtandó tevékenységekről.

A bemutató után a csoport vezetőjének el kell döntenie a következő kérdéseket:

- Elérte-e a prototípuskészítés hasznosságának határait, vagy további bemutatók hasznosak lehetnek még?
- Szükséges-e az eredetileg tervezett ráfordítandó időt meghosszabbítani, vagy további erőforrásokat bevonni? Ha igen, engedélyt kell kérni a vezetéstől.

- Vannak-e olyan problémák, amiket a vezetésnek jelenteni kell?

A csoport vezetőjének gondoskodnia kell arról, hogy minden szükséges változtatást a dokumentációban átvezessék.

12.2.6 SSADM termékek módosítása

A prototípus-készítési ciklus minden ismétlése újabb változtatási igényekkel járhat az SSADM más termékeire nézve is, például az igényelt rendszer logikai adatmodelljében okozhat változásokat. Ezeket a változtatásokat meg kell valósítani, és a prototípust újra be kell mutatni, annak a bizonyítására, hogy a gyakorlatban kivitelezhető, hasznos, és ez az, amit a felhasználó akar.

Az ellenőrzés után a csoport vezetőjének el kell juttatnia a változtatási igényeket az elemzőkhöz, akik helyzetüknél fogva jobban fel tudják mérni a változtatások hatásait. Az elemzők ezek után tájékoztatják a 3. szakasz szakmai irányítóját, aki elfogadja vagy visszautasítja a változtatásokat. Ösztönözni kell a spontán, gyors reagálásokat, változtatási igényeket, mivel megvalósításuknak előre nem látható jelentősége lehet, rámutathat például az elemzés hiányosságaira. Az is előfordulhat, hogy a prototípuskészítés közben jó ötletként elfogadott dolgok a szélesebb körben megvizsgálva nem tűnnek célszerűnek vagy hasznosnak.

Ha a felhasználókkal egyetértésben új követelményeket ismernek fel és fogadnak el, akkor a csoport vezetője ezeket közvetlenül felveheti a követelményjegyzékbe a prototípuskészítés lezárása után is.

A bemutatók végeztével minden végső változást fel kell venni a kísérő dokumentációba, valamint a követelményjegyzéket ki kell egészíteni bármely új vagy módosított követelménnyel, ami a prototípus-készítési tevékenységből eredt.

A csoport vezetőjének jelentést is kell készíteni a vezetés számára. A következő kérdésekre kell válaszolni:

- Megmaradt a prototípuskészítés a vezetés által eredetileg meghatározott kiterjedés keretein belül?
- Elérte a prototípuskészítés a kiterjedést leíró dokumentumban megfogalmazott célkitűzéseket? Ha nem, miért nem? (Lehet, hogy ez inkább a célkitűzésekre vonatkozó reagálás és nem a prototípuskészítés minősítése, azaz a célkitűzések voltak értelmetlenek vagy rosszul meghatározottak. Az erre vonatkozó értékelés hasznos lehet a jövőben.)
- Milyen változtatások történtek a követelmény-specifikációban, azaz milyen új követelmények keletkeztek, melyek változtak, mely SSADM termékek módosultak a prototípuskészítés eredményeképp?
- Hasznos volt-e a prototípuskészítés mint tevékenység, vagy nem hozott hasznot? Van-e valami, amit másképp lehetett vagy kellett volna csinálni?

13. Az SSADM projektek irányításának kérdései

Az SSADM alkalmazása önmagában nem tudja garantálni a projekt sikerét. Csak a helyesen alkalmazott projektirányítási módszerek alkalmazásával tudja szolgáltatni az elvárt eredményt. Az SSADM csak az elemzési és tervezési módszerekkel foglalkozik, míg projektirányítási kérdésekkel a PRINCE⁷¹ módszertan foglalkozik. Az SSADM és a PRINCE módszertan kapcsolódási pontjait, néhány gyakorlati tanácsot fogunk megtárgyalni, amely segítheti a projektirányítót egy SSADM szerint végrehajtott projektben.

Az SSADM nem foglalkozik a következő kérdésekkel, de egy projekt végrehajtása során figyelni kell ezekre:

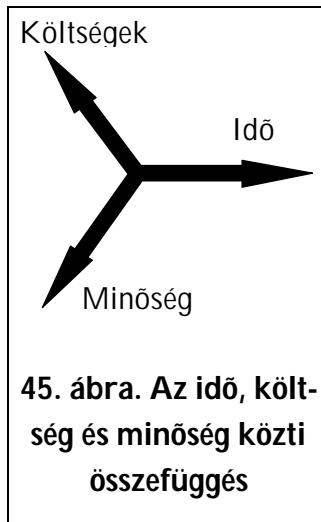
- a megközelítési mód kiválasztása
- a projekt indítása, formális kezdeményezése;
- a projekt szervezetének kialakítása;
- a projekttervezés, az ütemtervek, kivitelezési terv kialakítása;
- minőségirányítás;
- a projekt előrehaladásának a nyomon követése és ellenőrzése.

13.1 A megközelítési mód kiválasztása

Az SSADM nagyon rugalmas módszertan és ezért a legkülönbözőbb életciklus modellekkel lehet együtt használni (Isd. 13.10 SSADM alternatív életciklusok, testre szabási lehetőségek), továbbá a legkülönbözőbb alkalmazási környezetekhez és változatos méretű információrendszer készítési igényekhez is lehet illeszteni.

A projektirányítónak általában szembe kell néznie azzal az igénnyel, hogy a lehető legrövidebb időn belül egy jó minőségű rendszert kell leszállítania a lehető legalacsonyabb költségekkel.

⁷¹ Isd. [CCTA91]



Ez a három tényező egymással ellentétben áll, ezt próbálja az ábra érzékeltetni (45. ábra.). Bármelyik tényező megváltoztatása hatással van a másik kettőre:

- a ráfordítandó idő csökkentése a minőség változatlan fenntartása mellett a költségek növekedéséhez vezet;
- a ráfordítandó idő csökkentése és a felhasználható pénzek korlátozása gyenge minőségű termékhez vezet;
- ésszerű költség keretek és magas minőségi követelmények mellett a projekt időigénye nagy lesz.

13.2 Gyors alkalmazás fejlesztés⁷²

A gyors alkalmazás fejlesztés egyre népszerűbbé válik bizonyos területeken. Noha nincs szakmai konszenzus vagy ipari szabvány arra vonatkozólag, hogy mi tekinthető gyors alkalmazás fejlesztésnek, de általában azt értik ez alatt, hogy amilyen gyorsan csak lehet az igényeket minimálisan kielégítő rendszert állítsanak elő, általában a protípus készítés nagy mértékű alkalmazásával.

Ennek a megközelítési módnak az alkalmazásához azonban több előfeltételnek teljesülnie kell:

- a felhasználók intenzív és aktív részvétele (a kijelölt felhasználói képviselők munkaidejük 80-90%-t ezen a projekten kell tölteniük, akár tetszi ez a szervezeti egységek vezetőinek, akár nem);
- hatékony fejlesztő eszköz, amely lehetővé teszi az eredményes fejlesztést (integrált CASE, adatbáziskezelő, alkalmazás generátor, kódgenerátor), valamint automatikusan szolgáltatja az előírt rendszer dokumentációt;
- prototípus készítési gyakorlat, amely a kijelölt projekt tartomány lényeges elemeire koncentrál, és nagyon gyorsan képes eredményt létrehozni;
- az adott feladatra már létezik automatizált rendszer, melyet előképként, mintaként fel lehet használni az iteratív prototípus készítési folyamatnál;
- projektszervezet, keménykezű projektirányítóval.

Ennek a megközelítésnek a legfontosabb eleme az időkorlát⁷³, amely megszabja a feladat végrehajtási idejét és amelyet nem lehet áthágni, megsérteni (míg esetleg az egyéb paraméterek óhatatlanul sérülnek). Ez a fejlesztő csoport tagjait arra bátorítja, hogy valóban csak a legfontosabbra koncentráljanak, csak azokra az igényekre, amelyek a szervezet számára vala-

⁷² Rapid Application Development (RAD)

⁷³ development time constraint, Timebox

milyen haszonnal járnak, és elkerüljenek minden a szervezet / üzlet szempontjából fölösleges fejlesztést.

A gyors alkalmazás fejlesztést olyan feladatokra lehet használni, amelyek nem vesznek igénybe 3-6 hónapnál több időt, rendelkezésre áll megfelelő fejlesztő eszköz, beleértve a prototípus készítő eszközt is. Egy viszonylag kis fejlesztő csoportra van szükség, és olyan felhasználókra, akik lelkesen, teljes erővel vesznek részt a projektben.

Egy ilyen projekt fejlesztési megközelítésben gyakran előforduló technika a közös alkalmazás fejlesztés⁷⁴, amelyben közös munkaértekezleteket használnak, a felhasználók és a fejlesztő álláspontjainak egyeztetésére, a követelmények közös megfogalmazására. Ezeket az üléseket általában egy elnök, aki moderátori szerepet tölt be, vezeti le. Ezek a találkozók teszi lehetővé a szervezet szereplői számára, hogy azonosítsák a kérdéseket, problémákat, feloldják a konfliktusokat meghatározzák a követelményeket és rangsorba állítsák. Azért, hogy ezek a megbeszélések sikerrel járjanak a felhasználókat megfelelő felhatalmazással kell ellátni ahhoz, hogy a megfelelő döntéseket meghozhassák, és így ne pazarolják az időt fölösleges interjúkra és ezeket kiértékelő, véleményező és összegző megbeszélésekre.

Az SSADM-t, ha szükséges, lehet a gyors alkalmazás fejlesztési környezethez illeszteni, a 3-séma specifikációs architektúra megfelelő technikáit kell áttemelni, és a közös alkalmazás fejlesztéssel valamint az időkorlátos megközelítéssel kell ügyesen kombinálni.⁷⁵

13.3 A projekt indítása

Bármilyen projektről is legyen szó, az első lépés a projekt formális indítása. A tapasztalatok azt mutatják, hogy a projekt indításakor a tervezésre és a projekt alapvető célkitűzéseinek⁷⁶ meghatározására fordított idő busásan megtérül a későbbiekben. A projekt indításakor meg kell határozni a projekt szervezetet, a projekthez rendelt személyeket, valamint a fejlesztéshez szükséges infrastruktúrát, amely a projekt sima lefutásának elengedhetetlen feltétele.

Egy formális projekt alapításhoz tartozik:

- a projekt határainak és kiterjedésének (termék halmazának értelmében) meghatározása;
- kockázatokat, költségeket és a projektből származó előnyök és hasznok elemzése;
- a projekt sikeres befejezéséhez szükséges feladatok és termékek meghatározása.

Ebben a szakaszban kell elvégezni az SSADM testre szabását is. A testre szabáskor hozott döntéseket, indokaikat, a járulékos kockázatokat részletesen dokumentálni kell.

⁷⁴ Joint Application Development, JAD

⁷⁵ [CCTA94] {CCTA, 'Customising SSADM', ISE Library, HMSO, 1994.}, [Hargrave96].

⁷⁶ hivatkozási alap, Terms of Reference, ToR.

13.3.1 A projekt indítás tevékenységei

A következő tevékenységek tipikusan a projekt indításhoz tartoznak:

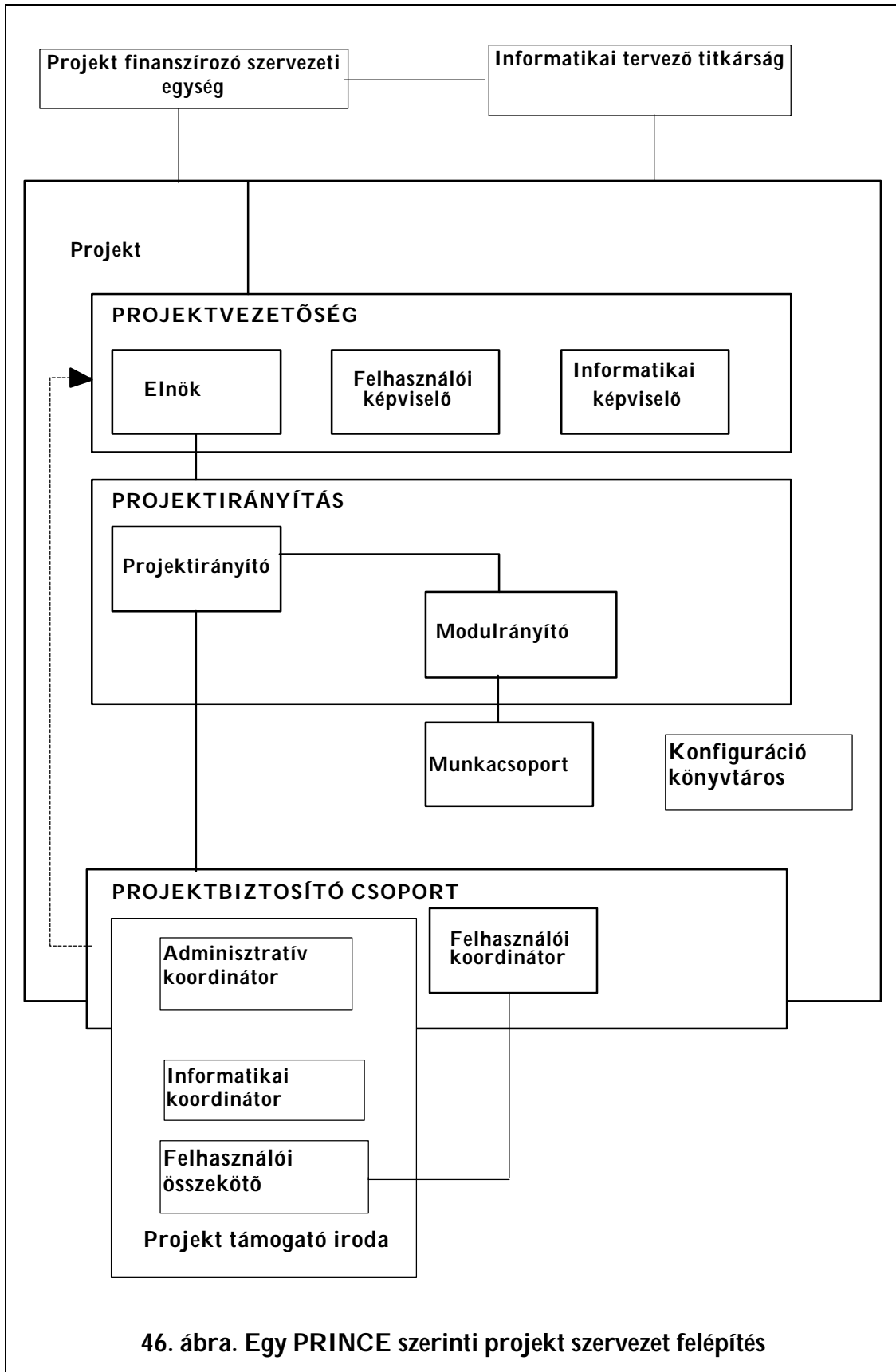
- a projekt alap paramétereinek meghatározása:
 - erőforrások;
 - a projekt határai és kiterjedése, mérete és bonyolultsága;
 - a célkitűzések pontos megfogalmazása.
- a projekt szervezet felállítása:
 - PRINCE féle projekt szervezet;
 - helyi, szervezeti előírás;
 - a projekt szerepkörök személyekhez rendelése;
 - a felhasználók képviselőinek kijelölése, tájékoztatása.
- a kockázatok, költségek, hasznok vizsgálata
 - műszaki, szervezeti / üzleti, biztonsági kockázatok elemzése és az ellenintézkedések megtervezése.
 - hatáselemzés
 - költség / haszon elemzés
- projekt tervek elkészítése
 - a strukturális modell illesztése, a döntések és indokaik dokumentálása;
 - a projekt elkészítendő termékeinek megállapítása az SSADM termék szerkezetére alapozva.
 - a projekttervek elkészítése a szervezet helyi előírásainak megfelelően (hálóterv, erőforrásterv, Gantt diagram);
 - az projekt elindítására a jóváhagyás megszerzése
 - a projekt alapító okirat elkészítése után a projekt vezetőség formális egyetértésének elnyerése.

13.4 A projekt szervezete

Az egyik lehetséges projekt szervezet felépítés az, amit a PRINCE javasol (46. ábra.). A projekt szervezet szerepköreit a PRINCE részletesen tárgyalja⁷⁷. A felhasználói összekötő akkor kap jelentőséget, ha a felhasználói koordinátor teljesen járatlan az informatikai kérdésekben, és nincs semmi tapasztalata információrendszert készítő projektek végrehajtásában.

⁷⁷ [CCTA91], CCTA (Central Computer and Telecommunication Agency), *PRINCE , Structured Project Management*, NCC Blackwell Ltd., Manchester, Oxford, 1991.

Ilyen esetben a felhasználói összekötő segítséget és útmutatást nyújt a felhasználói koordinátornak az informatikát érintő kérdésekben.



13.5 A tervezés

A terv azt az elkötelezettséget fogalmazza meg, hogy a projekt meghatározott céljait, termékeit, az előírt időre, az előírt költséggel és minőségben hozzák létre.

Egy SSADM projektre vonatkozó tervnek a következőket kell tudnia:

- az egész projekt és minden egyes modul (szakasz) termékeit határozza meg az SSADM termékszerkezete alapján;
- minden termékhez rendelje hozzá az előállításához szükséges tevékenységeket;
- írja elő az alkalmazandó minőség ellenőrzési eljárásokat, a minőség ellenőrzés módját (támaszkodva az egyes SSADM termékekhez csatolt minőségi kritériumokra);
- határozza meg az erőforrás igényeket;
- a termékek előállításához szükséges időt állapítsa meg
- tartalmazzon egy a teljes projektre vonatkozó költség-felhasználási diagrammot;
- határozza meg a projektszerepköröket, a hozzájuk tartozó feladatokat, hatásköröket, felelőségeket és rendelje hozzá személyekhez;
- tartalmazza azokat az eszközöket (pénzügyi, szervezési, hatalmi), amelyek segítik a fejlesztő csoport felállítását és az egyes célok elérését;
- a terv segítse a projekt előrehaladásának ellenőrzését és az ellenőrzési pontok meghatározását;
- a projekt szereplői, résztvevői közötti kommunikációt segítse a terv, annak egyik eszköze legyen.

Egy projekt terv általában a következő két főrészből áll:

- a **kivitelezési tervből**^{7B}, amely az egyes termékek előállításához szükséges tevékenységek, feladatok ütemezését jelenti;
- az **erőforrás tervből**, amely megmutatja azt, hogy az egyes erőforrásokból mennyire van szükség és az mennyibe kerül a kivitelezési terv sikeres végrehajtása érdekében.

A terv készítéséhez szükséges anyagok:

- a termékszerkezet ábra:
 - irányítási termékek, minőségi termékek és informatikai / műszaki termékek (SSADM termékek);
- a termékszármaztatási ábra;

- termékleírások;
- tevékenységek (SSADM alap strukturális modellje).

A tervszintek:

- projektszintű tervek;
- modulszintű tervek;
- szakaszszintű tervek.

13.6 A minőség tervezése

A minőség tervezésekor a következőket kell figyelembe venni:

- milyen szabványokat, szabályokat, előírásokat kell érvényesíteni, amelyek lehetnek helyiek, vagy a szervezeten kívüliek;
- az előírásokat betű szerint vagy a szellemükben kell alkalmazni, azaz bizonyos értelemszerű eltérések megengedettek-e;
- milyen minőségi szemlékre van szükség az egyes termékek véleményezéséhez, milyen legyen a szemle eljárása és a jóváhagyás módja;
- mi legyen a hibákat korrigáló eljárások rendje, a tesztelések, párhuzamos futtatások, stb. esetén.

A minőségi tervezés eredménye a kivitelezési terv része lesz, és beilleszkedik a különböző szintű tervekbe. Arra is van lehetőség, hogy tulajdonképpen a projekttől független minőségi tervet hozzanak létre, ebben az esetben ez a terv kevésbé fog változni a kivitelezési terv többi részéhez képest.

Minden SSADM termékhez van a termékleírásban egy minőségi kritérium halmaz, különösen azokra a termékekre, amelyek valamilyen információt adnak át az egyes lépések között. Az egyes SSADM szakaszok végén a szakasz által kibocsátott végtermékekre egy átfogó minőségi szemlét kell szerveznie a projektirányítónak. Ez a szemle kiegészítené az egyes termékekre végrehajtott egyedi minőségi szemléket és az a célja, hogy a szakasz záró értekezlet előtt leellenőrizze, hogy a szakasz végterméke mint egységes egész önellentmondásmentes-e és teljes-e.

13.7 A projekt előrehaladásának a nyomon követése és ellenőrzése

A projekt előrehaladását azért kell nyomon követni és ellenőrizni, hogy:

- vajon a szervezeti célkitűzésekkel összhangban áll-e a projekt, vagyis tartja-e az ütemtervet, és nem lépi túl sem a költség - sem az erőforrástervet;

⁷⁸ Műszaki terv, technikai terv (Technical Plan, Delivery Plan) is szokták nevezni

- vajon az összes termék teljesíti-e az előírt minőségi kritériumokat a termékleírásokban előírt módon.

A projektet nyomon követni a felhasznált erőforrásokon és az elkészült, minőségi szemléken sikeresen átment termékeken keresztül lehet.

Ellenőrizni az aktuális projekt állapotok és a tervezett teljesülések összevetésével lehet és ha ezek nincsenek összhangban akkor a szükséges korrekciós lépések megtehetőek. A tervezett teljesülés ellenőrzésébe természetesen beleértendő az előírt minőségi követelmények teljesülése. A cél az, hogy a lehető legkorábban észleljék a projekt tervektől való eltéréseket akkor, amikor a helyre hozataluk viszonylag még kis erőfeszítésbe kerül.

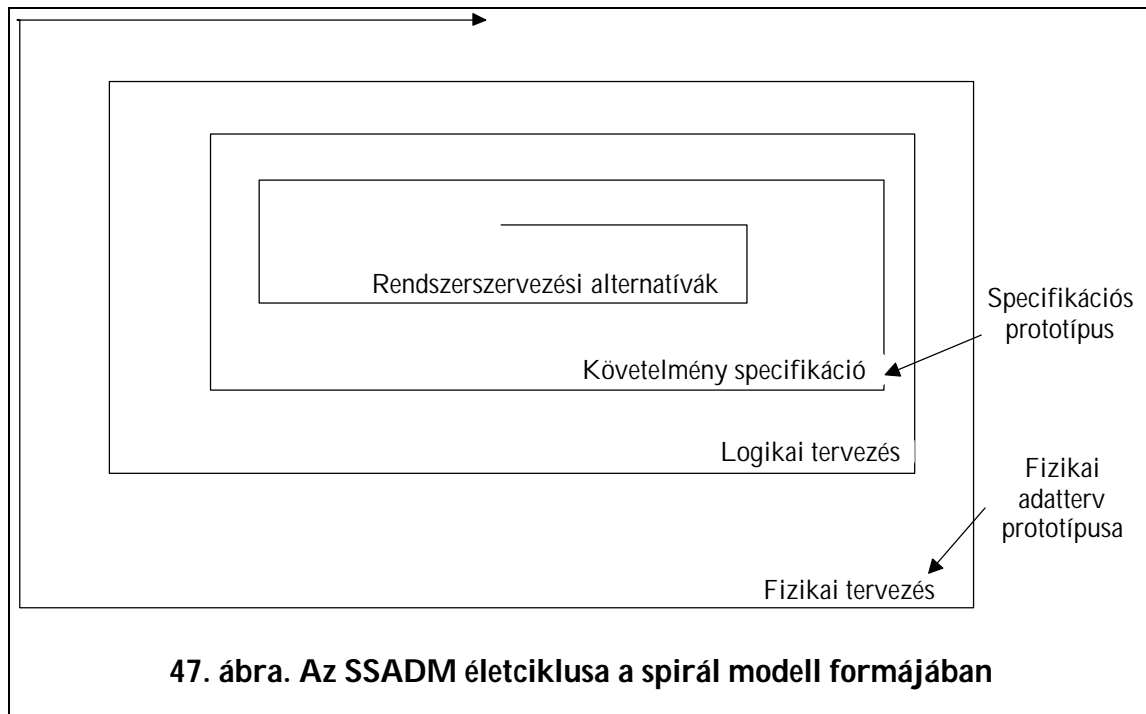
13.7.1 Ellenőrzési pontok

A PRINCE-ben előírt ellenőrzési pontok:

- Formális projekt indítás;
- SSADM modul közbeni ellenőrzés
 - előre tervezett a projekt terv szerint;
 - szükség szerint összehívandó, bizonyos problémák kezelésére;
- modul végi kiértékelés;
- formális projektzárás;
- munkaértekezletek:
 - a fejlesztő csoportok munkájának kiértékelése, a projekt előre haladására vonatkozólag. 1-2 hetente tartandó, havonta összefoglalót készít a projektirányító a projektvezetőség számára.
- tűrés, tolerancia mint irányítási eszköz:
 - a megadott tűrés határokon belül a saját hatáskörében dönthet a:
 - projektirányító a projekttervre;
 - a modul- / szakaszirányító a modul - / szakasztervre;
 - a projektvezetőség az egész projektre vonatkozóan.

13.8 Az SSADM és a spirál modell

Az SSADM strukturális modellje (Default Structural Model) nem követi szigorúan a vízésés életciklus modellt, de nagyon közel áll ehhez a megközelítéshez. A vízésés modellt "feltekerhetjük" egy spirál modell formájába. A specifikáció négy fázisa ábrázolható a spirál modellben.



Egy pl. SSADM projektet a spirál modell "Fejlesztés" nevű negyedébe helyezve egy olyan modellt kapunk, amely egy egy információrendszer teljes életciklusát leírja, azaz az rendszer evolúcióját az adaptív karbantartási ciklusokon keresztül.

13.9 Információrendszer adaptációk készítésének szakaszai

Látható, 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 (Isd. [CCTA95B], [Turner96], [Euromethod94]) ezt összefoglaló névvel információrendszer adaptációnak nevezi (IR-adaptáció információrendszer: adaptáció).

13.9.1 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 terméke*, a leendő információrendszerek terveinek egy portfóliója, ezt tulajdonképpen *tervezési terméknek* tekinthetjük.

13.9.2 Rendszerelemzés

Ez a szakasz a szervezet egy meghatározott működési területének helyzetét vizsgálja meg és egy helyzetfelmérési tanulmányt készítenek. 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.

13.9.3 Rendszertervezés

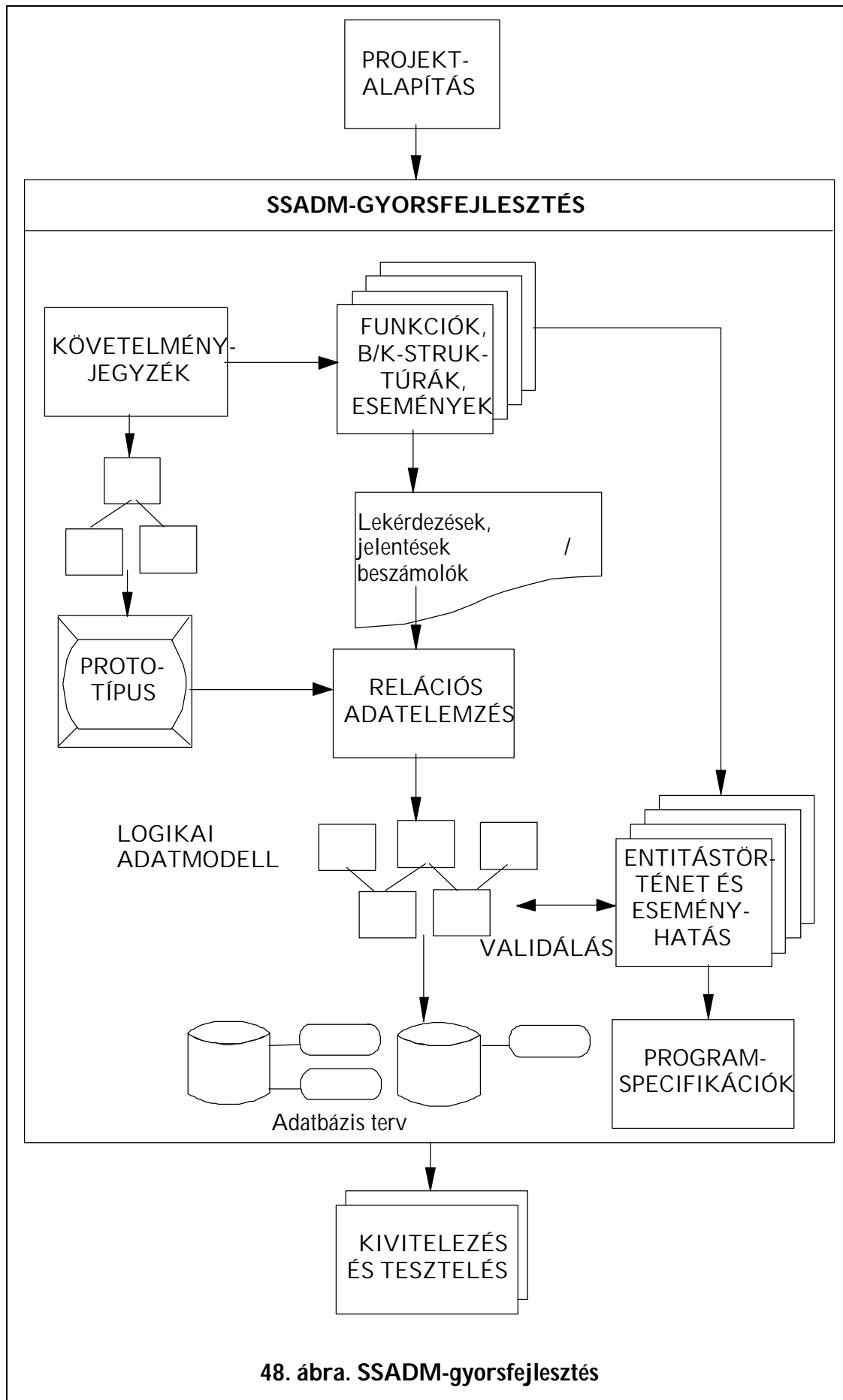
A rendszertervezési szakasz egy a leendő információrendszer vonatkozó előírást állít elő, általában elektronikus formában. Az alkalmazási terület kiterjedése sokkal szűkített, mint a megelőző rendszerelemzési szakaszban vizsgált területé.

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.

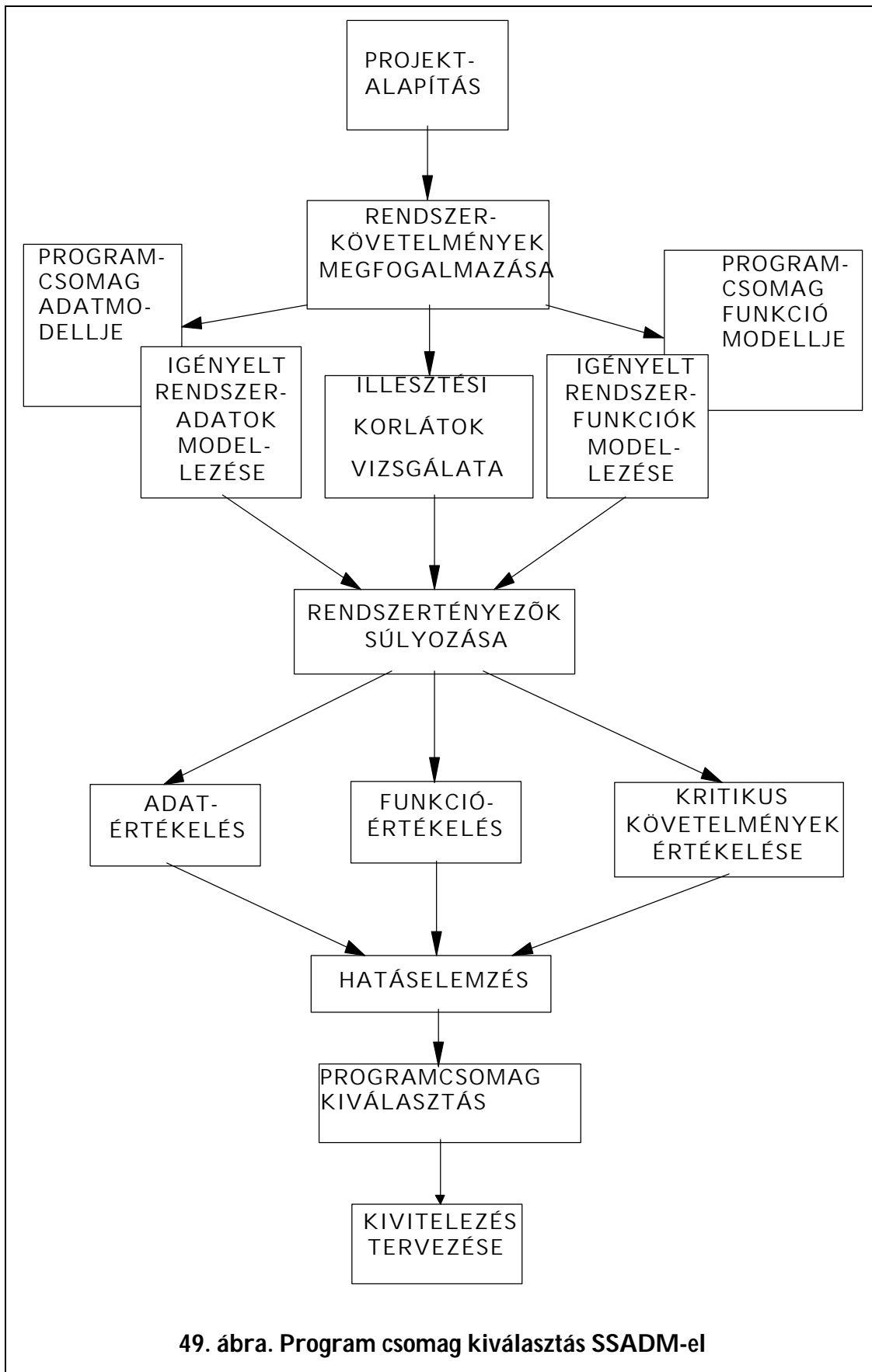
13.9.4 Rendszerkészítés (létrehozás)

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.



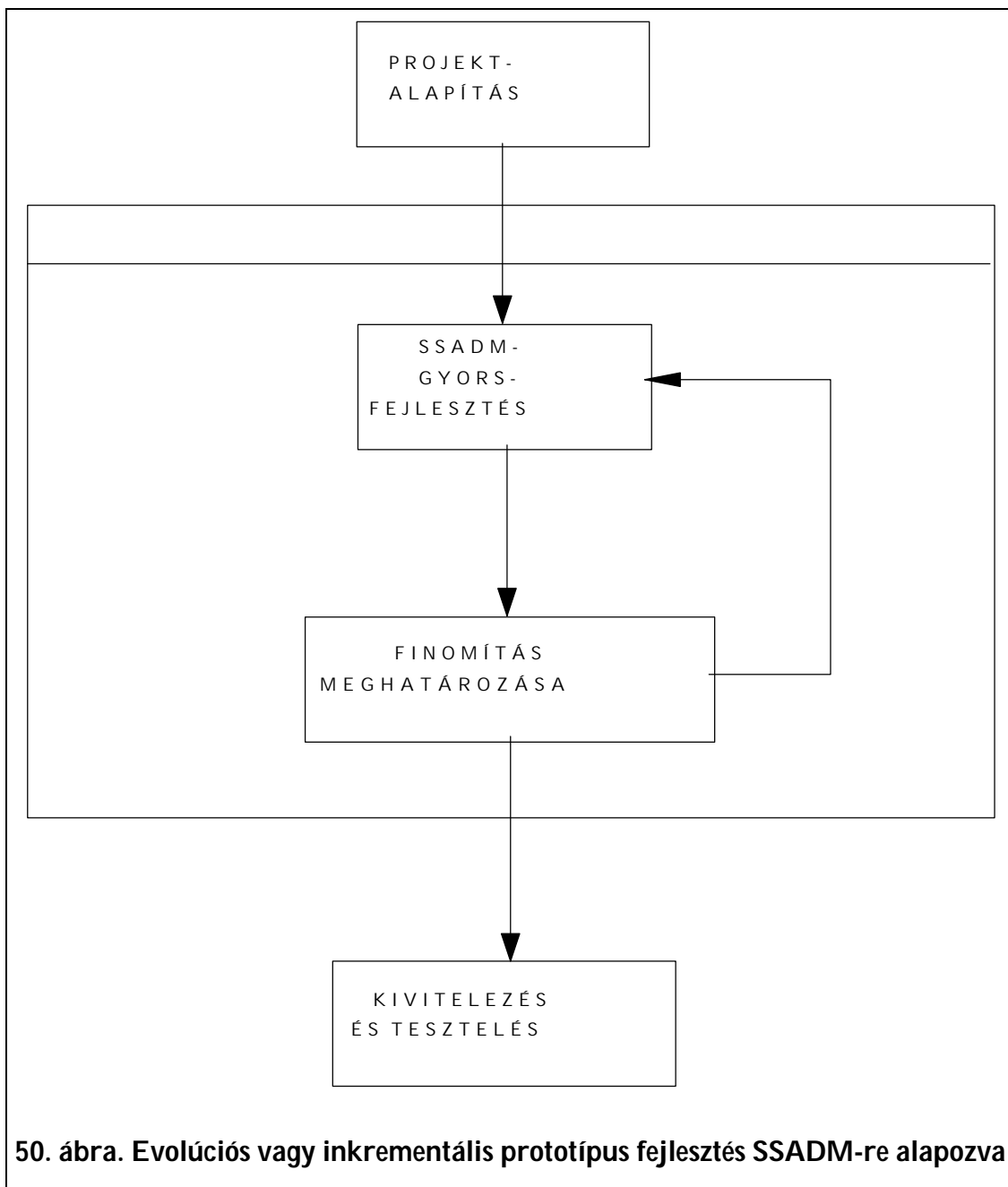
48. ábra. SSADM-gyorsfejlesztés



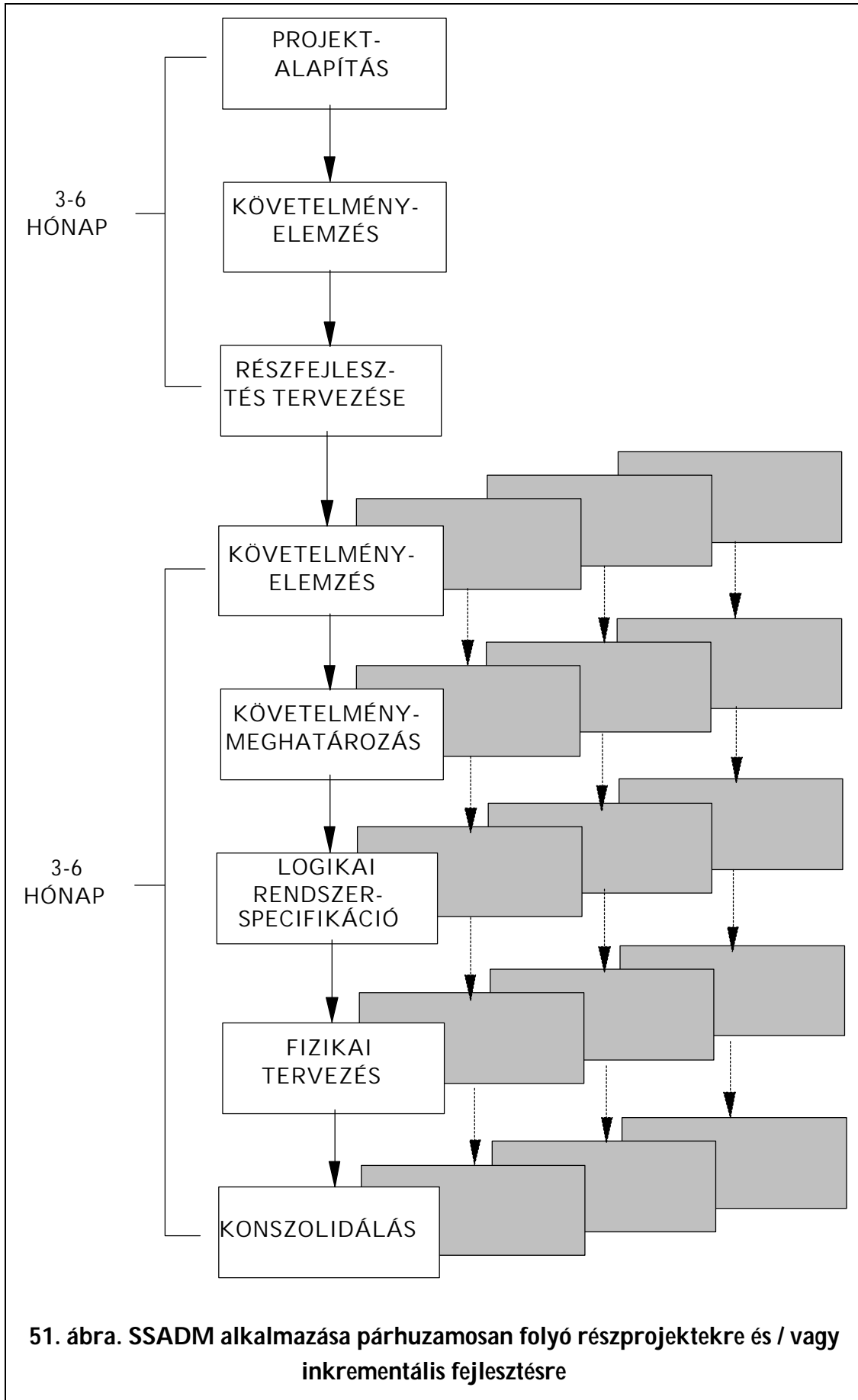
A szoftver fejlesztési környezet kiválasztása ideális esetben a rendszertervezési szakasz befejezése után történik meg, azaz miután a rendszert minden részletére kiterjedően már megtervezték.

13.10 SSADM alternatív életciklusok, testre szabási lehetőségek

Ebben a szakaszban illusztratív jelleggel néhány lehetséges testre szabási életciklust mutatunk be. Az időkorlátok miatt (Timeboxing)) gyakran alkalmazandó elv az, hogy a tervezési objektumok csak 10-20%-ra⁷⁹ kell a teljes mélységű elemzést elvégezni a megfelelő minőség biztosítása érdekében.



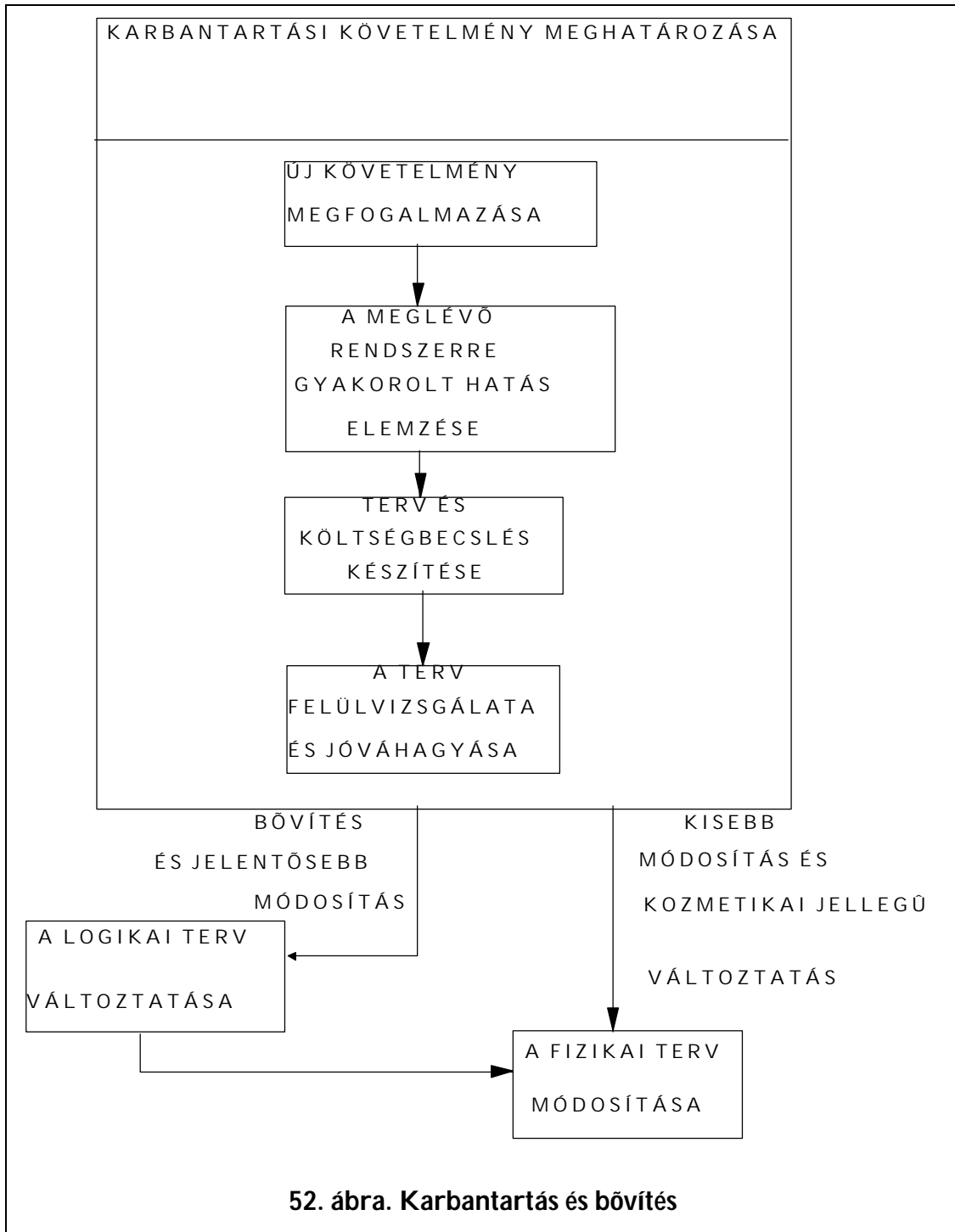
⁷⁹ Ez az elv egy általános rendszerelméleti elvből következik a (a gyakorlati tapasztalatok mellett), amit Pareto-elvnek hívnak és azt mondja, hogy egy rendszer működésének 80%-ért a rendszer funkcióinak 20%-a felel.



51. ábra. SSADM alkalmazása párhuzamosan folyó részprojektekre és / vagy inkrementális fejlesztésre

A karbantartás és üzemeltetés nem tartozik az SSADM módszertan hatálya alá, de a karbantartást igénylő feladatokhoz is illeszthető az SSADM. Három jellegzetes ok vezethet karbantartási igények felléptéhez:

- **Korrekción karbantartás.** Ez a típusú karbantartási igény akkor lép fel, amikor a az üzemelő rendszer nem felel meg a felhasználók által támasztott követelményeknek. Ennek sok oka lehet, de ide tartozhat a kezdeti vizsgálati, helyzetfelmérési szakaszokban tévesen felmért felhasználói követelmények által okozott hibák, vagy gyenge rendszer tervezés, esetleg rossz megvalósítás eredménye lehet ez.



52. ábra. Karbantartás és bővítés

- Tökéletesítő karbantartás. Ez a karbantartási típus akkor jelenik meg, amikor az új információrendszer adaptációt a felhasználók, megismerik, kiismerik, és tapasztalatok alapján meg tudják mondani a, hogy felhasználói szempontból, hol vannak hatékonytalan, esetleg hatástalan, eredménytelen megoldások. Ezt a fajta karbantartást úgy lehet tekinteni, mint amely a rendszer szolgáltatásait, funkcionalitását kívánja növelni anélkül, hogy az alapfeladatokon változtatna. Ilyen jellegű javítások lehetnek, az állományok adat elérésének gyorsítása (állomány kezelő rendszer vagy adatbázis kezelő rendszer cseréjével, esetleg az algoritmus javításával). Esetleg bizonyos numerikus számítások algoritmusának a felgyorsítása.
- Adaptív karbantartás (alkalmazkodó jellegű). A korrekciós és a tökéletesítő karbantartás általában rövid távra vonatkozik. Azonban hosszabb idő alatt a felhasználók által támasztott igények is változnak, a szervezetet körülvevő világ is változik. Ezeknek az összessége oda vezethet, hogy a szervezet információ igénye változik meg. Ez a karbantartás a rendszerek *evolúciós* továbbfejlesztését jelenti, vagyis a rendszer feladatainak, szolgáltatásainak és funkcionalitásának a megváltoztatását jelenti az átalakuló igények, felhasználói követelmények kielégítésére. Ez a karbantartás az evolúciós prototípus / rendszer készítéssel rokon, a különbség csak annyi, hogy a karbantartás esetén az egész folyamat időtávja hosszabb, és valójában beilleszkedik a szervezet ciklikus működésébe, kisebb karbantartási projektekre feladatokra darabolva, míg az evolúciós fejlesztés esetében egy adott projekten belül történik meg az iteráció sokkal rövidebb idő kereteken belül.

13.11 Projekt-változatok

Egy sor olyan projekt-típus van, amely hatással van a rendszertechnikai alternatívák kialakítására. Ezek befolyásolhatják az SSADM termékek szükségességét és részletességét is. A fő típusok a következők:

- Csomagválasztás
- Testre szabás
- Kulcsrakész rendszer
- Szolgáltatás

A 3. szakasz végére előállt a felhasználói követelmények teljes leírása a követelmény-specifikáció formájában. Ez biztos alapot ad a projekt további menetére vonatkozó döntésekhez.

13.11.1 Csomagválasztás

Ez egy olyan megvalósítási forma, ahol a követelményeket alkalmazáscsomagok beszerzésével elégítik ki. Itt a cél az, hogy a csomag által nyújtott lehetőségeket az igényelt funkcionalitásnak feleltessék meg. A rendszertechnikai alternatívák a funkciókra, ezek bemeneteire és kimeneteire és a hozzájuk tartozó mennyiségi adatokra fognak

koncentrálni. Ez a megközelítés alapos piacfelmerést, kipróbálást, vizsgálatot és tesztelést igényel. A csomagok és a követelmények illeszkedési foka nagyon fontos, a bemutatók során ezt kell kiemelni.

13.11.2 Testre szabás

A testre szabás jellegű fejlesztés azt jelenti, hogy a projekt munkacsoport átadja a fizikai rendszertervet egy házon belüli megvalósító csoportnak. Ha az igényelt konfigurációnak meg kell egyeznie a jelenlegivel, akkor a rendszertехnikai alternatívák kialakítása főleg kapacitástervezést igényel.

13.11.3 Szolgáltatás

Ez informatikai szolgáltatások beszerzését jelenti, a meghatározása: "megállapodás a szerződő féllel, amely lefedi a számítógépes és/vagy kommunikációs üzemeltetés nyújtására, illetve kapcsolódó erőforrásokra és helyszínekre vonatkozó irányítási és technikai felelősséget". Itt a számítógépes szolgáltatást egy harmadik fél nyújtja. Ilyenkor a rendszertехnikai alternatívának a rendszer határait, a határokat átlépő tranzakciókra és az igényelt szolgáltatási szintekre kell koncentrálnia. Ki kell alakítani az igényelt szolgáltatási szintekre vonatkozó szerződéses megállapodást.

13.11.4 Kulcsrakész rendszer

A kulcsrakész megoldás beszerzése a következőket jelenti: "egy teljes rendszer, amelyet felhasználók meghatározott csoportja részére terveztek. A szállító teljes felelősséggel tartozik a szoftver, hardver és dokumentáció tervezéséért és üzembehelyezéséért. A szállító gyakran az architektúráért is felel. A rendszer működésre kész, amint leszállították."

Itt felkérésre a potenciális szállítók egy technikai tervezési tanulmányt készítenek, amellyel bizonyítják rátermettségüket a követelményeknek megfelelő rendszer szállítására. Ezek a tanulmányok egy tender alapját képezik, amely a továbbiakban szerződéskötésben folytatódik. A szerződést elnyerő szállító ezek után elvégzi a rendszertехnikai alternatívák kialakítását, amit a projekt munkacsoport felülvizsgálhat.

14. Bibliográfia

14.1 Idegen nyelvű

- [Allen78] Allen, A. O., *Probability, statistics, and queueing theory*, Academic Press, London, 1978. ISBN 0-12-051050-2
- [Ashworth93] Asworth, C., Slater, L., *An Introduction to SSADM Version 4*, McGraw-Hill Book Company, London, 1993. ISBN 0-07-707725-3
- [Bertalanffy68] Von Bertalanffy L., *General System Theory: Foundations, Development, Applications*, Penguin, London, 1968.
- [Booch91] Booch, G., *Object-Oriented Design*, Benjamin/Cummings, Redwood City, Calif., 1991.
- [Burgess90] Burgess, R. S., *Structured Program Design: using JSP*, IEEE Stanley Thorners (Publishers) Ltd., Cheltenham, 1990. ISBN 0 7487 0360 8
- [Cameron83] Cameron, J.R., *JSP and JSD: The Jackson Approach to Software Development*, IEEE Computer. Soc., 1983.
- [CCTA90] CCTA (Central Computer and Telecommunication Agency), *SSADM Version 4 Reference Manuals*, Vols 1,2,3,4 NCC Blackwell, Manchester, Oxford, 1990.
- [CCTA90A] CCTA, *SSADM Version 3 and Capacity Planning*, Information Systems Engineering Division, CCTA, Norwich, 1990.
- [CCTA91] CCTA (Central Computer and Telecommunication Agency), *PRINCE , Structured Project Management*, NCC Blackwell Ltd., Manchester, Oxford, 1991.
- [CCTA93] CCTA, *Prototyping in an SSADM Environment*, ISE Library, HMSO, 1993.
- [CCTA94] CCTA, *Customising SSADM*, ISE Library, HMSO, 1994.
- [CCTA95] CCTA (Central Computer and Telecommunication Agency), *SSADM Version 4+ Reference Manual*, Vols 1,2,3 NCC Blackwell, Manchester, Oxford, 1995.
- [CCTA95A] CCTA (Central Computer and Telecommunication Agency), *SSADM Version 4+ Users Guide*, NCC Blackwell, Manchester, Oxford, 1995.
- [CCTA95B] CCTA (Central Computer and Telecommunication Agency), *Euromethod in Practice*, NCC Blackwell, Manchester, Oxford, 1995.
- [CCTA96] CCTA (Central Computer and Telecommunication Agency), *SSADM Version 4+, Version 4.3*, London, HMSO, The Stationery Office, 1996.
- [Checkland81] Checkland, P., *Systems Thinking, Systems Practice*, John Wiley, Chichester, 1981.
- [Checkland90] Checkland, P., Scholes, J., *Soft Systems: Methodology in Action*, John Wiley, Chichester, 1990.

- [Chen76] Chen, P. P., 'The Entity-Relationship Model: Towards a Unified View of Data', *ACM Transactions on Database Systems*, Vol. 1, No. 1, March 1976, pp 9-36, 1976.
- [Chen81] Chen, P. P.-S., 'A Preliminary Framework for Entity-Relationship Models', In P. P.-S. Chen (ed.), *Entity-Relationship Approach to Information Modeling and Analysis*, ER Institute, PO Box 617, Saugus, Calif. 91350, 1981.
- [Churchman68] Churchman, C., W., *The Systems Approach*, Dell Publishing Co., New York, 1968.
- [Coad91] Coad, P., Yourdon, E., *Object-Oriented Analysis*, Second Edition, Yourdon Press, Prentice Hall, Englewood Cliffs, New Jersey, 1991. ISBN 0-13-629981-4
- [Date90] Date, C. J., *An Introduction to Database Systems*, Addison-Wesley, 1990.
- [David92] David, A., *SSADM and Capacity Planning, Information Systems Engineering Library*, CCTA, London: HMSO, 1992. ISBN 0 11 330577 X
- [Downs92] Downs, E., Clare, P., Coe, I., *Structured Systems Analysis and Design Method, Application and Context*, Second Edition, Prentice Hall International (UK) Ltd., New York, London, 1992.
- [DSDM95] *Dynamic Systems Development Method*, The DSDM Consortium, (November) 1995. (Internet:info@dsdm.org; WWW Home page <http://www.dsdm.org>)
- [Eva92] Eva, M., *SSADM Version 4: A user's guide*, McGraw-Hill, 1992.
- [Gane79] Gane, C., Sarson, T., *Structured Systems Analysis: Tools and Techniques*, Prentice-Hall, NJ, 1979.
- [Gane90] Gane, C., *Computer Aided Software Engineering, the methodologies, the products and the future*, Prentice-Hall, 1990.
- [Griethuysen82] van Griethuysen (ed.), *Concepts and terminology for the conceptual schema and the information base, computers and information processing*, ISO/TC97/SC5/WG3 International Organisation for Standardisation, Geneva, Switzerland, 1982.
- [Hares90] Hares, J. S., *SSADM for the Advanced Practitioner*, John Wiley & Sons, Chichester, England, 1990.
- [Hargrave96] Hargrave, D., *SSADM4+ for Rapid System Development*, McGraw-Hill, London, 1996
- [Hewett89] Hewett, J., Durham, T., *CASE: The next step*, Ovum Ltd., 1989.
- [Hußmann94] Hußmann, H., *Formal Foundations for SSADM, An approach Integrating the Formal and Pragmatic Worlds of Requirements Engineering*, FAST-Bericht Nr. 94-02, Forschungsinstitut für Angewandte Software-Technologie e. V. (Arabellastr. 17, D-8195 München, Germany, e-mail:info@fast.de), Juni 1994.
- [ISE92] *Applying Soft Systems Methodology to an SSADM Feasibility Study*, ISE (Information Systems Engineering) Library, HMSO (Her Majesty Stationary Office), 1992.
- [ISE94] *Distributed Systems: Application Development*, ed. R.Duschl, Dr. J. Stewart, authors: Dr. M. Breu, A. Aue, J. Hall, K. Robinson, ISE (Information Systems

- Engineering) Library, HMSO (Her Majesty Stationary Office) / Siemens Nixdorf, 1994.
- [ISE94A] *SSADM and Client Server Applications*, ISE Library, HMSO, 1994.
- [Jackson82] Jackson, M., A., *System Development*, Prentice Hall, Englewood Cliffs, 1982. ISBN 0-13-880328-5
- [Johnson91] Johnson, A., *Capacity Planning, IT Infrastructure Library*, CCTA, London: HMSO, 1991. ISBN 0 11 330544 3
- [Jones86] Jones, C., B., *Systematic Software Development using VDM*, Prentice Hall International, London, 1986.
- [Kant92] Kant, K., *Introduction to Computer System Performance Evaluation*, McGraw-Hill, New York, 1992. ISBN 0-07-112668-6
- [Layzell89] Layzell, P., Loucopoulos, P., *Systems Analysis and Development*, Chartwell-Bratt, Studentlitteratur, 3rd edition, 1989. ISBN 0-86238-215-7.
- [Lazowska84] Lazowska, E. D., Zahorjan, J., Graham, G. S., Sevcik, K. C., *Quantitative System Performance: Computer System Analysis Using Queuing Network Models*, Prentice-Hall International, Inc., Englewood Cliffs, NJ., 1984. ISBN 0-13-746975-6
- [Longworth86] Longworth, G., Nichols, D., *SSADM Manual Vol. 1-2*, NCC Blackwell, 1986.
- [Longworth88] Longworth, G., Nichols, D., Abbot, J., *SSADM Developer's Handbook*, NCC Publications, Blackwell, Manchester, 1988.
- [Martin81] Martin, J., *Design and strategy for distributed data processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981. ISBN 0-13-201657-5
- [Martin88] Martin, J., McClure, C., *Structured Techniques, The Base for CASE*, Prentice Hall, Englewood Cliffs, New Jersey, ISBN 0-13-854936-2.
- [Martin89] Martin, J., *Information Engineering: a trilogy*, Vol. 1, Introduction, Prentice Hall, Englewood Cliffs, New Jersey, 1989. ISBN 0-13-464462-X
- [MartinFin81] Martin, J., Finkelstein, C., *Information Engineering*, Vols. 1. and 2., Prentice Hall, Englewood Cliffs, New Jersey, 1981.
- [Matheron90] Matheron, J. P., *Comprendre Merise, Outils Conceptuels et Organisationnels*, Editions EYROLLES, 1990.
- [Menascé94] Menascé, D. A., Almeida, V. A. F., Dowdy, L. W., *Capacity Planning and Performance Modelling, From Mainframes to Client-Server Systems*, Prentice Hall International, Inc., Englewood Cliffs, NJ., 1994. ISBN 0-13-035494-5
- [Meyer88] Meyer, B., *Object-Oriented Software Construction*, Prentice Hall, Hertfordshire, England, 1988.
- [Molnár95] Molnár, B., *A Methodology for Designing Responsive Information Systems, Integrating the Pragmatic and Theoretical Approaches within SSADM environment*, PhD thesis, Department of Mathematics and Computer Sciences, the Faculty of Electrical Engineering and Informatics, Technical University of Budapest, 1995.

- [Olle91] Olle, T. W., Hagelstein, J., Macdonald, I. G., Rolland, C., Sol, H. G., Van Assche, F. J. M., Verrijn-Stuart, A. A., *Information Systems Methodologies: A framework for understanding*, 2nd. edition, Addison-Wesley, Wokingham, England, 1991.
- [Pham91] Pham Thu Quang, Chartier-Kastler, C., *MERISE in Practice*, Macmillan Education Ltd, Houndmills, 1991.
- [Polack94] Polack, F., Whiston, M., Mander, K. C., *The SAZ method*, version 1.1, University of York, January 1994.
- [Rochfeld83] Rochfeld, A., Tardieu, H., 'Merise: *An information system design and development methodology*', *Information Management*, Vol. 6, No. 3., pp. 143-159, 1983.
- [Rumbaugh91] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., *Object-Oriented Modeling and Design*, Prentice Hall, Englewood Cliffs, New Jersey, 1991. ISBN 0-13-630054-5.
- [Shlaer88] Shlaer, S., Mellor, S. J., *Object-Oriented Systems Analysis: Modeling the World in Data*, Yourdon Press, Englewood Cliffs, New Jersey, 1988.
- [Skidmore92] Skidmore, S., Farmer, R., Mills, G., *SSADM Models and Methods Version 4*, NCC Blackwell, Manchester, Oxford, 1992. ISBN 0-85012-796-3
- [Spivey88] J. M. Spivey, *Understanding Z: A specification Language and its Formal Semantics*, Number 3 in Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, Cambridge, 1988.
- [Spivey89] J. M. Spivey, *The Z Notation: A Reference Manual*, Prentice Hall, Hemel Hempstead, 1989.
- [Tsichritzis78] Tsichritzis, D., C., Klug A.. *The ANSI//X3/SPARC DBMS Framework: Report of the Study Group on Data Base Management Systems*, American National Standard for Information Systems, X3, 1978
- [Turner90] Turner, W. S., Langenhorst, R. P., Hice, G. F., Eilers, H. B., Uijtenbroek, A. A., *SDM system development methodology*, Elsevier Science Publishers B.V. (North-Holland)/Pandata, 1990.
- [Turner96] Turner, P., Jenkins, T., *Euromethod and Beyond, Open Frameworks for European Information Systems*, International Thomson Computer Press, London, 1996.
- [Ungoed94] Ungoed, A., *A model of SSADM concepts for non-SSADMers*, Thesis, Brighton University, U. K., 1994.
- [Ward85] Ward, P. T., Mellor, S. J., *Structured Development for real-time systems*, Volumes I-III. New York, Yourdon Press, 1985-86.
- [Warnier81] Warnier, J. D., *Logical Construction of Systems*, Van Nostrand Reinhold, New York, 1981.
- [Yourdon75] Yourdon, E., Constantine, L., *Structured Design*, Yourdon Inc., New York, 1975.
- [Yourdon89] Yourdon, E., *Modern Structured Analysis*, Prentice-Hall International, Inc., Englewood Cliffs, NJ, 1989.

14.2 Magyar nyelvű

- [Bana94] Bana István, *Az SSADM rendszerszerkezési módszertan*, LSI, Számalk, Budapest, 1994.
- [BKE] Budapesti Közgazdaságtudományi Egyetem jegyzet, *Rendszerfejlesztés módszertana*, SSADM, összeállította: Molnár Bálint, adjunktus
- [Halassy94] Halassy Béla, *Az adatbázis tervezés alapjai és titkai*, IDG kft., Budapest, 1994.
- [Kincses93] Kincses L., (szerk.), *SSADM, Strukturált rendszerelemzési és -tervezési módszer*, MTA Információtechnológiai Alapítvány, 1993. (Isd. még <http://www.itb.hu> WWW lapon az Informatikai Tárcaközi Bizottság ajánlásai között)
- [Kovács95] Dr. Kovácsné Cohner Judit, Takács Tibor, *Ismerkedés az SSADM-mel*, Computer Books, Budapest, 1995.
- [Euromethod94] *Euromethod áttekintés*,
Euromethod vevői útmutató,
Euromethod szállítói útmutató,
Euromethod kivitelezés tervezési útmutató,
Euromethod esettanulmány, 1994--- Miniszterelnöki Hivatal Informatikai Koordinációs Iroda, Hirlapkiadó. (Isd. még <http://www.itb.hu> WWW lapon az Informatikai Tárcaközi Bizottság ajánlásai között)
- [Quittner93] Quittner Pál, *Adatbáziskezelés a gyakorlatban*, Akadémiai Kiadó, Budapest, 1993. ISBN 963 05 6587 0.
- [Vecsenyi88] Vecsenyi János, *Szervezeti problémamegoldás*, Marx Károly Közgazdaságtudományi Egyetem, Közgazdasági Továbbképző Intézet, Budapest, 1988.