

Diplomamunka

Ferenc Rudolf

5. éves programtervező matematikus hallgató

Adatfolyam-diagramok globális elemzése. Információs rendszer tervezése SSADM-mel.

Témavezető:

Dr. Gyimóthy Tibor

MTA-JATE Mesterséges Intelligencia Kutatócsoport

Szeged, 1997

Tartalomjegyzék

1. BEVEZETÉS.....	4
2. AZ SSADM-RŐL ÁLTALÁBAN	6
2.1. ÁTTEKINTÉS A MÓDSZERTANRÓL	6
2.1.1. A módszertan kialakulásának előzményei.....	6
2.2. AZ SSADM HASZNÁLATÁNAK ELŐNYEI	7
2.2.1. A rendszer elkészülésének időpontja jól tervezhető	7
2.2.2. A rendszer a felhasználó igényei szerint készül	7
2.2.3. A működési környezet változásait követő rendszer készül	8
2.2.4. A meglévő szakértői ismeretek hatékony és gazdaságos felhasználása	8
2.2.5. A hibázás kockázatának csökkentése.....	8
2.2.6. A rugalmasság növelése.....	8
2.2.7. A termelékenység növelése.....	8
2.3. AZ INFORMÁCIÓS RENDSZEREK ÉLETCEKLUSA ÉS AZ SSADM.....	9
2.4. SSADM-ET HASZNÁLÓ PROJEKT INDÍTÁSÁNAK ALAPFELTÉTELEI	10
2.4.1. Információ oldali problémák	11
2.4.2. Az eljárások problémái	11
2.4.3. Terjedelmi problémák.....	11
2.5. A MÓDSZER SZERKEZETE.....	11
2.6. A MÓDSZER ALAPELVEI	13
2.6.1. Az SSADM célja	13
2.6.2. A projekt résztvevői és nézőpontjaik.....	14
2.6.3. Követelmény-központúság	15
2.7. ÁLTALÁNOS ELEMZÉSI MÓDSZEREK	15
2.7.1. Dokumentum-elemzés.....	15
2.7.2. Interjúk	15
2.7.3. Kérdőívek.....	16
2.8. SSADM ELEMZÉSI TECHNIKÁK.....	16
2.8.1. Dokumentum-áramlási diagram.....	16
2.8.2. Fizikai adatfolyam-modell	16
2.8.3. Logikai adatfolyam-modell	17
2.9. A MODELLEK ELEMEI	17
2.9.1. Folyamatok	17
2.9.2. Adattárak	18
2.9.3. Adatfolyamok.....	19
3. AZ ADATFOLYAM-MODELL (DFM) GLOBÁLIS ELEMZÉSE	20
3.1. JELÖLÉSEK ÉS DEFINÍCIÓK	21
3.2. ADATFOLYAM FÜGGŐSÉGI GRÁF (DFDG)	22
3.3. KONZISZTENCIA VIZSGÁLATOK	32
3.4. KONZISZTENS MŰVELETEK A GLOBÁLIS DFDG-N.....	35
3.5. A KONSTRUKCIÓBAN REJLŐ TOVÁBBI LEHETŐSÉGEK	37

4. FELHASZNÁLÓI IGÉNYEK ELEMZÉSE	38
4.1. A FELADAT ISMERTETÉSE	38
4.2. AZ IGÉNYELT INFORMÁCIÓS RENDSZER FŐBB RÉSZEI	40
4.2.1. Projektnyilvántartás.....	40
4.2.2. Egységes partneryilvántartás.....	40
4.2.3. Számlanyilvántartás.....	41
4.2.4. Kimenő és bejövő információk nyilvántartása és projekthez kapcsolása.....	41
4.3. TOVÁBBI MEGVALÓSÍTANDÓ FELADATOK	42
4.3.1. Számítógépes konferenciázó rendszer	42
4.3.2. Az átállás folyamatával kapcsolatos igények	42
5. SSADM TERMÉKEK	43
5.1. DOKUMENTUM ÁRAMLÁSI DIAGRAM.....	43
5.2. KONTEXTUS ÁBRA (NULLADIK SZINTŰ ADATFOLYAM DIAGRAM).....	44
5.3. FIZIKAI ADATFOLYAM MODELL	45
5.4. LOGIKAI ADATFOLYAM MODELL	55
6. IRODALOM	68

1. Bevezetés

A számítástechnika és informatika területén tapasztalható gyors fejlődés ma óriási lehetőségeket biztosít a rendszerfejlesztők, programozók és ezen keresztül a felhasználók számára. A számítástechnikai erőforrásoknak a végfelhasználói igényekhez való igazítása minden szervezetnél nagy problémát jelent. A felhasználók rendelkezésére álló adatfeldolgozási támogatás kulcskérdése a hatékony, gyors programfejlesztés, a növekvő napi feladatok megoldása, valamint az aktuális szervezeti, működési, számítógépes rendszerek és adatállományok naprakész dokumentálása.

Az SSADM az angol „Structured Systems Analysis and Design Method”, azaz a „Strukturált Rendszerelemzési- és Tervezési Módszer” rövidítése, amely a fent említett problémakörre hatékony megoldást kínál. A brit kormányzatban ún. kormányzati szabványként alkalmazzák az információs rendszerek fejlesztésében. A módszer elkülönült egységekre osztja fel az információs rendszer fejlesztésének munkáit és hajlékonyan idomul a különböző feladatokhoz. Az SSADM olyan elterjedt technikákat használ, mint például az egyed modellezés, adatfolyam diagramok, Jackson jelöléstechnikát és elveket alkalmazó (Jackson jellegű) ábrák. Az ilyen technikákat használó fejlesztők könnyen beilleszkedhetnek az SSADM környezetbe.

Egy cég felkérésére olyan információs rendszer alapjait kellett lefektetnünk, mely igazodik jelenlegi és jövőbeni tevékenységeihez. A vállalat tevékenységei alapvetően építkezési projektek köré épülnek, ezért az információs rendszernek maximálisan támogatnia kell ezt, maradéktalanul ki kell elégítenie a cég által elnyert ISO 9001-es szabványt, és minden alrendszerét tekintve funkcionálisan integrált egységet kell alkotnia.

Az információs rendszerek elemzése és tervezése során a DFD-k meghatározó jelentőségűek a rendszer üzemszerű működésére nézve. Ezen termékek nagyfokú precizitást, hibamentességet követelnek meg, tehát szigorú minőségi követelményeknek kell megfelelniük. Egy fejlesztési munka során ez azt jelenti, hogy ezek olyan részmunkák, amelyek jelentős idő és munkaerő többletet igényelnek. Az ilyen munkák automatizálása, de legalább egyes mozzanatainak számítógépes támogatása jelentős szerepet játszhat az erőforrások megtakarítása, a munkafolyamat felgyorsítása terén, így nagyban hozzájárulhat, hogy jól használható, hibáktól mentes adatszolgáltatási rendszer készüljön. Ahhoz, hogy egyes tervezési fázisok számítógépes segítséget is felhasználhassanak, modellezni kell a megvalósításra kerülő technikát.

A munkát eredetileg hárman kezdtük el, mely dolgozat továbbjutott az Országos Tudományos Diákköri Konferenciára. Ez után a munkát ketten folytattuk, jelen dolgozat az általam megalkotott DFDG-ben (**DFDG – Dataflow Dependency Graph – Adatfolyam Függőségi Gráf**) mélyül el.

Ez a modell lehetővé teszi az adatfolyam diagramok globális elemzését, vizsgálatát és tesztelését matematikai módszerekkel. A matematikai modellt alapul véve számítógéppel megvalósítható egy sor olyan algoritmus, amely manuálisan csak fáradságos munkával, összehasonlíthatatlanul nagy hibaelkövetési eséllyel, és nagy időráfordítással készíthető csak el.

A felépített modellre megadhatóak olyan algoritmusok, melyekkel meghatározhatjuk, hogy egy adattárba mely külső egyedek és folyamatok szolgáltatnak adatokat, illetve egy külső egyed vagy folyamat, mely adattárakból kap információkat. Fontosak még a **slicing-eljárások**, melyeknek elsődleges szerepe a hibakereséseknél (debugging) nyilvánul meg. Hiba esetén ezekkel viszonylag könnyen meghatározható a hiba keletkezésének helye, illetve az mely csomópontokra van kihatással. A **független komponensek** lehetővé teszik a párhuzamos munkát, azaz egy időben több szálon is folyhat a rendszer fejlesztése. Másik előnye, hogy a kész rendszer adatfeldolgozását többprocesszoros környezetre optimalizálhatjuk.

Az általunk választott SSADM módszertant a 2. fejezetben mutatom be.

A 3. fejezetben definiálom az Adatfolyam Függőségi Gráfot, majd megadok olyan algoritmusokat és tételeket, amelyek kapcsolatot teremtenek az Adatfolyam Diagramok és a DFDG között.

A további fejezetekben egy konkrét vállalati információs rendszer elemzésének és tervezésének lépéseit mutatom be, külön hangsúlyt fektetve az Adatfolyam modellezésre.

2. Az SSADM-ről általában

2.1. Áttekintés a módszertanról

Az SSADM (Structured Systems Analysis and Design Method) magyarra Strukturált Rendszerelemzési és Tervezési Módszer-ként fordítható. Az Egyesült Királyságban kormányzati szabványként alkalmazzák információs rendszerek fejlesztésénél. Elkülönült modulokra osztja fel egy információs rendszer fejlesztését és megfelelően alkalmazkodik a megvalósítandó feladatokhoz.

2.1.1. A módszertan kialakulásának előzményei

Az SSADM információs rendszereken alapuló alkalmazások elemzésére és tervezésére szolgál. A módszer első változata az LBMS cég nevéhez fűződik. Ők készítették el, arra a pályázatra, amelyet 1980-ban a Központi Számítástechnikai és Távközlési Ügynökség (angol rövidítéssel CCTA) írt ki. A pályázatban megfogalmazott követelmények a következő igényeket támasztották a kidolgozandó módszertannal szemben:

- az ellenőrzés lehetősége legyen benne
- kipróbált módszereket alkalmazzon
- legyen alakítható
- legyen tanítható

1981-ben elfogadták az LBMS javaslatát és nem sokkal később valós projekteken kezdték alkalmazni. 1983 januárjától megkövetelték a fejlesztő csoportoktól a használatát a brit kormányzati projekteken.

1987 őszén az SSADM-et a CCTA által alapított Fejlesztés Felügyeleti Testület (Design Authority Board) felügyelete alá helyezték. A DAB a CCTA-tól függetlenül működik és a módszer fejlesztési ügyeivel foglalkozik.

1982-ben megalakítottak egy kormányzati felhasználói csoportot.

1988-ban a CCTA támogatásával megalakult egy nyilvános felhasználói csoport is (SSADM User's group), amelynek képviselője van a Fejlesztés Felügyeleti Testületben.

1988-ban a Brit Számítástechnikai Társaság keretén belül működő Információs Rendszerek Vizsgabizottsága (IS Examination Board, ISEB) egy ellenőrzési rendszert

hozott létre SSADM-et oktató tanfolyamok minősítésére. A hivatalosan minősített tanfolyamok résztvevői meghatározott vizsgák sikeres letétele után megkaphatják az SSADM szakértői igazolást. 1991-óta azon SSADM-et használó projektek résztvevői számára, amelyek a brit kormányzati szervek részére készülnek, előírás követeli meg a szakértői igazolást.

A módszertan sikerét mutatja, hogy 1991-ben a CCTA által kiadott SSADM Szolgáltatások Jegyzékében, megtalálható 139 tanácsadó cég, 28 engedélyezett tanfolyamot nyújtó cég, 30 CASE eszköz gyártó és 35 olyan negyedik generációs eszközöket gyártó cég, amely SSADM-hez kapcsolódó útmutatóval rendelkezik.

2.2. Az SSADM használatának előnyei

Különböző környezetekben, más-más feladatot megvalósító információs rendszerek fejlesztésénél legtöbbször azonos problémákkal találjuk szemben magunkat. A módszer előnyei ezen problémák kiküszöbölése által mutatják meg magukat.

2.2.1. A rendszer elkészülésének időpontja jól tervezhető

A szerződésbe foglalt határidők betartása legtöbbször a megfelelő tervektől, a megfelelő vezetési és ellenőrzési rendszerektől függ.

Az SSADM szerkezetéből, hierarchikus felépítéséből és termékközpontúságából adódóan megengedi, hogy a feladatokat elemi szintekig lebontsuk, ami által tudjuk: mit kell előállítani, mikor és hogyan. Meghatározott helyeken hangsúlyt fektet a projekt menetének ellenőrzésére, megfelelő részletességgel kitérve a felügyelet paramétereire. Mivel részletes termékleírásokat használ, többé kevésbé pontosan becsülhető az elvégzendő munka mennyisége.

2.2.2. A rendszer a felhasználó igényei szerint készül

A módszertan a követelmény központúságából adódóan, szükségessé és lehetővé teszi a felhasználók bevonását a fejlesztés menetébe. Az áttekinthető ábrák (grafikus technikák) használatával, a prototípus készítéssel, az alternatívák felvázolásával bármely projektben kommunikációs csatornák állnak rendelkezésünkre a felhasználó irányába.

2.2.3. A működési környezet változásait követő rendszer készül

A rendszer az SSADM előírásai alapján olyan dokumentációval egészül ki, amely tartalmazza a működési terület célkitűzéseit és a fejlesztők szándékait. A nem mindig azonos megközelítéssel bíró nézeteket ötvöző specifikáció tartalmazza azokat az alapvető információkat, amelyek nélkülözhetetlenek a rendszer karbantartásához és továbbfejlesztéséhez.

2.2.4. A meglévő szakértői ismeretek hatékony és gazdaságos felhasználása

Mivel az SSADM igen elterjedt technikákat használ, (pl. egyed modellezés, adatfolyam ábrák, Jackson jelöléstechnikát és elveket alkalmazó ábrák), ezért az ilyen technikákat ismerő és használó fejlesztők otthonosan mozoghatnak az SSADM környezetben is.

2.2.5. A hibázás kockázatának csökkentése

A felhasználók és a tapasztalt fejlesztők bevonásával a rendszer minősége növelhető a hibák korai felfedezésével. A különböző technikák eredményeinek összevetése és a többszemponútú megközelítés hatékonyan biztosítja a teljességet és az összeillőséget. A fejlesztés során elkészülő dokumentumok minőségi követelményeinek pontos meghatározásával, a tesztelés módjának leírásával az SSADM jobb minőségbiztosítást tesz lehetővé és megkönnyíti az ISO 9001 szabvány bevezetését.

2.2.6. A rugalmasság növelése

A projektirányításhoz tartozik meghatározni az elkészítésre kerülő termékeket. Az SSADM leírja a termékek elkészítésére vonatkozó összes tevékenységet. Amennyiben az irányítás elég szakmai tapasztalattal rendelkezik, az erőfeszítések a kritikus termékekre összpontosíthatók.

2.2.7. A termelékenység növelése

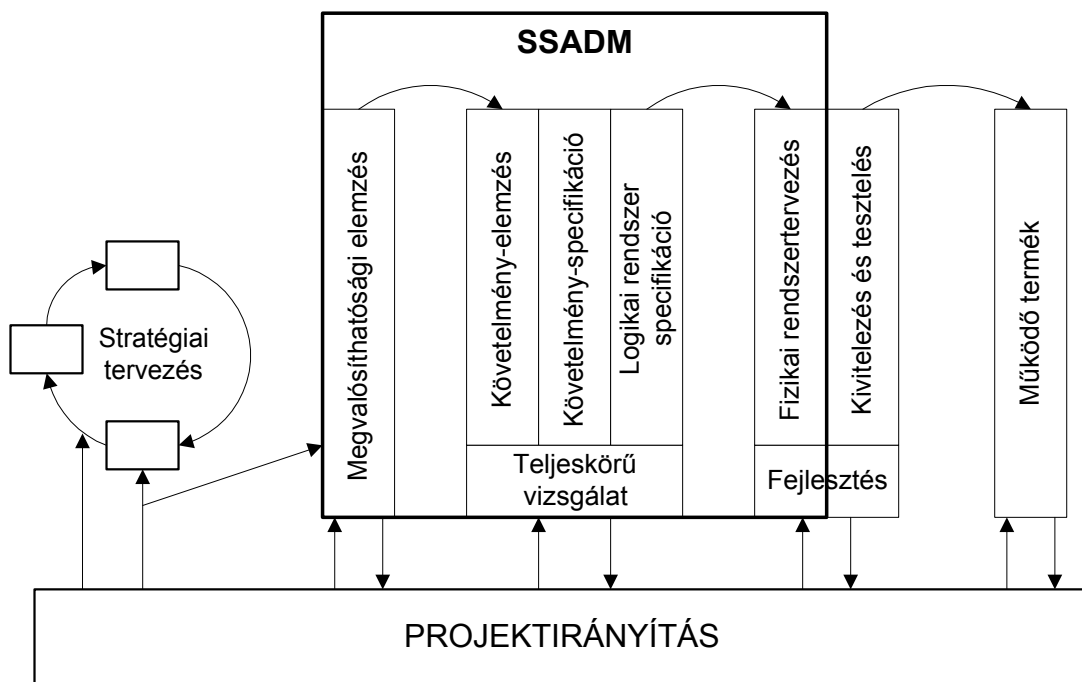
A módszer, jól dokumentált technikái révén, jól tanítható és érthető. Ezáltal nagyobb esélye van annak, hogy az első próbálkozás is sikeres legyen. A termék-központúság megszabadít a felesleges tevékenységek elvégzésétől, illetve a minden apróságra kiterjedő, túlzottan részletes dokumentáció készítésétől.

2.3. Az információs rendszerek életciklusa és az SSADM

Az információs rendszerek elemzésének és tervezésének feladataihoz az SSADM biztosít egy sor termék-meghatározást és az ezekhez kapcsolódó eljárásokat. Egy megfelelően megszervezett, vezetett és ellenőrzött projektben ezen dokumentációk formátuma megkönnyíti használatukat.

Egy vállalat információs rendszerének megtervezése, legyen az SSADM technikát használó vagy sem, az adott cég üzleti elvárásaitól, informatikai stratégiájától erősen befolyásolt. Egy SSADM projekt kezdeményezése előtt a cég ehhez fűződő dokumentumait megfelelő elemzésnek kell alávetni.

Azon projektek, amelyek alkalmazásokat állítanak elő, többnyire lineáris lefolyásúak, míg az informatikai stratégiai tervezés néhány éves ciklusban ismétli a feladat meghatározást, a kivitelezést és a felülvizsgálatot, ami sok projektet eredményezhet, köztük akár SSADM-et használókat is.



2.1. ábra Az SSADM helye az életciklusban

Egy információs rendszer fejlesztésének tipikus menete a következő:

- információs rendszerek stratégiai tanulmánya, melyben szerepelnie kell az adott információs rendszer projektjének is (többek között)
- megvalósíthatósági tanulmány
- teljes körű vizsgálat (a specifikáció létrehozására)

- fejlesztési projekt (a fizikai rendszerterv létrehozására és a rendszer felépítésére).

Stratégiai tervezésnél az SSADM nem használható, mivel ennek több, a vállalat munkájával kapcsolatos szakterületről származó feladatnak meg kell felelnie. Ez nem jelenti azt, hogy a technikai közül nem lehet néhány hasznos, például a szervezeti működés (üzleti/működési terület) modelljének az elkészítésénél (logikai adatmodellezés és adatfolyam-modellezés). Az SSADM technikáival nem lehet behatárolni a cég szervezeti erősségeit illetve gyengeségeit, az esetleges sikertényezőket vagy az üzleti lehetőségeket.

Ahol az SSADM-et már jól lehet használni az a megvalósíthatóság elemzés. Ebben a szakaszban segítségére lehet az elemző munkacsoportnak a javasolható alkalmazások meghatározásában és az informatika felhasználásában rejlő lehetőségek felderítésében. Azonban teljes képet nem ad az SSADM, mivel itt olyan kérdésekkel is foglalkozni kell, mint például a szervezeti és pénzügyi megvalósíthatóság, amelyeket az SSADM technikája hatékonyan támogat, de ezen kívül igényelnek a módszeren túlmutató egyéb technikákat és szaktudást is.

A megvalósíthatósági tanulmány hivatkozási alapként szolgál egy alkalmazást fejlesztő projekt számára. Függetlenül attól, hogy készült-e ilyen dokumentum vagy sem, az elemző munkacsoportnak szüksége lesz az ún. "projektalapító okirat"-ra, amely tartalmazza a projekt célkitűzéseit, kiterjedését és korlátait.

A teljes körű vizsgálat meghatározza a rendszer működési követelményeinek részleteit, amelyeket a következőképpen csoportosíthatunk:

- részletesen meghatározott funkcionális és adatokra vonatkozó követelmények, a minőség mérését lehetővé tevő objektív mértékekkel
- logikai rendszerterv, a működés eseményeit és a lekérdezési követelményeket kezelő műveletekkel, illetve a felhasználó kölcsönhatásokkal
- a technikai környezet leírása, a rendszert megvalósító hardver, szoftver és szervezeti elemek leírásával.

A fejlesztési tevékenység viszi tovább a projektet. Magába foglalja a kivitelezést és a tesztelést is, de ide tartoznak a felhasználók jóváhagyási illetve elfogadási eljárásai, valamint a hardver és szoftver beszerzés.

2.4. SSADM-et használó projekt indításának alapfeltételei

Informatikai projekt alapításakor a projektvezetőség az, amely testület meghatározza, hogy a célkitűzéseket milyen módon lehet a legkisebb költség- és idő

ráfördítással elérni. Az SSADM használata akkor jelent hatékony megoldást, ha a vezetőség a következőkben tárgyalt kérdésekre sorra igennel tud válaszolni.

2.4.1. Információ oldali problémák

- A rendszer által kezelendő információ elegendően strukturált a modellezéshez?
- Ábrázolható stabil, áttekinthető logikai adatszerkezet?

A rendszer által igényelt adatbázisok elkészítésekor felmerülhet az a probléma, hogy világosan megfogalmazható szerkezettel nem rendelkező szövegeket vagy túlzottan strukturált statisztikákat kell kezelni. Ezeket SSADM technikákkal nehézkes kezelni, modellezni. Megoldást jelenthet ilyenkor a problémát kiküszöbölő programcsomagokkal kiegészíteni az SSADM-et.

2.4.2. Az eljárások problémái

- A modellezéshez elegendően strukturáltak és megfelelő pontossággal rendelkeznek a javasolt rendszer által végzendő eljárások?
- Rajzolható felső szintű adatfolyam-ábra?

A hatékonyság érdekében meg kell határozni az eljárások jellegét, vagyis, hogy a rendszer egyes almoduljai általános célú informatikai támogatást igényelnek (pl. táblázatkezelés, elektronikus levelezés), vagy éppenséggel erőteljesen specializált eszközöket igényelnek (pl. számviteli függvények). Ebben az esetben az SSADM-et célszerű más technikákkal együtt használni, a kevésbé pontos funkciók kiszűrésére

2.4.3. Terjedelmi problémák

- Jól meghatározható az alkalmazás kiterjedése?
- Rajzolható kontextus ábra?

2.5. A módszer szerkezete

Az SSADM alapvetően két nagy részre bontható. Az egyikbe tartozik az SSADM törzsrésze (alapvető SSADM), míg a másikban vannak a hozzá kapcsolódó egyéb útmutatók.

Az SSADM termékei révén biztosítja az elemzőnek azon keretek felépítését, amellyel a rendszer működési területének követelményeit jól érthetően, világosan lehet dokumentálni. Ez később folyamatosan finomodik minél pontosabb az igények

részleteire vonatkozó tudás. Ami ezt segíti az az SSADM három nézőpontbeli megközelítése.

Ez a három nézőpont:

- funkciók (a felhasználók nézeteit tükrözik az eseményekre reagáló rendszer-feldolgozási folyamatokról)
- események (ezek lehetnek a működési terület valós eseményei, mint például „Számlák beérkezése”, vagy olyan rendszer által indított események, mint például egy napi kimutatás elkészítésének indítása)
- adatok (a rendszer adatokat kezel és tart karban annak érdekében, hogy nyújtani tudja a rendszer funkcionalitását)

Ez a megközelítés lehetségessé teszi még idejekorán a hibák kiszűrését a felhasználói követelmények részletes meghatározásakor.

A rendszerelemzéstől és rendszertervezéstől a projektirányításig, pénzügyi tervezésig és szervezeti irányításig terjedő tevékenységeket egy projekt-munkacsoportnak kell elvégeznie.

A kapacitástervezés, az adatbázisok és elosztott-rendszerek tervezése, a becslések és a termelékenység mérése mind más technikai szakértőket igényelnek. Az SSADM nem tartalmazza mindezeket az eljárásokat ugyanolyan részletesen mint a konkrét fejlesztői tevékenységeket.

Az SSADM törzsrészébe azon technikák és eljárások tartoznak, ami termékeket és eljárásokat jelent a következőkhöz:

- Megvalósíthatóság
- Követelmény-elemzés
- Követelmény-specifikáció
- Logikai rendszerspecifikáció
- Fizikai rendszertervezés

Az így leírt módszert kiegészítik ún. kapcsolódó útmutatók, amelyek egy sor vezetési és technikai kérdést fednek le.

2.6. A módszer alapelvei

2.6.1. Az SSADM célja

A módszer célja az, hogy segítsen a projekt tagjainak az informatikai stratégia részeként kitűzött információs rendszerre vonatkozó követelmények pontos elemzésében és a követelményeknek legjobban megfelelő információs rendszer megtervezésében.

A SSADM-mel végzett munka mindig egy világosan meghatározott projekt része, amelyet a következők jellemeznek:

- a projekten dolgozók dokumentum formájában megkapják munkájuk kiterjedésének leírását és az általuk elérendő üzleti/működési követelményeket (formális projekt-indítás)
- világosan megfogalmazott és beazonosítható céllal rendelkeznek, amely végső soron a fizikai rendszerspecifikáció előállítása (ennek nagyobb részét az SSADM fizikai rendszerspecifikációja alkotja)

A fizikai specifikáció két nagyobb részt foglal magába:

- az adattervet, melyet többnyire már egy konkrét adatbázis-kezelő rendszer fizikai adatbázisának fogalmaival kell leírni
- a feldolgozási tervet, amely támogatja azokat a feldolgozási folyamatokat, amelyeket a valós világ eseményeire válaszoló felhasználók határoznak meg.

Az SSADM moduláris felépítéséből adódóan nem csak átfogó elemzési-tervezési munkát enged meg, hanem könnyen alkalmazható közelebbi célokat kitűző projektekben is.

Példaként említhetők itt a következő részfejlesztések:

- ha a cél a megvalósítási lehetőségek felmérése, akkor önálló megvalósíthatósági elemzés készíthető
- ha a cél az aktuális helyzet felmérése és rendszerszervezési javaslatok kidolgozása, akkor önálló követelményelemzés készíthető
- egy információs rendszer megvalósításának technikai lehetőségeit és következményeit leíró követelményspecifikáció alapján technikai környezetre vonatkozó javaslatok kialakítása

2.6.2. A projekt résztvevői és nézőpontjaik

Egy projekten belül a sikeres véghezvitel felelőssége megoszlik a résztvevők között, akik a következő csoportokba sorolhatók:

- felhasználók (részvétel)
- vezetők (ellenőrzés)
- fejlesztők (használat)

A rendszer leírása előtt a hatékony munka érdekében meg kell állapítani minden egyes ilyen szerepkörnek a kitűzött céljait és prioritásait.

Felhasználók:

Az informatikai támogatást maximálisan a felhasználók igényei szerint kell megtervezni és megvalósítani, ezért az SSADM-ben magas prioritásúak. A felhasználók bevonása a módszer minden fázisában jól meghatározott és jól követhető. Minden szinten kifejezhetik elvárásaikat és üzleti/működési igényeiket a készülő rendszerrel szemben.

Mivel az SSADM grafikus termékei olyan ábrázolási módokkal készülnek, amelyek viszonylag könnyen érthetőek a felhasználók számára, létrejöhét egy kétirányú kommunikáció, mely a felhasználói igények világosabb megértéséhez vezethet. Ez pedig a felhasználói igények kielégítésének minél nagyobb fokát teszi lehetővé.

Vezetők:

A SSADM projekt ellenőrző, vezető szereplőinek a módszer által biztosított strukturáltság, termék-központú megközelítés nagy segítséget jelent. A moduláris felépítés, a munkafolyamat szakaszokra való bontása, a feladat hierarchikus szerkezete olyan támogatást jelentenek, amelyek az irányítást hathatósabbá teszik

A projekt bármely állapotában világosan látható:

- mik a célok
- milyen termékek készültek eddig el és milyenek fognak még elkészülni
- az adott időben milyen munkavégzés folyik
- milyen technikákat használnak fel az elkészítendő termékek előállítására

Fejlesztők:

A rendszerelemzők és tervezők munkáját szintén az SSADM termék-központú szerkezete támogatja. A munkafolyamat ütemeiben elkészítendő termékek jól

meghatározottak, az előállításukra irányuló technikák pontos leírása megtalálható a módszertanban.

A fejlesztők szempontjából fontos, hogy a technikák egy szigorú és átfogó rendszert alkotnak, hogy a termékek és technikák közötti kölcsönhatások leírása is elegendően részletes a módszer projektbeli megfelelő használatához.

2.6.3. Követelmény-központúság

A kritikus követelmények azonosítására az SSADM egy követelmény-meghatározás nevű technikát használ. Ezen technikával a munkacsoport figyelmét a működési terület felhasználóira és funkcióira irányítja, így pontosítva a projektindító anyagokat, melyek előző stratégiai illetve megvalósíthatósági tanulmányokból származnak.

A követelmény-elemzés során a cél létrehozni egy olyan központi dokumentumot, amelyet a projektirányítás és a fejlesztők a projekt befejezéséig végig használnak. Ez a követelményjegyzék.

A követelmény-specifikáció tehát több különböző részletes specifikációs termék együttese, amely a rendszer iránti igények teljes kifejezését adja.

2.7. Általános elemzési módszerek

2.7.1. Dokumentum-elemzés

Dokumentum-elemzésből állapítható meg, hogy mi az adott szervezet feladata, hogyan épül fel, és milyenek a függelmi kapcsolatok. A leggyakrabban használt dokumentum típusok:

- szervezeti ábra,
- munkaköri leírások

2.7.2. Interjúk

A dokumentumok tanulmányozása jó bevezető áttekintést ad, de mivel nem feltétlenül biztosít naprakész és élő információt szükségesek az interjúk is. A dokumentumokból megtudható, mi az adott szervezet feladata, hogyan tervezi céljait elérni, kik és hogyan kell hogy dolgozzanak, a munkaköri leírásokból pedig kiderülnek a felelősségi, hatásköri szabályok. A következő lépésben tehát meg kell

győződni arról, mennyire igazak a leírtak. Erre szolgál az interjú, mint a helyzetfelmérés egyik fontos eszköze.

2.7.3. Kérdőívek

Azon kérdésekre, melyek nem specifikusak a cég egyes részeire kérdőívekkel lehet választ kapni. Nem előre megadott válaszokat célszerű kérni, mivel ha nem fedik az összes lehetőséget, akkor a megkérdezettek véleményére sohasem derül fény.

2.8. SSADM elemzési technikák

A módszertan ajánlásait követve az alábbi lépéseknek kell megvalósulniuk:

- Egyfelől ki kell alakítani a jelenlegi környezet folyamatainak fizikai és logikai képét a dokumentum-áramlási és az adatfolyam-modellezési technika segítségével,
- Másfelől meg kell határozni a követelmények megvalósíthatóságát.

2.8.1. Dokumentum-áramlási diagram

A dokumentum áramlási diagramon az követhető nyomon, hogy milyen dokumentumok mozognak az információ feldolgozási folyamat résztvevői között. A dokumentum áramlási ábrán a legfontosabb adatfolyamok és az őket kibocsátó illetve fogadó személyek, szervezetek vagy rendszerek szerepelnek. Ezzel meghatározható a rendszer határa, kiterjedtsége.

2.8.2. Fizikai adatfolyam-modell

A jelenlegi környezet folyamatait az adatfolyam-modellezési technika segítségével lehet a legjobban felmérni. Ezzel le lehet írni a nagyobb külső objektumokat a rendszeren kívül, amelyek információk forrásai illetve befogadói, a rendszeren belüli folyamatokat, az adatok lerakatait, amelyek időlegesen tárolják az információt, és a közöttük lévő adatfolyamokat. A rajzolás során tisztázódik a felmérés alá vont rendszer kiterjedése, főbb felépítése és működése. A cél a jelenlegi fizikai folyamatok modellezése, az összes hiányossággal, felesleges ismétlődéssel és hibával együtt.

2.8.3. Logikai adatfolyam-modell

A jelenlegi környezet folyamatainak fizikai vonatkozásait itt meg kell szüntetni, az adattárolási kettősségeket fel kell oldani, a folyamatokat pedig logikus szerkezetbe kell rendezni. Itt történik a folyamatok összevetése az adatokkal, egy olyan megfeleltetést adva, amely kizárja, hogy a folyamatok által használt különböző adattárak ugyanazokra az adatokra vonatkozzanak. A cél az, hogy meghatározott szabályok alkalmazásával kiszűrődjenek a fizikai elemek és a felesleges többszörözések a fizikai folyamatok modelljéből, kialakítva egy olyan logikai képet a működésről, amely valószínűleg az új rendszerben is érvényes lesz. Ez a logikai kép lesz az alapja a további lépéseknek, azaz a rendszerszervezési alternatívák kiválasztásának és az igényelt rendszer meghatározásának.

2.9. A modellek elemei

2.9.1. Folyamatok

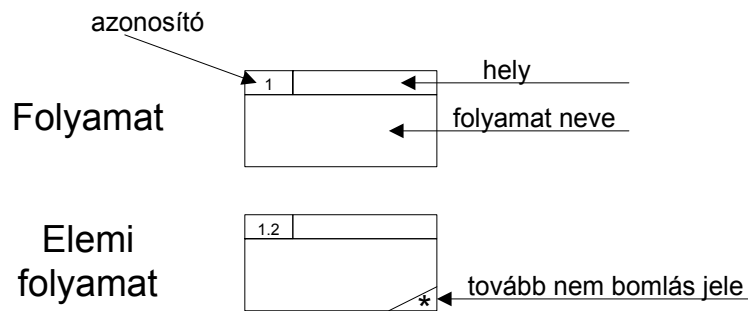
A folyamatok olyan átalakító tevékenységek, melyek a bemenő adatokat kimenő adatokká alakítják.

A folyamatokat egy doboz jelöli, a felső részén két kisebb, elválasztott területtel (azonosító és hely). Minden folyamatnak van egy azonosító sorszáma, de ez nem utal semmilyen sorrendiségre. Minden folyamatnak van egy neve.

A fizikai modell folyamatain meg van jelölve a fizikai hely is, ahol az a folyamat végbemegy, ami általában egy szervezeti egység, vagy egy munkakör neve lehet.

A folyamatok felbomolhatnak, ami tulajdonképpen az adatfolyam ábrák hierarchiáját kialakítja. A felső szinten szereplő folyamatok mindegyikéhez lehet rajzolni egy külön ábrát, ami az adott folyamat egyszerűbb alfolyamatait ábrázolja. Az ilyen alsóbb szintű folyamatokat a tartalmazó folyamat azonosítójával és egy azon belüli sorszámmal van azonosítva.

A tovább nem bomló folyamatokat a jobb alsó sarokban csillaggal jelöltük. Ezek az elemi folyamatok.



2.2. ábra Folyamatok

2.9.2. Adattárak

Az adattárak azok a helyek, ahol az adatok nyugvópontra jutnak a rendszeren belül. Egyik végén nyitott téglalap jelöli őket. Egy azonosítóval és egy névvel rendelkeznek. A rajz áttekinthetősége miatt ugyanazon adattárat meg lehet ismételni. Ilyenkor minden egyes előfordulás egy függőleges vonallal meg van jelölve.

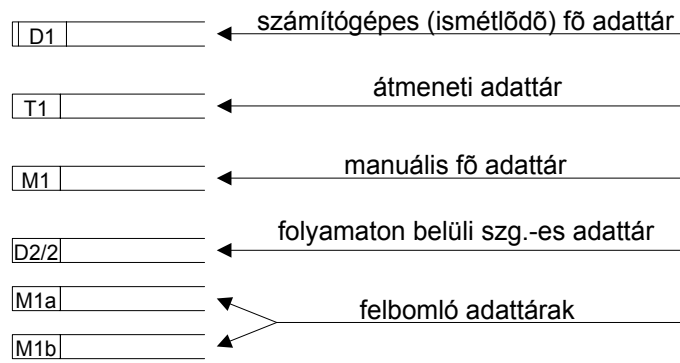
A fizikai rendszer adattárai konkrét helyeket jelölnek.

A logikalizálás után az adattárak már semmilyen fizikai tárolásra történő utalással nem rendelkeznek.

Kétféle adattár van: Állandó (vagy fő) adattár és átmeneti adattár. A fő adattárakat egy 'M' vagy 'D' betű, és egy tetszőleges egyedi szám azonosítja. A 'D' a számítógépes adattárra utal, az 'M' pedig a manuális, azaz kézi adattárra. Az átmeneti adattárakat a 'T' (tranzien) betű és egy szám azonosítja, és olyan helyeket jelölnek, ahol csak ideiglenesen tartózkodnak az adatok, a bekerülés után a következő, ami történhet velük, az a kikerülés. Ha egy átmeneti adattár egyben manuális is, azt egy zárójeles 'M' jelöli a 'T' után.

Ha egy adattár egy alsóbb szintű ábrán jelenik meg, egy adott folyamat belsejében, akkor azt a betűjel után a folyamat azonosítója, egy '/' és egy sorszám azonosítja.

Az adattárak alsóbb szinten felbomolhatnak. Ilyenkor az azonosítójuk a felbontott adattár azonosítójából és egy betű kiegészítésből áll.



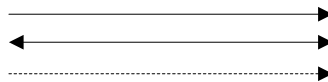
2.3. ábra Adattárak

2.9.3. Adatfolyamok

A rendszerben mozgó információt az adatfolyamok fejezik ki, amiket nyilak jelölnek. A felső szintű ábrán csak a fontosabb adatáramlások jelennek meg, míg a részletek az alsóbb szintű ábrákon.

Minden adatfolyamhoz tartozhat egy név, ami röviden utal a tartalmára.

Az adatok a rendszeren belül csak egy folyamat hatására mozognak, azaz nem léteznek közvetlenül adattárak közötti, illetve külső egyedek és adattárak közötti adatfolyamok.



2.4. ábra Adatfolyamok

3. Az Adatfolyam-modell (DFM) globális elemzése

Az információs rendszerek elemzése és tervezése során bizonyos termékek meghatározó jelentőségűek, a rendszer üzemszerű működésére nézve. Azon termékek, amelyek nagyfokú precizitást, hibamentességet követelnek meg, tehát amelyeknek szigorú minőségi követelményeknek kell megfelelniük, meghatározzák azt, hogy hogyan teljesülnek a végfelhasználó rendszerrel szemben támasztott ügyviteli, illetve üzleti elvárásai.

Egy fejlesztési munka során ez azt jelenti, hogy léteznek olyan részmunkák, amelyek jelentős idő és munkaerő többletet igényelnek. Az ilyen munkák automatizálása, de legalább egyes mozzanatainak számítógépes támogatása jelentős szerepet játszhat az erőforrások megtakarítása, a munkafolyamat felgyorsítása terén.

Ahhoz, hogy egyes tervezési fázisok számítógépes segítséget is felhasználhassanak modellezni kell a megvalósításra kerülő technikát.

Az SSADM-ben az adatfolyam diagramok (DFD-k) több kritikus szakaszban is szerepelnek. Például a megvalósíthatóság-elemzésben, az adott működő rendszer modellezésében, elemzésében és a megvalósítandó információs rendszer tervezésében.

A DFD-k azon termékek közé tartoznak, amelyekről elmondható mindaz, amit a bevezető néhány mondatban említettünk. Minőségi vizsgálata és különböző szempontok szerinti elemzése olyan műveletek, amelyek automatizálása nagyban hozzájárulhat, hogy jól használható, hibáktól mentes adatszolgáltatási rendszer készüljön.

A következőkben bevezetésre kerülő DFDG előállításával olyan modellt állítottunk elő, amely lehetővé teszi az adatfolyam diagramok elemzését, vizsgálatát és tesztelését matematikai módszerekkel. A matematikai modellt alapul véve számítógéppel megvalósítható egy sor olyan algoritmus, amely manuálisan csak fáradságos munkával, összehasonlíthatatlanul nagy hibaelkövetési eséllyel, és nagy időráfordítással készíthető csak el.

Az alábbiakban be fogok vezetni egy irányított gráfot, melynek csomópontjai a DFM külső egyedei, adattárai és processzusai, élei pedig az adatfolyamok és a DFM processzusainak és adattárainak felbontása során kapott tartalmazási relációk.

3.1. Jelölések és definíciók

Vezessük be a következő jelöléseket:

- k_i – **külső egyed**, ahol $i \in N$ sorszám.
- p_c^s – **processzus**, ahol $s \in N$ jelöli, hogy a DFM hierarchikus felépítésében hányadik szinten van; $c \in N$ processzus azonosító, melyet a processzus DFD-beli azonosítójából prímszámkodolással kapunk (pl. 3.4.2-es processzusnál $c=2^3*3^4*5^2=16200$), illetve *nulla*, ha a teljes információs rendszerről van szó.
- $d_{c,i,\alpha}^j$ – **adattár**, ahol $j \in \{1,2,3,4\}$ (1-digitális, 2-manuális, 3-logikai, 4-tranziens) az adattár típusa; $c \in N$ az adattárat tartalmazó processzus kódolt azonosítója ha az adattár lokális, illetve *nulla* ha az adattár globális; $i \in N$ sorszám; $\alpha \in \{\lambda, a, b, c, \dots, z, aa, ab, ac, \dots, az, ba, bb, bc, \dots\}$ a felbomló adattárak i sorszámon belüli azonosítója (2.9.2 fejezet).

Rendezzük ezeket az objektumokat típusonként külön-külön sorba, így ezeket ezentúl egységesen tudjuk kezelni.

3.1.1. Definíció: n_i^t – **csomópont**, ahol $t \in \{1,2,3\}$ (1-külső egyed, 2-processzus, 3-adattár); $i \in N$ típuson belüli sorszám.

3.1.2. Definíció: Legyen $V = \{n_i^t : t \in \{1,2,3\} \text{ és } i \in N\}$ a **csomópontok halmaza**.

3.1.3. Definíció: $e_{i,\alpha}^t(n_{i_1}^{t_1}, n_{i_2}^{t_2})$ – **irányított él** $n_{i_1}^{t_1} \in V$ és $n_{i_2}^{t_2} \in V$ között, ahol $t \in \{1,2\}$ (1-tartalmazási él, 2-adatfolyam él); $i \in N$ sorszám; $\alpha \in \Sigma^*$ ($\Sigma = \{\lambda, a, \acute{a}, b, \dots, z, zs\}$ ábécé) az adatfolyam neve; a következő megszorításokkal:

- ha $t_1 = t_2 \Rightarrow i_1 \neq i_2$ azaz nincs hurokél.
- ha $t = 1$, akkor
 - ◆ $(t_1 = 2 \text{ és } t_2 = 2)$ vagy $(t_1 = 3 \text{ és } t_2 = 3)$, azaz vagy csak processzusok vagy csak adattárak között haladhat tartalmazási él.
 - ◆ az $n_{i_1}^{t_1}$ -hez illetve $n_{i_2}^{t_2}$ -hez hozzárendelt processzusok szintazonosítói (s_1, s_2) között a következő összefüggés kell, hogy fennálljon: $s_2 = s_1 + 1$.
 - ◆ $\alpha = \lambda$.
- ha $t = 2$, akkor
 - ◆ ha $t_1 = 1$ vagy $t_1 = 3 \Rightarrow t_2 \neq 1$ és $t_2 \neq 3$ azaz $t_2 = 2$, ami azt jelenti, hogy külső egyed és adattár között nem lehet adatfolyam él.

3.1.4. Definíció: Legyen $E = \{e_{i,\alpha}^t : t \in \{1,2\}, i \in N \text{ és } \alpha \in \Sigma^*\}$ az **irányított élek halmaza**.

3.2. Adatfolyam Függőségi Gráf (DFDG)

3.2.1. Definíció: DFDG - Dataflow Dependency Graph (Adatfolyam Függőségi Gráf) alatt egy olyan irányított gráfot értünk, melynek pontjai V -beli, élei pedig E -beli elemek.

3.2.2. Definíció: Globális DFDG-nek nevezzük azt a DFDG-t, amelyet a 3.2.1. algoritmus állít elő a DFM-ből.

3.2.1. Algoritmus (Globális DFDG előállítása DFM-ből): A top-level szinttől lefelé haladva hajtsuk végre az alábbiakat:

Előkészítő rész: Vegyük a p_0^0 csomópontot, mint az információs rendszert és képezzük le az n_0^2 -ra. Legyen $r=1$ és térjünk rá az iterációs részre.

Iterációs rész: Az r . szintre végezzük el a következő lépéseket:

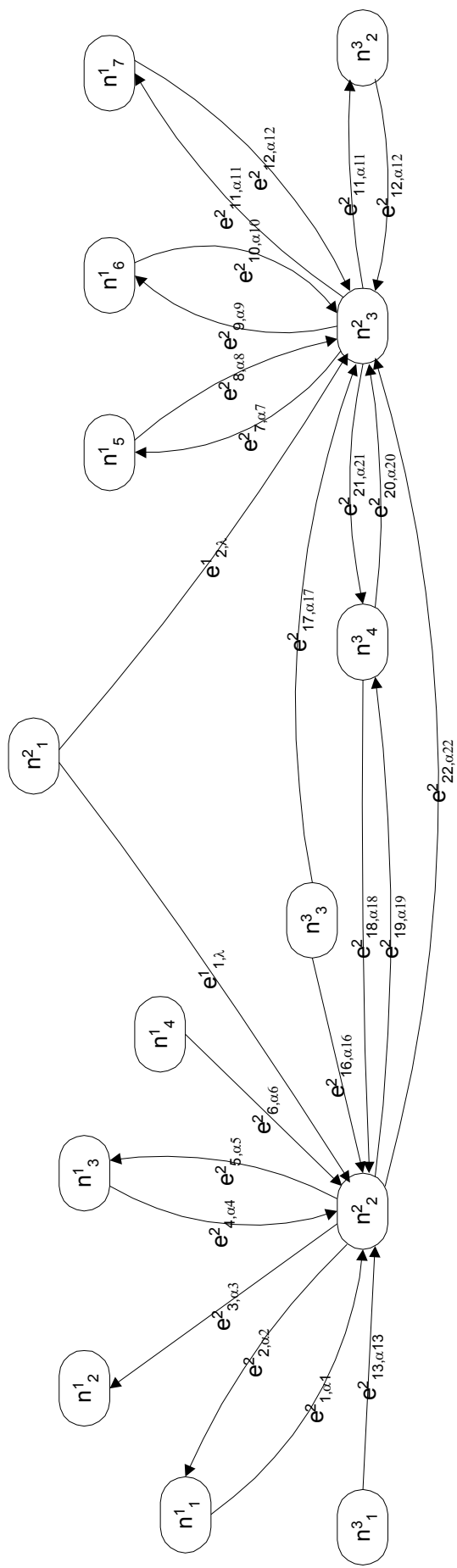
1. Vegyünk egy még fel nem dolgozott r . szintű DFD-t. Ha nincs ilyen, akkor ugorjunk a 7. lépésre.
2. Jelöljük az eddig még fel nem vett külső egyedet k_i -vel, ahol i a legkisebb olyan index, ami még nem szerepel felvett külső egyed indexeként.
3. Jelöljük az eddig még fel nem vett adattárakat $d_{c,i,\alpha}^j$ -vel, ahol j az adattár típusa; c a felbontandó processzus DFD-beli azonosítója (globális adattár esetén nulla); i a legkisebb olyan index, ami még nem szerepel ebben a lépésben felvett adattár indexeként; α a felbomló adattárak megfelelő betűazonosítója.
4. Jelöljük a DFD-n lévő processzusokat p_c^s -val, ahol $s=r$; c azonosító, melyet a DFD-beli azonosítójából prímszámkódolással kapunk.
5. Képezzük le az előző három pontban kapott k_i , $d_{c,i,\alpha}^j$, p_c^s elemeket a soron következő n_i^r csomópontokra, és vegyük fel ezeket a globális DFDG-be.
6. Ugorjunk az 1. lépésre.
7. Vegyük fel az összes olyan $e_{i,\lambda}^1 = (n_{i_1}^3, n_{i_2}^3)$ élet, ahol $n_{i_2}^3$ az $n_{i_1}^3$ adattár felbontásából adódott, a 3. pontban felvett adattár (az adattárak közti tartalmazási élek megadása).
8. Vegyük fel az összes olyan $e_{i,\lambda}^1 = (n_{i_1}^2, n_{i_2}^2)$ élet, ahol $n_{i_2}^2$ az $n_{i_1}^2$ ($r-1$)-dik szintű processzus felbontásából adódott, a 4. pontban felvett processzus (a processzusok közti tartalmazási élek megadása).

9. Vegyük fel a DFD-ken található összes adatfolyamot, mint a gráf adatfolyam éleit. Ha $\exists e_{j_1, \alpha}^2(n_{i_2}^2, n_{i_3}^2), e_{j_2, \alpha}^2(n_{i_1}^2, n_{i_3}^2), e_{j_3, \alpha}^2(n_{i_2}^2, n_{i_4}^2), e_{k_1, \lambda}^1(n_{i_2}^2, n_{i_1}^2), e_{k_2, \lambda}^1(n_{i_3}^2, n_{i_4}^2)$ élek, ahol $n_{i_1}^2$ és $n_{i_4}^2$ r -dik, $n_{i_2}^2$ és $n_{i_3}^2$ pedig $(r-1)$ -dik szintű processzusok, akkor legyen $e_{j_4, \alpha}^2(n_{i_1}^2, n_{i_4}^2)$ új él a globális DFDG-ben (az azonos szinten lévő DFD-k között futó élek megkonstruálása).
10. Ha felvettünk a 3. lépésben olyan $d_{c,i,\alpha}^j$ adattárat, ahol $\alpha \neq \lambda$ (felbontott adattár része), akkor:
- ha $\exists e_{i_1, \beta}^2(n_{i_1}^3, n_{i_3}^2)$, ahol $n_{i_1}^3 = d_{c,i,\alpha}^j$, akkor legyen $e_{i_2, \beta}^2(n_{i_2}^3, n_{i_3}^2)$ új él a globális DFDG-ben, ahol $n_{i_2}^3 = d_{c,i,\lambda}^j$.
 - ha $\exists e_{i_1, \beta}^2(n_{i_3}^2, n_{i_1}^3)$, ahol $n_{i_1}^3 = d_{c,i,\alpha}^j$, akkor legyen $e_{i_2, \beta}^2(n_{i_3}^2, n_{i_2}^3)$ új él a globális DFDG-ben, ahol $n_{i_2}^3 = d_{c,i,\lambda}^j$.
11. Ha a 4. pontban felvett processzusok mindegyike le van zárva (elemi processzus), akkor vége az eljárásnak, ellenkező esetben legyen $r=r+1$ és ugorjunk az 1. lépésre.

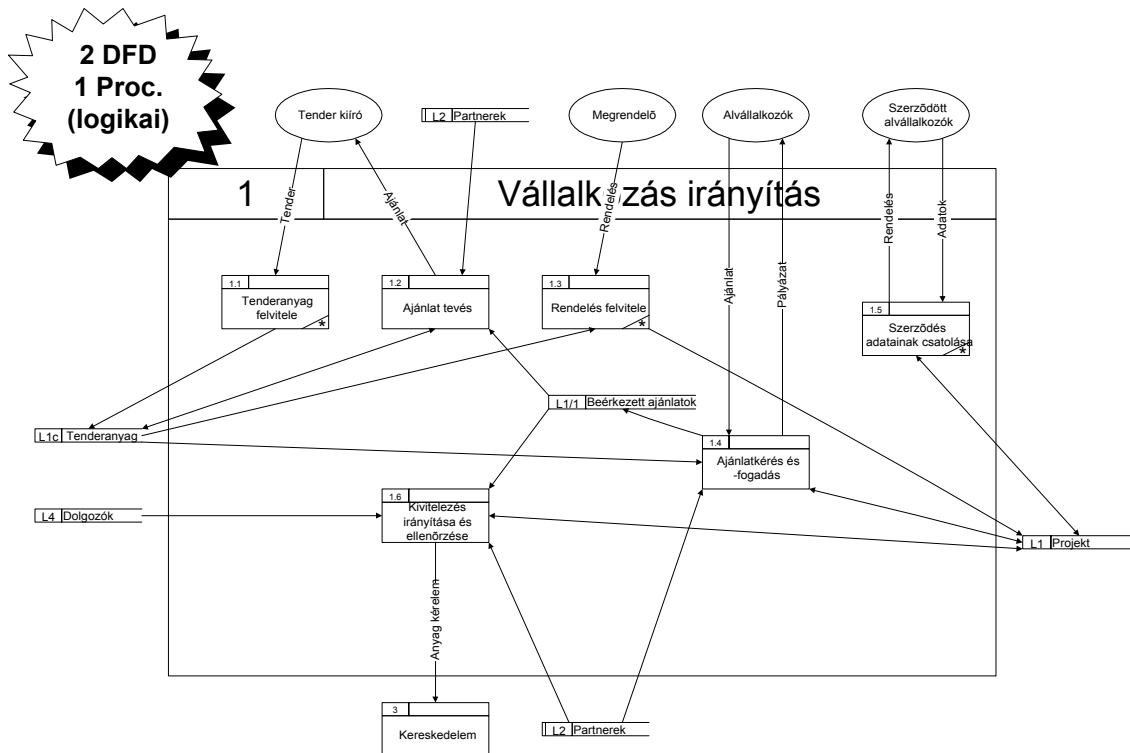
Megjegyzés: A globális DFDG az összes DFD-t egyidejűleg reprezentálja, és lehetőséget teremt a DFD-k által leírt információs rendszer egységes ábrázolására és analizálására.

Példa a Globális Adatfolyam Függőségi Gráfra:

Az alábbiakban kiragadtam egy részletet a 5.4. fejezetben található logikai DFD-kból, amin keresztül demonstrálom a DFDG felépítését a fenti algoritmus segítségével. Mivel az így felépülő DFDG igen bonyolult lenne, a DFD-eket a példa kedvéért lecsökkentettem, elhagytam több elemet. A következőkben megadom azt a két DFD-t amelyek az algoritmus inputját képezik, majd az outputot, a hozzájuk tartozó DFDG-t.

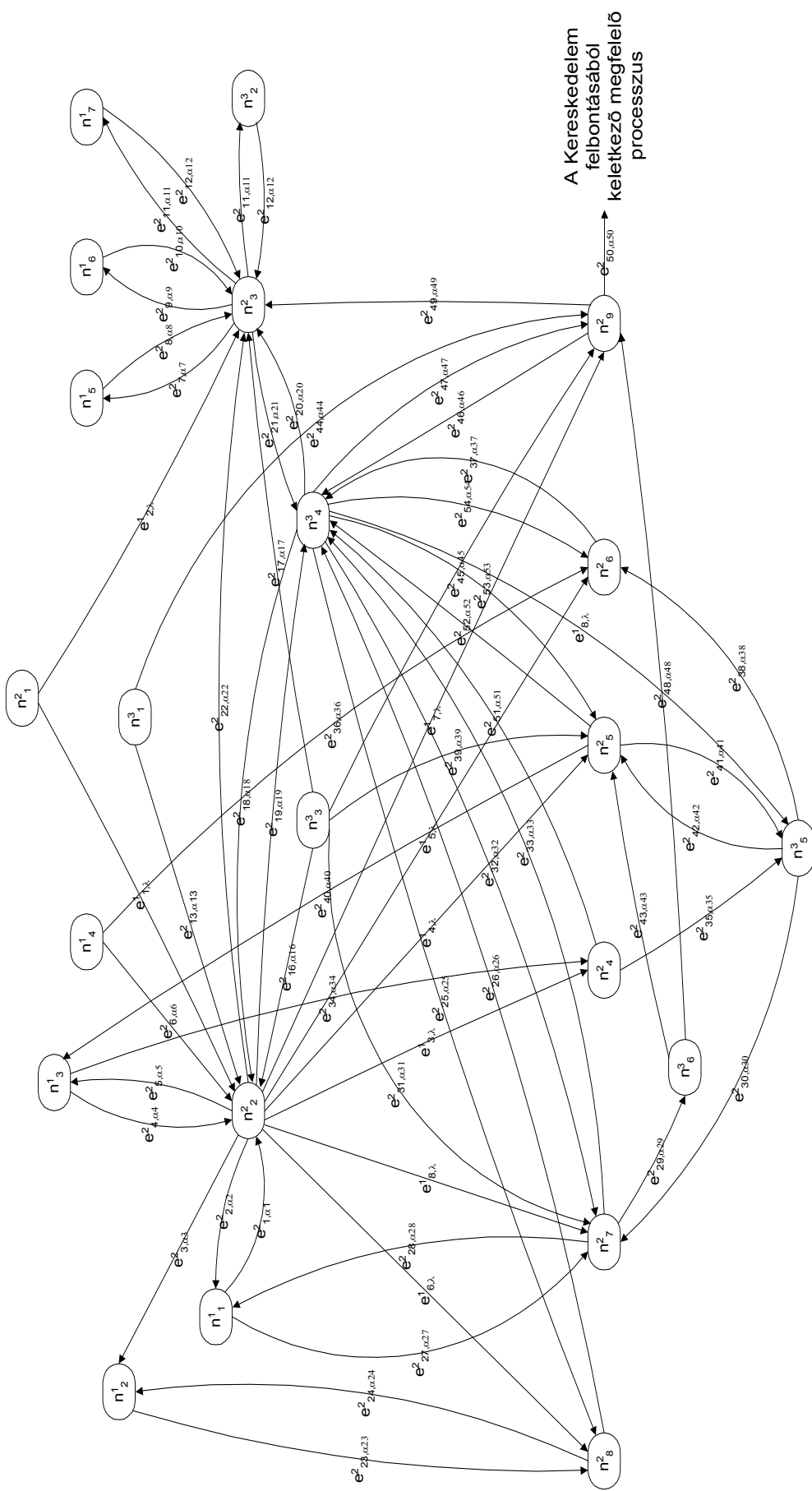


3.2. ábra



3.3. ábra

Az iteráció második lépése a 3.3. ábrán látható második szintű DFD feldolgozásával folytatódik (3.4. ábra). A 2.-4. lépéseiben hozzárendelődnek a DFD-n található – az előző iteráció során még fel nem vett – elemekhez a fejezet elején bevezetett jelölések. Az 5. lépésben áttér az egységes jelölésmódra. Mivel a példánkban most is csak egy DFD van, az algoritmus a 7. Ponttal folytatódik. Mivel az n^3_5 az n^3_4 adattár felbontásából adódott, felvételre kerül az $e^1_{8,\lambda}$ tartalmazási él. A 8. lépésben behúzódnak a tartalmazási élek. A 9. lépésben felvevődnek a DFD-n szereplő összes adatfolyamnak megfelelő gráfbeli élek, valamint az $e^2_{50,\alpha50}$ él. (Az ábrán ez az él azért „lóg a levegőben”, mert a példában nem szerepel az összes 1. szintű processzus felbontása.). A 10. Lépés felveszi az $e^2_{51,\alpha51}$ – $e^2_{54,\alpha54}$ éleket.



3.4. ábra

A következő táblázatokban találhatóak a DFD egyes elemeinek a fejezet elején bevezetett megfelelő jelölések. A táblázatok nem tartalmazzák az összes DFD-kben levő elemet, a többi megfeleltetés ezek alapján könnyen kikövetkeztethető.

Külső egyedek:

DFD-beli név	Külső egyed jelölés	Csomópont jelölés
Alvállalkozók	k_1	n^1_1
Szerződött alvállalkozók	k_2	n^1_2
Tender kiíró	k_3	n^1_3
Megrendelő	k_4	n^1_4
.....

3.1. táblázat**Processzusok:**

DFD-beli név	Processzus jelölés	Csomópont jelölés
Információs rendszer	p^0_0	n^2_1
Vállalkozás irányítás	p^1_2	n^2_2
Kereskedelem	p^1_8	n^2_3
.....

3.2. táblázat**Adattárak:**

DFD-beli név	Adattár jelölés	Csomópont jelölés
Dolgozók	$d^3_{0,1,\lambda}$	n^3_1
Kereskedelmi folyamatok	$d^3_{0,2,\lambda}$	n^3_2
Partnerek	$d^3_{0,3,\lambda}$	n^3_3
Projektek	$d^3_{0,4,\lambda}$	n^3_4
.....

3.3. táblázat

3.2.3. Definíció: Lokális DFDG-nek nevezzük a globális DFDG egy olyan részgráfját, amely pontosan egy DFD-t reprezentál.

Jelölés: A p_c^s processzus felbontását reprezentáló DFD-hez tartozó lokális DFDG-t $DFDG_c$ -vel jelöljük.

3.2.4. Definíció: Lokális DFDG halmaznak nevezzük az összes lokális DFDG-k halmazát.

Az alábbiakban megadok két olyan algoritmust, amelyek segítségével globális DFDG-ből elő tudjuk állítani a lokális DFDG halmazt és fordítva.

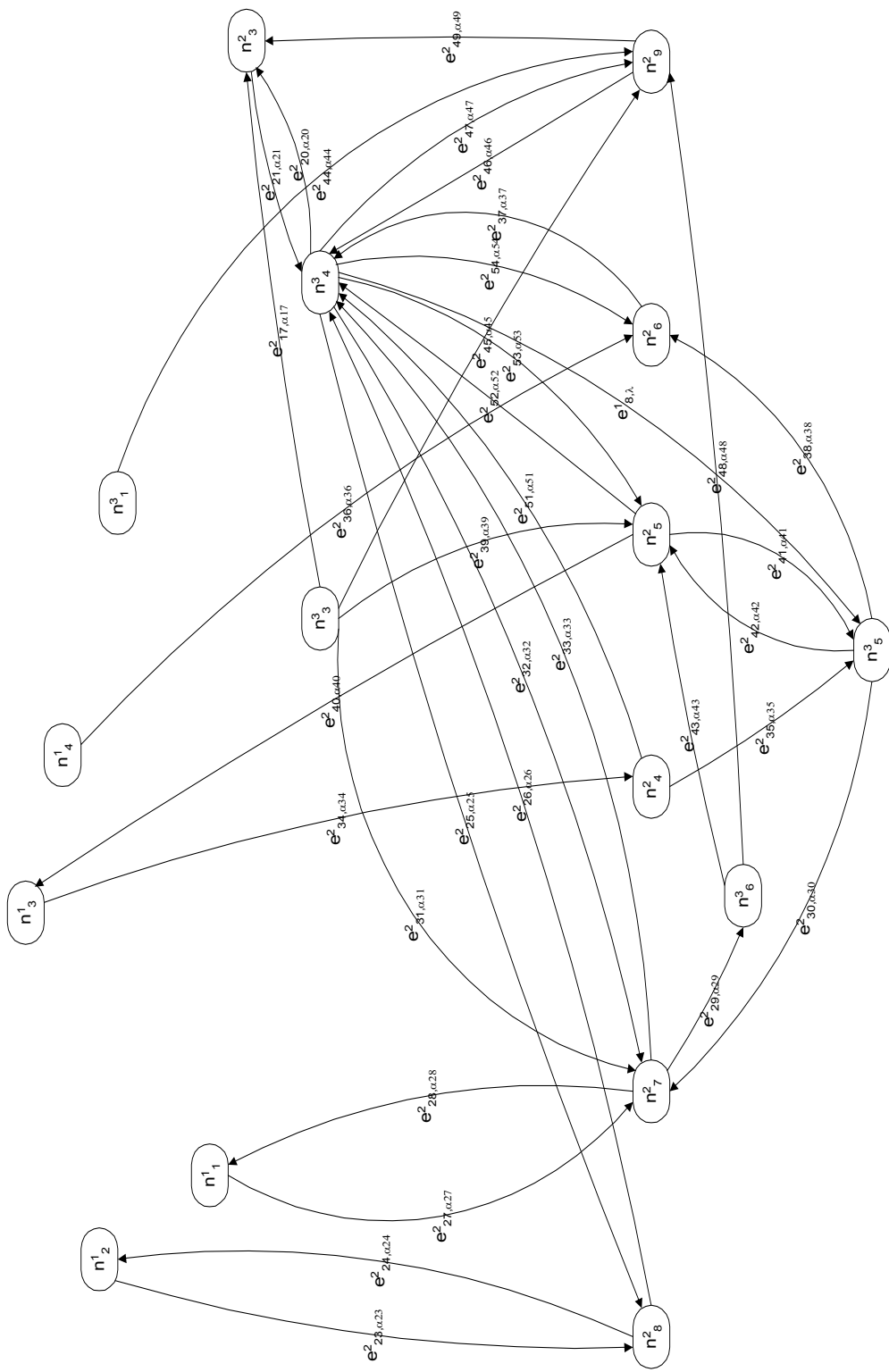
3.2.5. Definíció: PHT - Process Hierarchy Tree - Processzus Hierarchia Fának nevezzük a globális DFDG olyan részgráfját, melyet úgy kapunk, hogy elhagyjuk az összes n_i^t csomópontot, melyre $t \in \{1, 3\}$ és az összes $e_{i,\alpha}^2$ élet.

Megjegyzés: A PHT a processzus tartalmazási hierarchiát szemlélteti (csak a processzusok és a köztük futó tartalmazási élek szerepelnek benne).

3.2.2. Algoritmus (Lokális DFDG halmaz előállítása globális DFDG-ből): A PHT-ben a gyökértől kezdve szintenkénti bejárással a levelek kivételével minden p_c^s csomópontra végezzük el a következő lépéseket a globális DFDG-n (minden egyes iterációt a teljes globális DFDG-re végezzük):

1. Töröljük az összes olyan $p_c^{s'}$ csomópontot, ahol $s' > s+1$, illetve $s' = s+1$ *esetén* azokat, melyeknek más a szülőjük (c, c' prímtenyezős felbontásából megállapítható), valamint a hozzájuk tartozó lokális adattárákat.
2. Töröljük az összes olyan $n_{i_2}^{t_2}$ csomópontot, amelyhez nincs $e_{i_1,\alpha}^2(n_{i_1}^{t_1}, n_{i_2}^{t_2})$ és $e_{j,\beta}^2(n_{i_2}^{t_2}, n_{i_1}^{t_1})$ adatfolyam él, ahol $n_{i_1}^{t_1}$ a p_c^s közvetlen leszármazottja.
3. Nevezzük az így előállított gráfot lokális $DFDG_c$ -nek, és legyen eleme a lokális DFDG halmaznak.

A 3.5. ábrán szemlélhetjük az algoritmus egy iterációs lépésének eredményét. Az algoritmus bemenete a 3.4. ábrán látható globális DFDG, az aktuális csomópont pedig az $n_{i_2}^2$. Könnyen belátható, hogy a kapott gráf megfelel a 3.3. ábrán levő DFD-ből előálló lokális DFDG-nek.



3.5. ábra

3.2.3. Algoritmus (Globális DFDG előállítása lokális DFDG halmazból):

Előkészítő rész: Vegyük fel a globális DFDG-be a p_0^0 -nak megfelelő n_0^2 csomópontot. Legyen $r=0$ és térjünk rá az iterációs részre.

Iterációs rész: Végezzük el a következő lépéseket (r . iteráció):

1. Vegyük sorba a PHT r -edik szintjén található p_c^r processzusokat. Amennyiben egyik processzushoz sem létezik lokális DFDG_c, akkor vége az algoritmusnak.
2. Vegyük fel a globális DFDG-be a p_c^r processzusokhoz tartozó lokális DFDG_c-k minden csomópontját, ha még nem szerepel benne.
3. Vegyük fel az összes olyan $e_{i,\lambda}^1=(n_{i_1}^2, n_{i_2}^2)$ élet, ahol $n_{i_1}^2=p_c^r$, $n_{i_2}^2$ pedig eleme a lokális DFDG_c-nek (a processzusok közti tartalmazási élek megadása).
4. Vegyük fel a lokális DFDG_c-kben található összes adatfolyamot, mint a globális DFDG adatfolyam éleit. Ha $\exists e_{j_1,\alpha}^2(n_{i_2}^2, n_{i_3}^2)$, $e_{j_2,\alpha}^2(n_{i_1}^2, n_{i_3}^2)$, $e_{j_3,\alpha}^2(n_{i_2}^2, n_{i_4}^2)$ élek, ahol $n_{i_2}^2=p_{c_1}^r$, $n_{i_3}^2=p_{c_2}^r$ ($c_1 \neq c_2$), valamint $n_{i_1}^2 \in V(\text{lokális DFDG}_{c_1})$ és $n_{i_4}^2 \in V(\text{lokális DFDG}_{c_2})$, akkor legyen $e_{j_4,\alpha}^2(n_{i_1}^2, n_{i_4}^2)$ új él a globális DFDG-ben (az azonos szinten lévő DFD-k között futó élek megkonstruálása).
5. Legyen $r=r+1$ és ugorjunk az 1. pontra.

3.2.1. Tétel: A globális DFDG és a lokális DFDG halmaz egyértelműen előállíthatók egymásból.

Bizonyítás: Az állítás a 3.2.2. és a 3.2.3. algoritmusokból következik. \square

3.2.4. Algoritmus (DFM előállítása lokális DFDG halmazból): Végezzük el a következő lépéseket a lokális DFDG halmaz minden elemére:

1. Legyen a lokális DFDG_c a soron következő.
2. Hozzunk létre egy új, üres DFD-t, mely a p_c^s processzus felbontását fogja szemléltetni.
3. Keressük meg a lokális DFDG_c-ben a külső egyedeket reprezentáló csomópontokat majd vegyük fel a DFD-be az ezeknek a csomópontoknak megfelelő külső egyedeket.
4. Keressük meg a lokális DFDG_c-ben a processzusokat reprezentáló csomópontokat majd vegyük fel a DFD-be az ezeknek a csomópontoknak megfelelő processzusokat.
5. Keressük meg a lokális DFDG_c-ben az adattárákat reprezentáló csomópontokat majd vegyük fel a DFD-be az ezeknek a csomópontoknak megfelelő adattárákat.
6. A lokális DFDG_c-ben szereplő adatfolyam éleknek megfelelő adatfolyamokat rajzoljuk be a DFD-be.

3.2.2. Tétel: A lokális DFDG halmazból előállítható az összes DFD.

Bizonyítás: A 3.2.4. algoritmusból következik. □

Következmény: A lokális DFDG_c-ből képezhető a *c* processzus azonosítójú folyamat felbontásából adódó DFD.

3.3. Konzisztencia vizsgálatok

3.3.1. Definíció: Egy **DFM** akkor **konzisztens**, ha a hozzá tartozó DFD-kre teljesülnek a következők:

- a) Adott felsőbb szintű folyamat összes be- és kimeneti adatfolyamának szerepelnie kell a lebontás következő szintjén is.
- b) Akkor szerepelhet egy DFD-n lokális adattár, ha egynél több vele azonos szintű folyamat használja.
- c) Az adatok a rendszeren belül csak egy folyamat hatására mozoghatnak, azaz nem létezhetnek közvetlenül adattárak közötti, közvetlenül külső egyedek közötti illetve külső egyedek és adattárak közötti adatfolyamok.
- d) A processzusok nem lehetnek adatok forrásai illetve végfelhasználói (legalább egy bemenő és egy kimenő élnek léteznie kell).
- e) Az adattárakba kell mind bemenő, mind kimenő adatfolyam, azaz minden adatot valamikor létre kell hozni és valamikor fel kell használni (legalább egy bemenő és egy kimenő élnek léteznie kell).

3.3.2. Definíció: Egy **globális DFDG** akkor **konzisztens**, ha teljesülnek a következők:

- a) Ha $\exists e_{j_1, \alpha}^2(n_{i_1}^2, n_{i_2}^2)$, akkor $\exists e_{j_2, \alpha}^2(n_{i_1}^2, n_{i_4}^2)$, $e_{j_3, \alpha}^2(n_{i_3}^2, n_{i_2}^2)$ és $e_{j_4, \alpha}^2(n_{i_3}^2, n_{i_4}^2)$ élek, ahol $n_{i_1}^2$ és $n_{i_2}^2$ *r*-dik szintű processzusok, és $\exists e_{k_1, \lambda}^1(n_{i_1}^2, n_{i_3}^2)$ és $e_{k_2, \lambda}^1(n_{i_2}^2, n_{i_4}^2)$ élek.
- b) Ha $\exists e_{j_1, \alpha}^2(n_{i_1}^2, n_{i_2}^t)$, ahol $t \in \{1, 3\}$, akkor $\exists e_{j_2, \alpha}^2(n_{i_3}^2, n_{i_2}^t)$, $e_{k_1, \lambda}^1(n_{i_1}^2, n_{i_3}^2)$.
- c) Ha $\exists e_{j_1, \alpha}^2(n_{i_2}^t, n_{i_1}^2)$, ahol $t \in \{1, 3\}$, akkor $\exists e_{j_2, \alpha}^2(n_{i_2}^t, n_{i_3}^2)$, $e_{k_1, \lambda}^1(n_{i_1}^2, n_{i_3}^2)$.
- d) Ha $\exists d_{c, i, \alpha}^j$ adattár, ahol $\alpha \neq \lambda$ (felbontott adattár része), akkor:
 - ha $\exists e_{i_1, \beta}^2(n_{i_1}^3, n_{i_3}^2)$, ahol $n_{i_1}^3 = d_{c, i, \alpha}^j$, akkor $\exists e_{i_2, \beta}^2(n_{i_2}^3, n_{i_3}^2)$ él, ahol $n_{i_2}^3 = d_{c, i, \lambda}^j$.
 - ha $\exists e_{i_1, \beta}^2(n_{i_3}^2, n_{i_1}^3)$, ahol $n_{i_1}^3 = d_{c, i, \alpha}^j$, akkor $\exists e_{i_2, \beta}^2(n_{i_3}^2, n_{i_2}^3)$ él, ahol $n_{i_2}^3 = d_{c, i, \lambda}^j$.
- e) Ha egy $d_{c, i, \alpha}^j$ csomópont lokális adattárat reprezentál, akkor léteznie kell két olyan élnek, amelyeknek egyik végpontja a $d_{c, i, \alpha}^j$ csomópont, a másik pedig két különböző processzus.

- f) A processzusokat és az adattárákat reprezentáló csomópontoknak legalább egy bemenő és egy kimenő éllel kell rendelkezniük.

3.3.1. Tétel: Ha egy DFM konzisztens, akkor a 3.2.1. algoritmussal belőle létrehozott globális DFDG is konzisztens.

Bizonyítás: Azt kell belátni, hogy ha a DFM teljesíti a 3.3.1. definícióban leírt feltételeket, akkor a globális DFDG teljesíti a 3.3.2. definícióban leírtakat. Vegyük sorra a 3.3.2. definícióban felsorolt feltételeket. Az **a)**, **b)**, **c)** tulajdonságok következnek a 3.3.1. definíció **a)** pontjából, valamint a 3.2.1. algoritmus 9-edik lépéséből. A **d)** tulajdonságot a 3.2.1. algoritmus 10-edik lépése biztosítja. Az **e)** tulajdonság következik a 3.3.1. definíció **b)** pontjából, valamint a 3.2.1. algoritmus 3. és 9. lépéséből. Az **f)** tulajdonság következik a 3.3.1. definíció **d)** és **e)** pontjából valamint a 3.2.1. algoritmus 9. lépéséből. \square

3.3.3. Definíció: Egy **lokális DFDG halmaz** akkor **konzisztens**, ha teljesülnek a következők:

- a) Ha egy lokális DFDG-ben $\exists e_{i,\alpha}^2(n_{i_1}^2, n_{i_2}^t)$, ahol $t \in \{1, 2, 3\}$, és $n_{i_1}^2 = p_c^s$, akkor a lokális DFDG_c-ben $\exists e_{j,\alpha}^2(n_{i_3}^2, n_{i_2}^t)$.
- b) Ha egy lokális DFDG-ben $\exists e_{i,\alpha}^2(n_{i_2}^t, n_{i_1}^2)$, ahol $t \in \{1, 2, 3\}$, és $n_{i_1}^2 = p_c^s$, akkor a lokális DFDG_c-ben $\exists e_{j,\alpha}^2(n_{i_2}^t, n_{i_3}^2)$.
- c) Ha $\exists d_{c,i,\alpha}^j$ adattár, ahol $\alpha \neq \lambda$ (felbontott adattár része), akkor:
- ha $\exists e_{i,\beta}^2(n_{i_1}^3, n_{i_3}^2)$, ahol $n_{i_1}^3 = d_{c,i,\alpha}^j$, akkor $\exists e_{i_2,\beta}^2(n_{i_2}^3, n_{i_3}^2)$ él, ahol $n_{i_2}^3 = d_{c,i,\lambda}^j$.
 - ha $\exists e_{i,\beta}^2(n_{i_3}^2, n_{i_1}^3)$, ahol $n_{i_1}^3 = d_{c,i,\alpha}^j$, akkor $\exists e_{i_2,\beta}^2(n_{i_3}^2, n_{i_2}^3)$ él, ahol $n_{i_2}^3 = d_{c,i,\lambda}^j$.
- d) Ha egy $d_{c,i,\alpha}^j$ csomópont lokális adattárat reprezentál, akkor léteznie kell két olyan élnek, amelyeknek egyik végpontja a $d_{c,i,\alpha}^j$ csomópont, a másik pedig két különböző processzus.
- e) A processzusokat és az adattárákat reprezentáló csomópontoknak legalább egy bemenő és egy kimenő éllel kell rendelkezniük.

3.3.2. Tétel: Ha a globális DFDG konzisztens, akkor a 3.2.2. algoritmussal előállított lokális DFDG halmaz is konzisztens.

Bizonyítás: Azt kell belátni, hogy ha a globális DFDG teljesíti a 3.3.2. definícióban leírt feltételeket, akkor a lokális DFDG halmaz teljesíti a 3.3.3. definícióban leírtakat. Vegyük sorra a 3.3.3. definícióban felsorolt feltételeket. Az **a), b)** tulajdonságok következnek a 3.3.2. definíció **a), b), c)** pontjaiból, a 3.2.2. algoritmus pedig nem törli ezeket az éleket. A **d), e)** tulajdonságok egyértelműen következnek a 3.3.2. definíció **e), f)** pontjaiból, a 3.2.2. algoritmus pedig nem törli ezeket az éleket. □

3.3.3. Tétel: Ha a lokális DFDG halmaz konzisztens, akkor a 3.2.3. algoritmussal előállított globális DFDG is konzisztens.

Bizonyítás: Azt kell belátni, hogy ha a lokális DFDG halmaz teljesíti a 3.3.3. definícióban leírt feltételeket, akkor a globális DFDG teljesíti a 3.3.2. definícióban leírtakat. Vegyük sorra a 3.3.2. definícióban felsorolt feltételeket. Az **a), b), c)** tulajdonságok következnek a 3.3.3. definíció **a), b)** pontjaiból, valamint a 3.2.3. algoritmus 4. lépéséből. Az **d)** tulajdonság következik a 3.3.3. definíció **d)** pontjaiból, valamint a 3.2.3. algoritmus 4. lépéséből. Az **e), f)** tulajdonságok következnek a 3.3.3. definíció **d), e)** pontjaiból, valamint a 3.2.3. algoritmus 4. lépéséből. □

3.3.4. Tétel: Ha a lokális DFDG halmaz konzisztens, akkor a 3.2.4. algoritmussal előállított DFM is konzisztens.

Bizonyítás: Az algoritmus menetéből könnyen kikövetkeztethető. □

Következmény: A globális DFDG akkor és csak akkor konzisztens, ha a lokális DFDG halmaz is konzisztens.

3.3.5. Tétel: Egy DFM akkor és csak akkor konzisztens, ha a 3.2.1. algoritmussal belőle létrehozott globális DFDG is konzisztens.

Bizonyítás:

⇒: A 3.3.1. tételből közvetlenül adódik.

⇐: A globális DFDG-ből a 3.2.2. algoritmus segítségével hozzuk létre a lokális DFDG halmazt, majd ebből a 3.2.4. algoritmussal a DFM-et. Mivel a 3.3.2. és 3.3.4. tétel szerint mindkét algoritmus megőrzi a konzisztenciát, az állítást beláttuk. □

3.4. Konzisztens műveletek a globális DFDG-n

3.4.1. Definíció: Konzisztens egy művelet, ha egy konzisztens globális DFDG-n végrehajtva az eredményül kapott globális DFDG is konzisztens.

A következőkben megadok egy **adatfolyam él beszűrő** algoritmust, mely konzisztens műveletet valósít meg. Az új élre teljesülnie kell, hogy az egyik végpontja egy még fel nem bontott processzus, a másik végpontja pedig egy:

- külső egyed, vagy
- még fel nem bontott processzus, vagy
- nem felbomló adattár, vagy egy felbomlottak egy részadattára.

3.4.1. Algoritmus (Adatfolyam él hozzávétele a globális DFDG-hez): Legyen a beszűrendő él $e_{j,\alpha}^2(n_{i_1}^2, n_{i_2}^t)$.

1. Ha $t=1$, akkor vegyük fel az $e_{j,\alpha}^2$ élet és ha az $e_{i,\lambda}^1=(n_{i_3}^2, n_{i_1}^2)$ éleben $n_{i_3}^2 \neq p_0^0$, akkor hívjuk meg rekurzívan az algoritmust az $e_{j,\alpha}^2(n_{i_3}^2, n_{i_2}^t)$ -re.
2. Ha $t=2$, valamint $n_{i_1}^2$ és $n_{i_2}^t$ egy szinten vannak, akkor vegyük fel az $e_{j,\alpha}^2$ élet és ha az $n_{i_1}^2$ őse $p_{c_1}^s$, az $n_{i_2}^t$ őse pedig $p_{c_2}^s$ és $p_{c_1}^s \neq p_{c_2}^s$, akkor legyenek $e_{j_1,\alpha}^2(n_{i_1}^2, p_{c_2}^s)$ és $e_{j_2,\alpha}^2(p_{c_1}^s, n_{i_2}^t)$ új élek a gráfban, majd hívjuk meg rekurzívan az algoritmust az $e_{j,\alpha}^2(p_{c_1}^s, p_{c_2}^s)$ -re.
3. Ha $t=2$, valamint $n_{i_1}^2$ és $n_{i_2}^t$ különböző szinteken vannak, akkor vegyük fel az $e_{j,\alpha}^2$ élet és ha az $n_{i_1}^2$ őse $p_{c_1}^{s_1}$, az $n_{i_2}^t$ őse pedig $p_{c_2}^{s_2}$ akkor, ha $s_1 > s_2$, $e_{j_1,\alpha}^2(p_{c_1}^{s_1}, n_{i_2}^t)$ -re, ellenkező esetben pedig $e_{j_2,\alpha}^2(n_{i_1}^2, p_{c_2}^{s_2})$ -ra hívjuk meg rekurzívan az algoritmust.
4. Ha $t=3$ és $n_{i_2}^t$ nem felbomló adattár, akkor vegyük fel az $e_{j,\alpha}^2$ élet és ha az $e_{i,\lambda}^1=(n_{i_3}^2, n_{i_1}^2)$ éleben $n_{i_2}^2 \neq p_0^0$, akkor hívjuk meg rekurzívan az algoritmust az $e_{j,\alpha}^2(n_{i_3}^2, n_{i_2}^t)$ -re.
5. Ha $t=3$ és $n_{i_2}^t$ egy felbomló adattár része, akkor vegyük fel az $e_{j,\alpha}^2$ és az $e_{j_1,\alpha}^2(n_{i_3}^2, n_{i_2}^t)$ éleket, ahol $e_{i,\lambda}^1=(n_{i_3}^2, n_{i_2}^t)$.

Az $e_{j,\alpha}^2(n_{i_2}^t, n_{i_1}^2)$ élre értelemszerűen megfordítjuk az algoritmusban szereplő összes adatfolyam él irányát.

Az él beszűrő algoritmus mellé megadok egy **adatfolyam él törlő** algoritmust is, mely szintén konzisztens műveletet valósít meg. A törlendő élre ugyanazoknak a tulajdonságoknak kell teljesülnie, mint az él beszűrő algoritmusnál beszűrendő élre.

3.4.2. Algoritmus (Adatfolyam él törlése a globális DFDG-ből): Legyen a törlendő él $e_{j,\alpha}^2(n_{i_1}^2, n_{i_2}^t)$.

1. Ha $t=1$, akkor töröljük az $e_{j,\alpha}^2$ élet és ha az $e_{i,\lambda}^1=(n_{i_3}^2, n_{i_1}^2)$ élben $n_{i_2}^2 \neq p_0^0$, akkor hívjuk meg rekurzívan az algoritmust az $e_{j',\alpha}^2(n_{i_3}^2, n_{i_2}^t)$ -re.
2. Ha $t=2$, valamint $n_{i_1}^2$ és $n_{i_2}^t$ egy szinten vannak, akkor töröljük az $e_{j,\alpha}^2$ élet és ha az $n_{i_1}^2$ öse $p_{c_1}^s$, az $n_{i_2}^t$ öse pedig $p_{c_2}^s$ és $p_{c_1}^s \neq p_{c_2}^s$, akkor töröljük az $e_{j_1,\alpha}^2(n_{i_1}^2, p_{c_2}^s)$ és $e_{j_2,\alpha}^2(p_{c_1}^s, n_{i_2}^t)$ éleket a gráfból, majd hívjuk meg rekurzívan az algoritmust az $e_{j',\alpha}^2(p_{c_1}^s, p_{c_2}^s)$ -re.
3. Ha $t=2$, valamint $n_{i_1}^2$ és $n_{i_2}^t$ különböző szinteken vannak, akkor töröljük $e_{j,\alpha}^2$ élet és ha az $n_{i_1}^2$ öse $p_{c_1}^{s_1}$, az $n_{i_2}^t$ öse pedig $p_{c_2}^{s_2}$ akkor, ha $s_1 > s_2$, $e_{j_1,\alpha}^2(p_{c_1}^{s_1}, n_{i_2}^t)$ -re, ellenkező esetben pedig $e_{j_2,\alpha}^2(n_{i_1}^2, p_{c_2}^{s_2})$ -ra hívjuk meg rekurzívan az algoritmust.
4. Ha $t=3$ és $n_{i_2}^t$ nem felbomló adattár, akkor töröljük az $e_{j,\alpha}^2$ élet és ha az $e_{i,\lambda}^1=(n_{i_3}^2, n_{i_1}^2)$ élben $n_{i_2}^2 \neq p_0^0$, akkor hívjuk meg rekurzívan az algoritmust az $e_{j',\alpha}^2(n_{i_3}^2, n_{i_2}^t)$ -re.
5. Ha $t=3$ és $n_{i_2}^t$ egy felbomló adattár része, akkor töröljük az $e_{j,\alpha}^2$ és az $e_{j',\alpha}^2(n_{i_3}^2, n_{i_2}^t)$ éleket, ahol $e_{i,\lambda}^1=(n_{i_3}^3, n_{i_2}^t)$.

Az $e_{j,\alpha}^2(n_{i_2}^t, n_{i_1}^2)$ élre értelemszerűen megfordítjuk az algoritmusban szereplő összes adatfolyam él irányát.

Csomópont felvétele akkor lesz konzisztens, ha teljesítjük az alábbiakban felsorolt megkötéseket (az élek felvétele a 3.4.1. algoritmussal történhet):

- Külső egyed esetén fel kell venni legalább egy ki- vagy bemenő adatfolyam élet is.
- Processzusnál fel kell venni legalább egy ki- és egy bemenő adatfolyam élet is.
- Adattár esetén fel kell venni legalább egy ki- és egy bemenő adatfolyam élet is.

Csomópont törlésénél ügyelnünk kell arra, hogy nem törölhető az a csomópont, melyből tartalmazási él indul. A törlés úgy történik, hogy töröljük a csomópontot, valamint az összes belőle induló és bele érkező élet a 3.4.2. algoritmussal.

Tekintsük át röviden az ebben a fejezetben ismertetett alapvető konzisztens műveletek gyakorlati jelentőségét. Mivel a globális DFDG egy DFM által leírt információs rendszert reprezentál, a műveletek alkalmazása a rendszert tekintve működésbeli változtatást jelent. A fenti műveletek ügyelnek arra, hogy csak olyan változtatásokat lehessen eszközölni, melyek nem sértik a rendszer integráltságát.

3.5. A konstrukcióban rejlő további lehetőségek

Backward slice (hátra szeletelés): olyan részgráf keresése, amely tartalmazza azokat a csomópontokat és adatfolyam éleket, melyektől az aktuális csomópont függ.

Forward slice (előre szeletelés): olyan részgráf keresése, amely tartalmazza azokat a csomópontokat és adatfolyam éleket, melyek az aktuális csomóponttól függenek.

A slicing-eljárások elsődleges szerepe a hibakereséseknél (debugging) nyilvánul meg. Hiba esetén így viszonylag könnyen meghatározható a hiba keletkezésének helye, illetve a hiba mely csomópontokra van kihatással.

Megadhatóak olyan algoritmusok, melyekkel meghatározhatjuk, hogy egy adattárba mely külső egyedek és folyamatok szolgáltatnak adatokat, illetve egy külső egyed vagy folyamat, mely adattárakból kap adatokat.

Független komponensek keresése: Olyan maximális részgráfok keresése, melyeknek csúcspontjai nem függenek közvetlenül más csúcspontoktól.

Az ilyen részgráfok egymástól függetlenül kezelhetőek, lehetővé téve így a párhuzamos munkát, azaz egy időben több szálon is folyhat a rendszer fejlesztése. Másik előnye, hogy a kész rendszer adatfeldolgozását többprocesszoros környezetre optimalizálhatjuk.

4. Felhasználói igények elemzése

4.1. A feladat ismertetése

Feladatunk egy országos kiterjedésű építkezési vállalat jelenlegi információs rendszerének elemzése, hátrányainak feltárása, valamint ennek jövőbeni fejlesztési alternatíváinak kidolgozása volt. A cég Magyarországon az első nyolc vállalat egyikeként nyerte el az ISO 9001-es minősítést. Működési formájára jellemző, hogy fővállalkozóként vállalja a megbízásokat (egy időben akár többet is), a munka nagyobb részét alvállalkozókkal végezteti el, de bizonyos részműveleteket önerőből is megoldhat.

A vállalat egy országos központtal, néhány nagyobb városban levő irodával, valamint több mobil telephellyel rendelkezik. Ezek között az információáramlás jelenleg még nem számítógépesített, telefonra és faxra van korlátolva. A cég kitűnően el van látva hardverrel, ami azonban csak a CAD rendszerek használatának köszönhető, számítógépes információs rendszerről a mai modern értelemben még nem beszélhetünk. Az egyes osztályokon már léteznek adatbázis kezelő programok, melyek a feladatuknak részben eleget tesznek, azonban ezek különböző fejlesztésűek, heterogén tárolási formákat alkalmaznak, együttműködésük megoldatlan.

Kiindulási alapként a cég Műszaki Fejlesztés Osztály vezetőjétől a következő kétoldalas vázlatot kaptuk kézhez:

A projektorientált nyilvántartási rendszer lényege, hogy a cég folyamatait olyan módon legyen képes lekezelni, hogy a nyilvántartás egy online projektinformációs rendszerbe legyen integrálva.

A rendszer fő részei:

- I. Projektnyilvántartás
 - Ajánlatadási stádium
 - ◆ Projekt ismert adatainak felvétele
 - ◆ Ajánlatkérések kiküldése
 - ◆ Érkező ajánlatok és egyéb információk gyűjtése, értékelése, elemzése
 - A cég, mint vállalkozó által kötött szerződések
 - ◆ Szerződésre vonatkozó adatok átvétele az ajánlati adatokból
 - ◆ Szerződés alapadatainak rögzítése
 - ◆ Módosítások lekezelése

- Kapcsolódó szerződések
 - ◆ Szerződésre vonatkozó adatok átvétele az ajánlati adatokból
 - ◆ Szerződés alapadatainak rögzítése
 - ◆ Módosítások lekezelése
 - Szerződések pénzügyi állapotának online jelzése
 - ◆ Szerződés adatok és a számlarendszerbe bevitt számlák alapján jelezve a cég összesenre, szerződésenként, alszerződésenként a szükséges adatokat
 - Jelentések
 - ◆ Különböző listák előállítása
 - ◆ Szükséges statisztikai jelentések előállítása
 - ◆ Szükséges vezetői információk előállítása
- II. Egységes partneryilvántartás
- Egységes, közös partner adatbázis
 - ISO minősítések nyilvántartása
 - Feleljen meg valamennyi rendszerbe integrált programnak
 - Feleljen meg a titkárságnak, telefonszám keresésre, levélcímzésekre, számok felhívására, faxok küldésére, e-mail-re
 - Lehetővé kell tenni, hogy a partnereket bárki az adatbázisból automatikusan hívathassa fel, küldhessen részükre faxot
- III. Kimenő és bejövő információk nyilvántartása és a projekthez kapcsolása
- Levelezések
 - Faxok
 - Telefonok
 - Elektronikus küldemények
- IV. Számlanyilvántartás
- Számlarendszer működése a jelenlegi funkciókkal, de a projektek a számlarendszer adatbázisát olvassák.
- V. Munkaszámra kifizetett bérek offline módon, havonta:
- A jelenlegi bérrendszerhez egy program a havi munkaszámra vonatkozó bérfizetési adatokat egy fájlba kiírja, melyet a projekt program használ.

A programrendszert egy megfelelő keretrendszerbe kell ágyazni, amely az információkat egységesen kezeli, és lehetővé teszi a csoportmunkát.

A programrendszernek a cég ISO eljárásaira kell épülnie, illetve az eseményeket az abban leírt módon lehessen lekezelni.

A vállalat fix telephelyeit, és az ideiglenes építési helyeket össze kell kötni először internettel, majd online hálózattal egy közös hálózatba: ez lehetővé teszi az információküldés idejének lerövidítését, költségek csökkentését (helyi beszélgetési

díj + a kapcsolattartási díj a fizetendő összeg). Olcsóbb és gyorsabb lesz, mint a fax. Rajzfájlok, színes képek, videók, hangok is küldhetők.

Meg kell valósítani a cégen belüli e-mail szolgáltatást.

A munkatársak képzésére időt kell szakítani.

4.2. Az igényelt információs rendszer főbb részei

4.2.1. Projektnyilvántartás

A leendő információs rendszer egyik legfontosabb modulja. Alapvető feladata, hogy a projektet a kezdettől, vagyis az ajánlatadási stádiumtól kezdve végigvezesse, pótolva a jelenlegi rendszer hiányosságát, mely a projektet csak a szerződés megkötésétől számítva tekinti létezőnek. Ez által a rendszerben meg fognak jelenni azon kiadások és bevételek is, melyek a tender megvásárlásával, illetve eladásával kapcsolatosak. Ezek a jelenlegi rendszerben nem kerültek nyilvántartásba.

Másrészt lehetővé fogja tenni a projekt ismert adatainak felvételét, mely magába foglalja a pályázatok alapján kiküldött és beérkezett ajánlatok adatainak rögzítését. Továbbá tartalmazni fogja a cég, mint vállalkozó által kötött szerződésekre-, illetve a projekthez kapcsolódó további szerződésekre vonatkozó adatok átvételét az ajánlati adatokból, a szerződés alapadatainak rögzítését, valamint a felmerülő módosítások lekezelését.

További elengedhetetlen elvárás a rendszerrel szemben, hogy egy adott projektre vonatkozó szerződések pénzügyi állapotát online módon megjelenítse, illetve képes legyen a vállalat pillanatnyi likviditásáról információt nyújtani, különböző jellegű lekérdezéseket lehetővé tenni, statisztikai jelentéseket, szükséges vezetői információkat előállítani. Ezért a figyelem az irányítási szükségletből fakadó döntéstámogatás és az azt kiszolgáló adatfeldolgozási háttér összhangjának létrehozására is összpontosult.

Általánosságban elmondhatjuk azt, hogy a kiépítendő rendszer elsődleges célja, hogy az információfeldolgozás révén elégítse ki a hatékony irányítás kritériumait, másfelől tegye ezt meg úgy, hogy lehetőség nyíljon a nagy mennyiségű adat feldolgozására, méghozzá gyorsan, és úgy, hogy támogassa vagy tegye lehetővé bonyolult döntések meghozatalát.

4.2.2. Egységes partneryilvántartás

A cég működési rendszeréből adódhat olyan vállalkezési forma, hogy az elnyert tender számos részfeladatát alvállalkozókkal végezteti el. Ezért a tender kiírók adatai mellett nyilván kell tartani az összes alvállalkozót is egy egységes, közös, jól strukturált partner adatbázisban.

Itt meg kell jegyezni, hogy a jelenlegi rendszer tárolási anomáliákat mutat, ugyanazon partnerek több helyen különböző módon vannak nyilvántartva.

Mivel a cég az ISO 9001 szabvány szerint működik, szükséges a partnerek ISO minősítésének nyilvántartása. Ezen minősítési kategóriák megállapítása és ellenőrzése a cég minőségbiztosítási vezetőjének hatáskörébe tartozik. A fent leírtak által biztosítja majd a kiépítendő rendszer, hogy csak megfelelő minősítéssel rendelkező partnerek vegyenek részt a projektekben.

A partneryilvántartástól elvárando, hogy megkönnyítse a titkárság munkáját, vagyis alkalmas legyen:

- telefonszámok visszakeresésére
- telefonszámok automatikus felhívására
- automatikus levélcímzésekre
- faxok automatikus küldésére
- e-mail-re, stb.

4.2.3. Számlanyilvántartás

A számlák két helyen játszanak fontos szerepet: Egyrészt a gazdasági hivatal számára szolgáltatnak információkat, innen kérhető le globálisan tekintve a cég pillanatnyi likviditása, másrészt a vállalkozás és műszaki igazgatóság számára nyújtanak naprakész pénzügyi információt projektekre lebontva, segítve ezzel az egyes projektek esetenként nehezen áttekinthető folyamatainak nyomon követését.

Szükséges a számlák nyilvántartása a jelenlegi funkciók megtartásával, megszüntetve a jelenleg kimutatható tárolási anomáliákat, amit a kétszeres adattárolás illetve adatfelvitel vált ki.

4.2.4. Kimenő és bejövő információk nyilvántartása és projekthez kapcsolása

Egy modern információs rendszer biztosítja a levelek, faxok, telefonok, elektronikus küldemények digitális módon való tárolását, mely számos előnyt nyújt a hagyományos megoldásokkal szemben. Természetes igényként jelentkezik tehát az iroda automatizálás, mely gyűjtőfogalom kifejezi a hagyományos irodai munkák számítógéppel való elvégzetetését. Ezek közül a legfontosabb az automatikus iktatás és a több szempont szerinti gyors és hatékony visszakeresés. Nagy előnye, hogy eltűnik a sok papír és ez által sokkal áttekinthetőbbé válik az így tárolt információ. Az automatikus iktatás mellett fontos képessége az igényelt rendszernek, hogy minimális emberi beavatkozással szét lehessen válogatni és osztani a különböző beérkező küldeményeket.

4.3. További megvalósítandó feladatok

4.3.1. Számítógépes konferenciázó rendszer

Mivel a cég több földrajzi elhelyezkedésben egymástól távol eső irodából, illetve fix- és mobil telephelyből áll, az ezek közötti gyors és hatékony információáramlásról is gondoskodni kell. A vállalat jelenlegi adatforgalma ismeretében elmondható, hogy ennek megoldása internettel illetve intranettel képzelhető el.

Ez a megoldás kevésbé költségigényes és az adatátvitel sebességét tekintve sokkal jobb mutatókkal rendelkezik, mint a fax, de még rajzfájlok, színes képek, hangok küldését is megengedi, valamint lehetőség nyílik videó konferenciák szervezésére is.

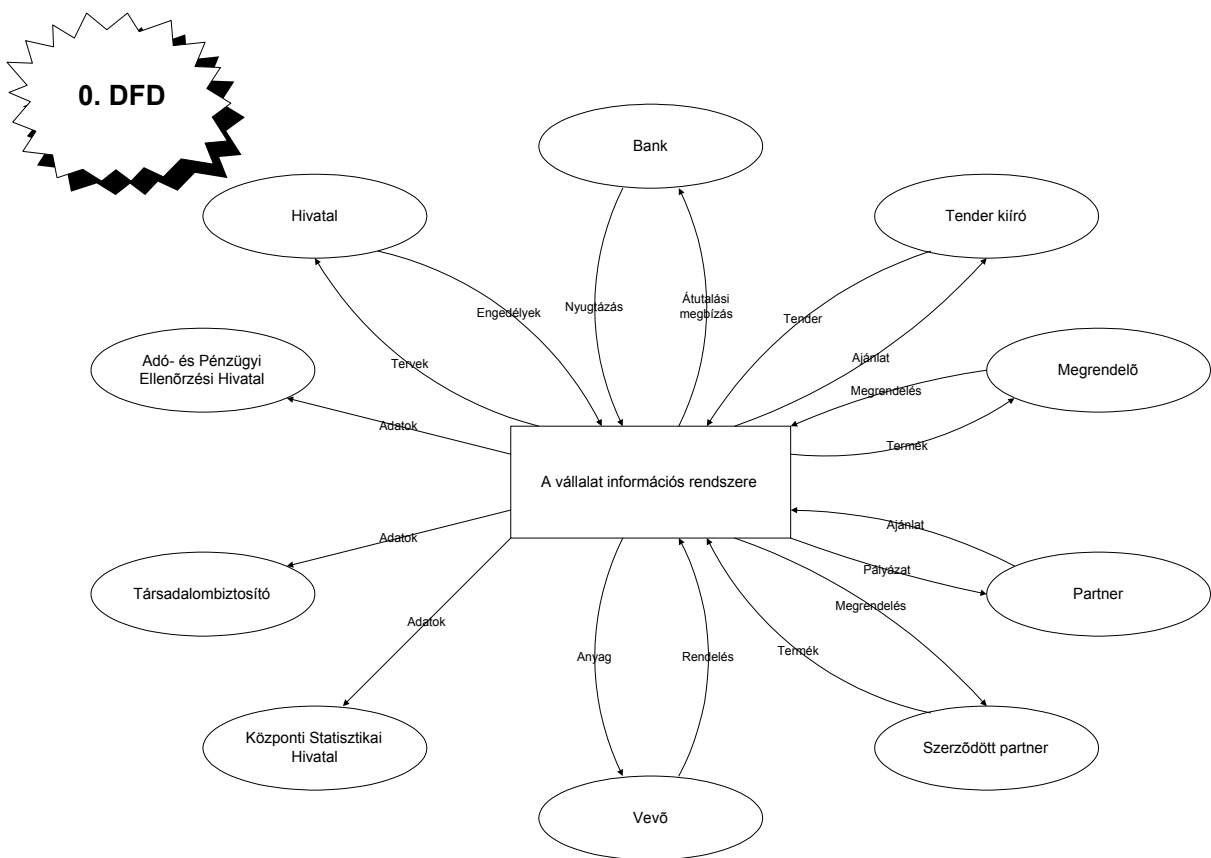
4.3.2. Az átállás folyamatával kapcsolatos igények

A jelenleg működő rendszerből a jól használható alrendszerek változatlanul hagyása, integrálása fájdalom mentesebbé teheti az átállást. A programrendszernek egységes felhasználói interfészt kell biztosítania, mely az információkat egységesen kezeli, és lehetővé teszi a csoportmunkát.

Megoldandó feladat még a régi rendszerről való problémamentes és biztonságos átállás, mindenre kiterjedő tesztelés, valamint a munkatársak betanítása.

ilyen az *Adó- és Pénzügyi Ellenőrzési Hivatal*, a *Társadalombiztosító*, stb. Az ezekhez kapcsolódó egyirányú adatok a rendszer által generált jelentések vagy bemeneti információk. A következő olyan csoport, ami nincs benne a rendszerben, azok az egyedek alkotják, amelyekben új információ keletkezik. Egy információs rendszer ugyanis nem képes önálló értéktételeket mondani és általában nem képes bonyolultabb döntésekre. Egyszerűen csak feldolgozza, tárolja és csoportosítja a rendszerbe került adatokat. Például a *Tender kiíró* új információt hoz létre azzal, hogy a tender adatai bekerülnek a rendszerbe. Ugyanígy új információ születik a *Beszállítóknál* az építkezési anyaggal kapcsolatos ajánlat beérkezésével.

5.2. Kontextus ábra (Nulladik szintű adatfolyam diagram)



5.2. ábra

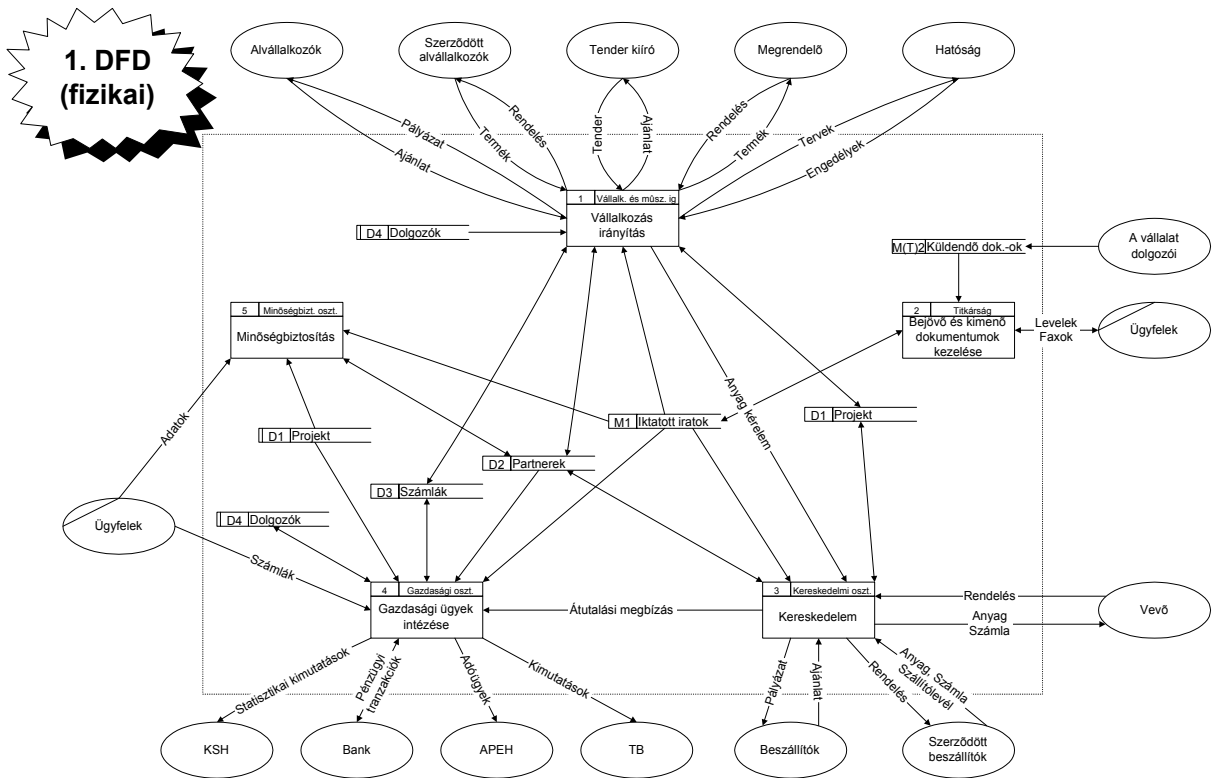
A rendszer körvonalának meghatározása után a következő lépés a kontextus ábra (5.2. ábra) elkészítése, amit a rendszer 0. szintű adatfolyam ábrájának (0. szintű DFD) is nevezhetünk. A rendszer ezen nem bomlik szét, a hangsúly a rendszer kiterjedésén és környezetének ábrázolásán van. A kontextus ábra tehát a belső felépítést nem taglalja.

5.3. Fizikai adatfolyam modell

A következő lépés az 1. szintű (Top level) DFD elkészítése a dokumentum áramlási ábra alapján (5.3. ábra). Ez annyit jelent, hogy a dokumentum áramlási ábrát átalakítjuk, feltüntetjük a szükséges fő folyamatokat és felvesszük a rendszerben használt globális, vagyis több folyamat által is igénybe vett adattárakat. A rendszer határán kívül eső egyedek változatlanok maradnak. A cél az volt, hogy a jelenlegi fizikai folyamatokat modellezzük, az összes hiányossággal, felesleges ismétlődéssel és hibával együtt. Mivel a valós helyzet túlságosan rendezetlen és zavaros volt, kénytelenek voltunk idealizálni, ezáltal részben elmozdultunk az igényelt rendszer irányába.

Már ezen az ábrán is nagyon szépen elkülönülnek vállalat működésének főbb folyamatai, mint például a *Vállalkozás irányítás* a Vállalkozási és Műszaki Igazgatóságon vagy a *Gazdasági ügyek intézése* a Gazdasági Osztályon, és az azok közötti főbb adatfolyamok. Ezek a folyamatok még nagyon bonyolultak, ezért a későbbiekben további részfolyamatokra lesznek bontva. Annak ellenére, hogy a vállalat adatainak tárolása nagyrészt már számítógépesített, megtalálhatóak még a hagyományos, manuális adatrögzítési formák is (pl. *Iktatott iratok*). Az ábrából kitűnik az is, hogy a jelenlegi információs rendszer nem támogatja a cég által elnyert ISO 9001-es minősítés által megkövetelt előírásokat. Például a *Partnerek* adattárba jelenleg szinte bárki vihet fel bármilyen új partnert, nem ellenőrizve azt, hogy az adott partner megfelel-e a minőségbiztosítási előírásoknak. Ezt a hiányosságot majd a logikalizálás során oldjuk fel.

A fő folyamatok közül kiragadjuk a *Vállalkozás irányítást*, melyet részletesen ismertetünk. A *Tender kiíróval* való kapcsolat egyrészt a beérkező tender formájában, másrészt az elkészített ajánlatban nyilvánul meg, melyet az *Alvállalkozók* pályáztatásával, azok ajánlatait felhasználva állít össze. Ha már elnyert tenderről van szó, akkor a *Tender kiíró Megrendelővé* minősül át, az *Alvállalkozókat* pedig szerződteti. A tender alapadatai bekerülnek a *Projekt* adattárba, az esetleges új alvállalkozók pedig a *Partnerekbe*. A saját munkaerő adataihoz a *Dolgozók* adattáron keresztül fér hozzá. A munka során felmerülő anyag kérelmet közvetlenül továbbítja a *Kereskedelem* folyamatnak.



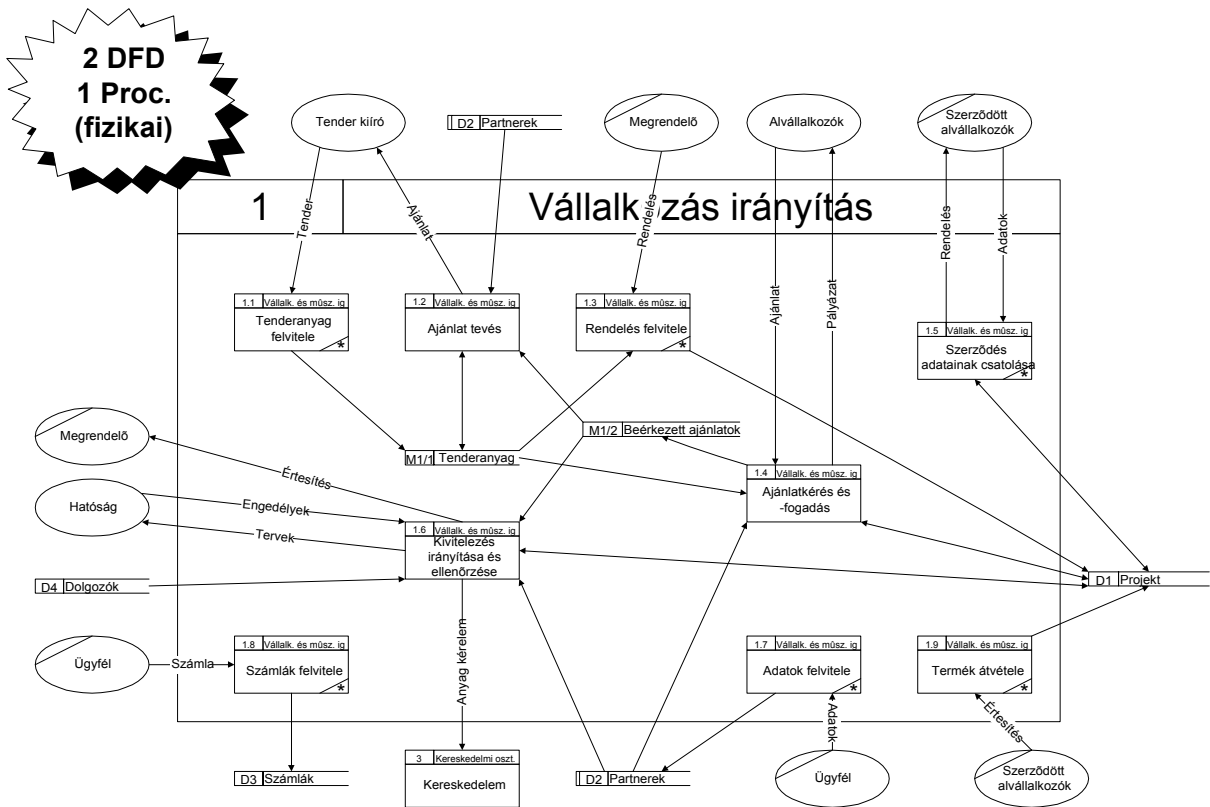
5.3. ábra

A következőkben az alsóbb szintű adatfolyam diagramok kidolgozása következik. Itt a már meglévő, magasabb szintű folyamatok belsejének megismerése a cél. Az átláthatóság kedvéért minden egyes magasabb szintű folyamat felbontásának eredménye külön ábrára került.

Az alábbiakban a „Vállalkozás irányítás” folyamat lebontását követjük részletesen nyomon (5.4. ábra).

Ezen a 2.szintű DFD-n a felsőbb szintű folyamat összes bemeneti és kimeneti adatfolyama szerepel. Itt megjelennek lokális adattárak is, melyeket csak ez az alfolyamat használ (pl. *Beérkezett ajánlatok*). Ezen a szinten vannak már olyan folyamatok is, melyek továbbbontására az adatfolyam modellezésben nincs szükség, ugyanis ezek már elemi folyamatokat írnak le (pl. *Számlák felvitele*). Ezek részletes leírása *Jackson módszerrel* történhet.

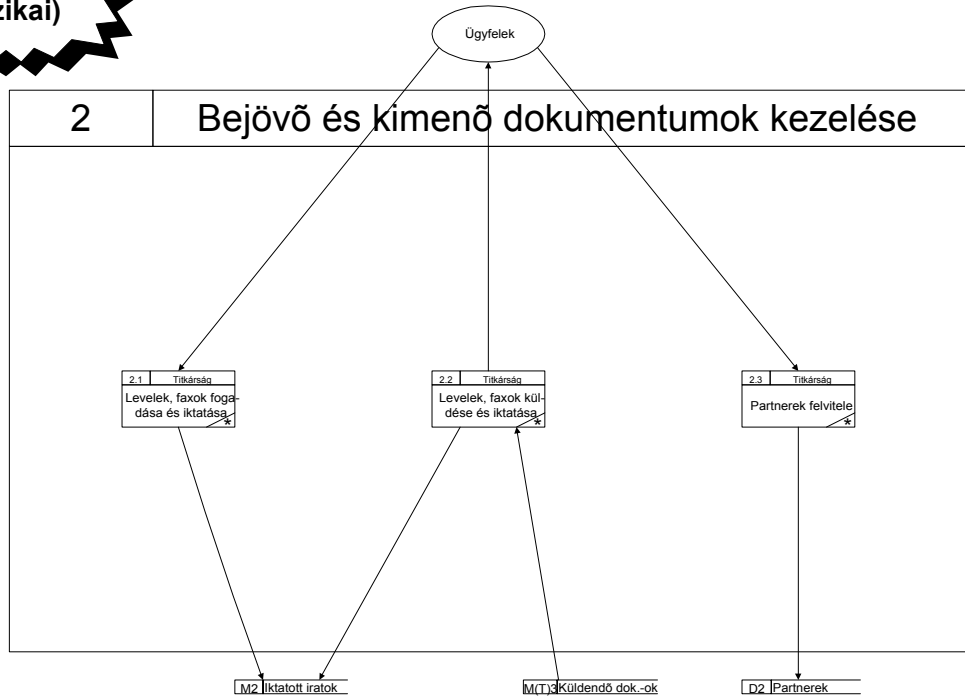
Most nézzük meg részletesen mi történik a *Tender kiírótól* érkező pályázattal. A beérkezett tenderanyagot a *Tenderanyag felvitele* folyamat a *Tenderanyag* lokális manuális adattárba írja be. Az *Ajánlat tevés* folyamat a *Tenderanyag*, *Beérkezett ajánlatok* lokális- és a *Partnerek* globális adattárakból vett információ alapján ajánlatot generál. A pályázat elnyerése után a *Tenderanyag* adatai a *Rendelés felvitele* folyamat által átkerülnek a *Projekt* adattárba.



5.4. ábra

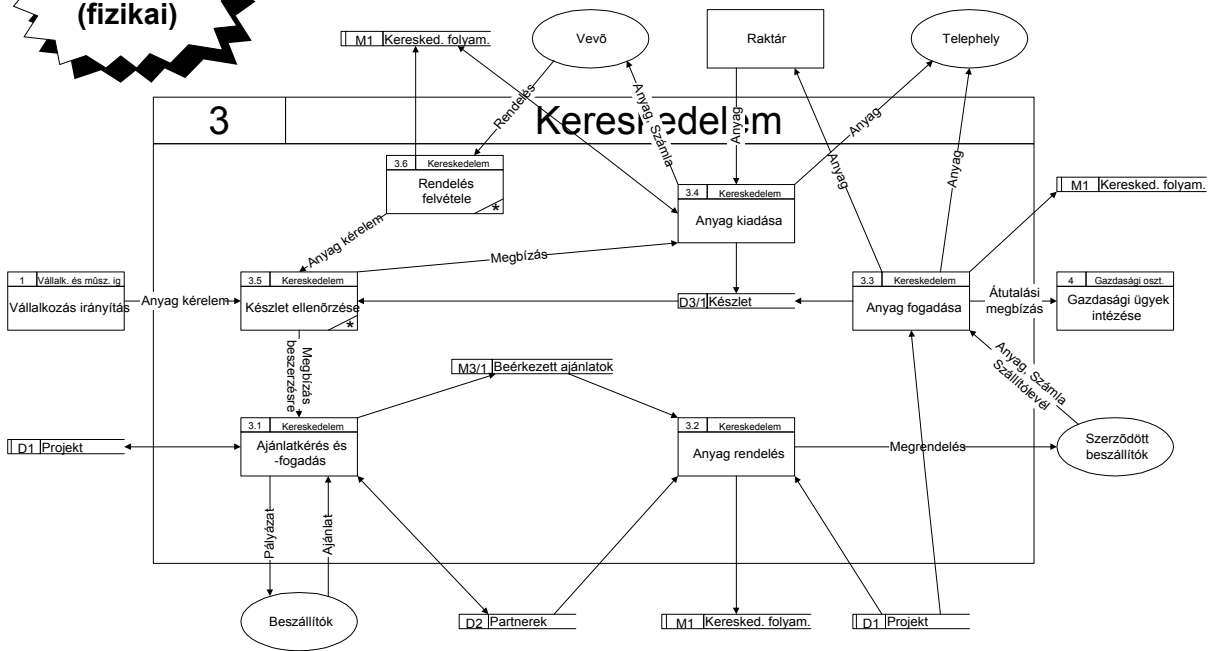
A többi fő folyamatot nem részletezzük.

**2 DFD
2 Proc.
(fizikai)**



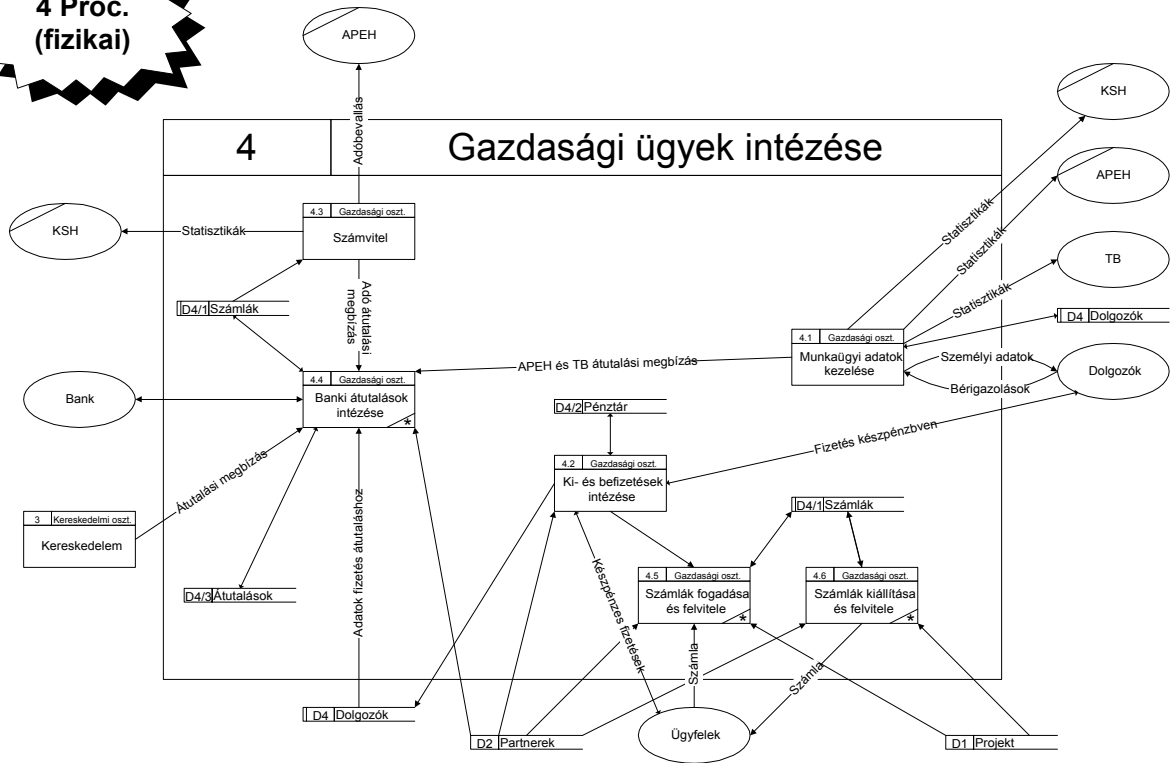
5.5. ábra

**2 DFD
3 Proc.
(fizikai)**



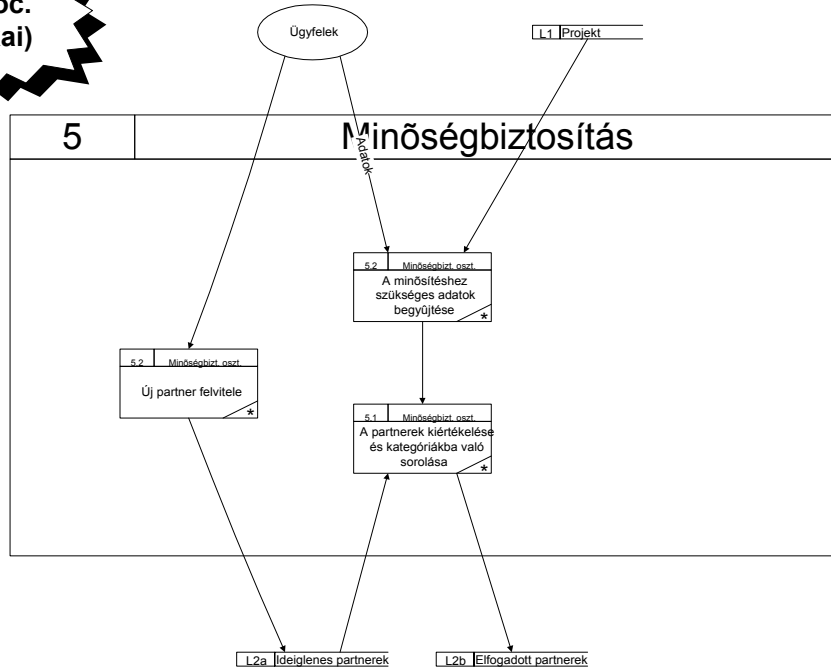
5.6. ábra

**2 DFD
4 Proc.
(fizikai)**



5.7. ábra

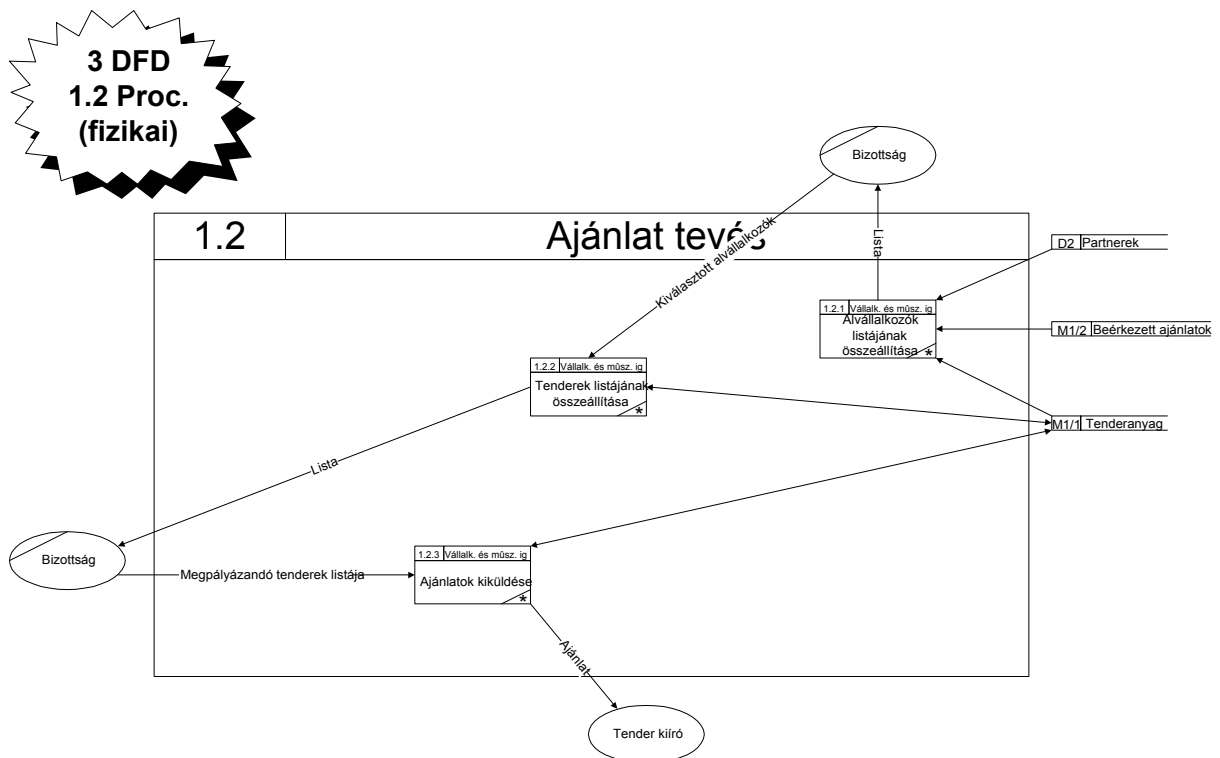
**2 DFD
5 Proc.
(fizikai)**



5.8. ábra

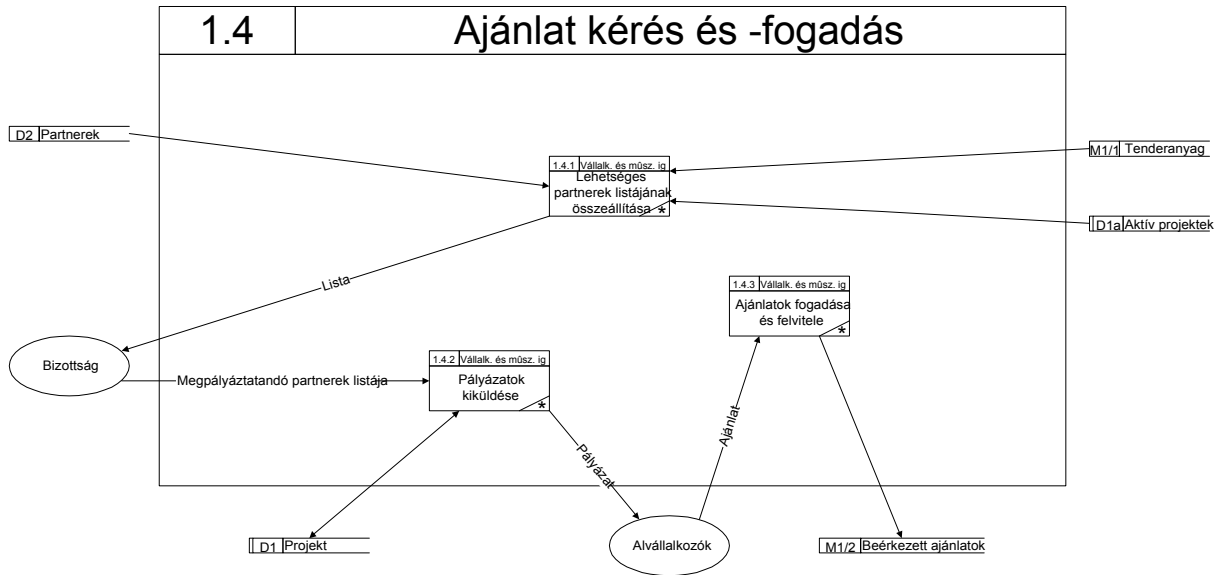
Ha a második szintű DFD-k még tartalmaznak nem lezárt folyamatokat, akkor azokat tovább kell bontani. Három szintű felbontással a valós folyamatok nagy része modellezhető, negyedik szintű DFD-kre az esetek többségében nincs szükség.

Mivel egy folyamaton belül új információ nem keletkezhet, a *Beérkezett ajánlatokból az Alvállalkozók listájának összeállítása* után egy „Külső egyed” (*Bizottság*) kell, hogy döntsön a projektben résztvevő alvállalkozókról. Esetünkben a *Bizottság* nem más, mint a cég vezérkara. Hasonló mechanizmus játszódik le a megpályázandó tenderek kiválasztásánál is (5.9. ábra).



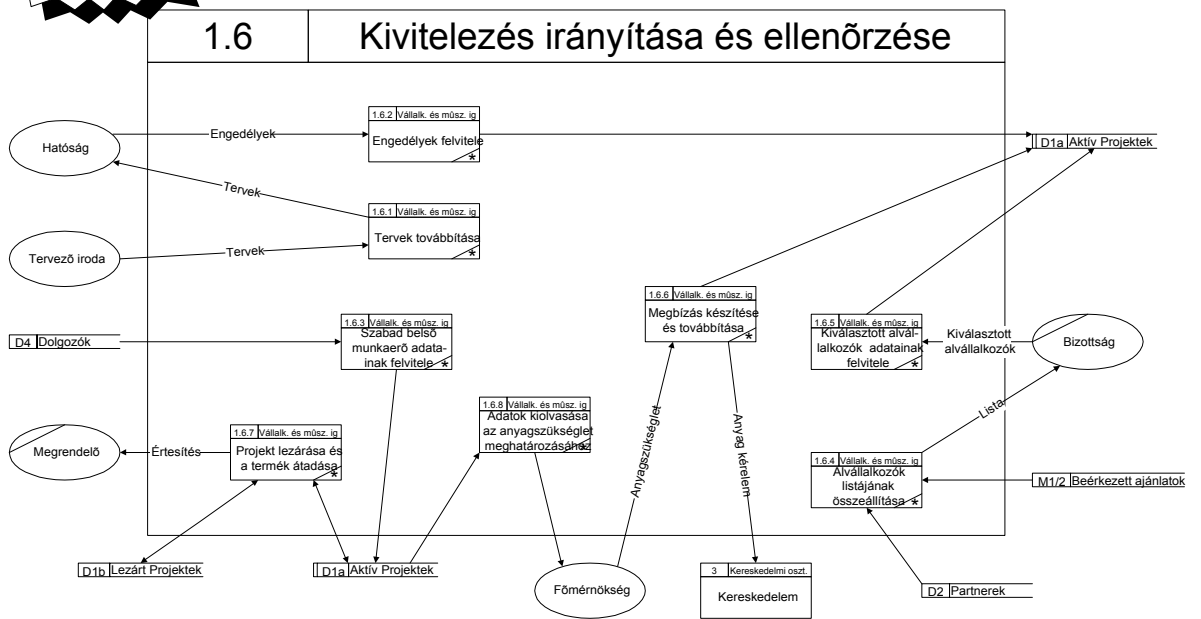
5.9. ábra

3 DFD
1.4 Proc.
(fizikai)



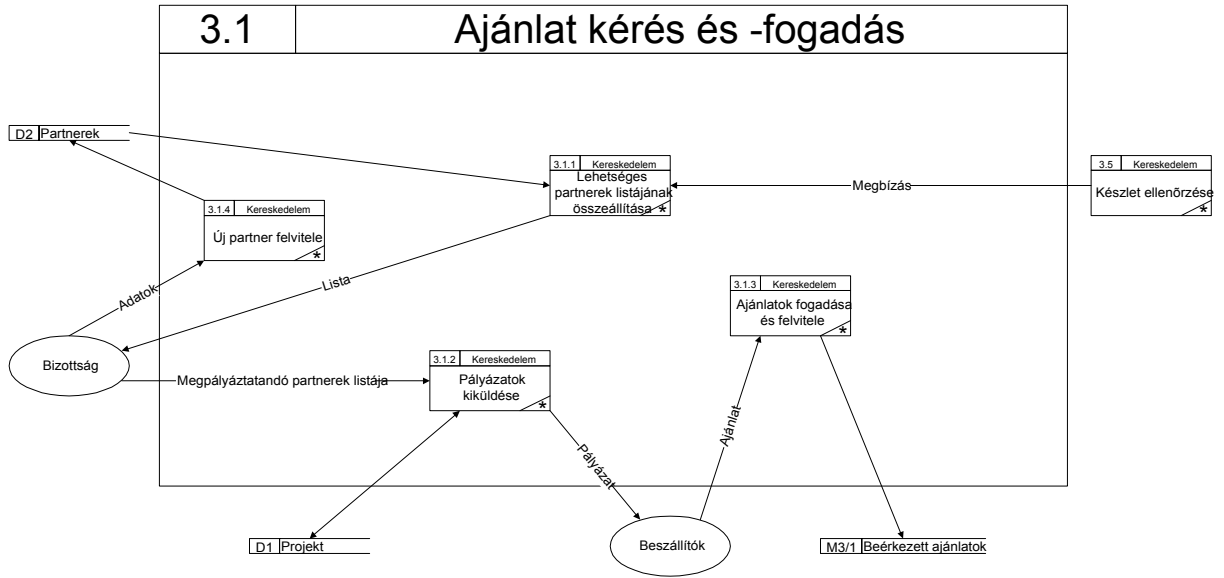
5.10. ábra

3 DFD
1.6 Proc.
(fizikai)



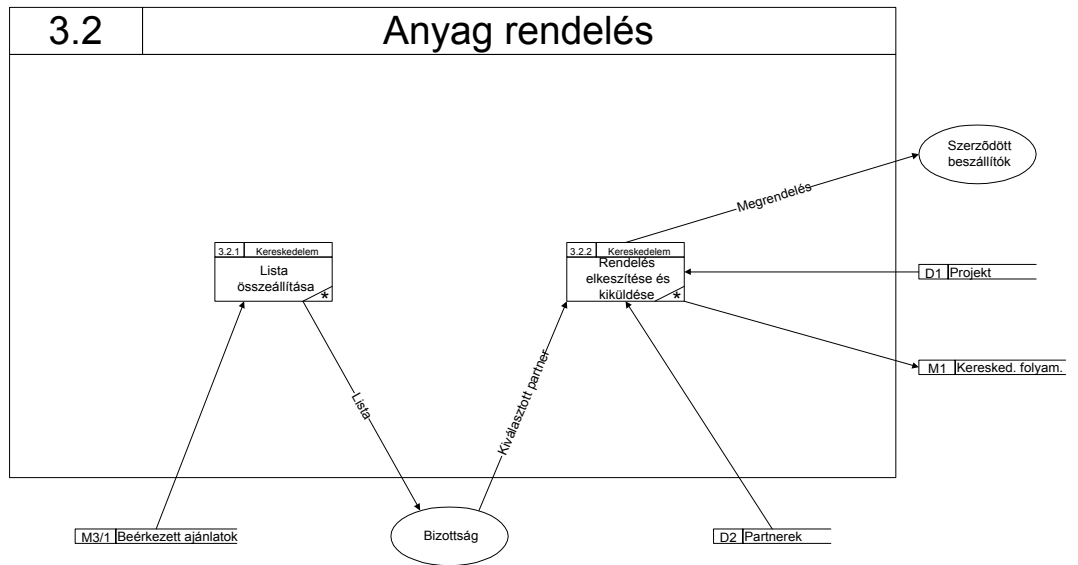
5.11. ábra

3 DFD
3.1 Proc.
(fizikai)



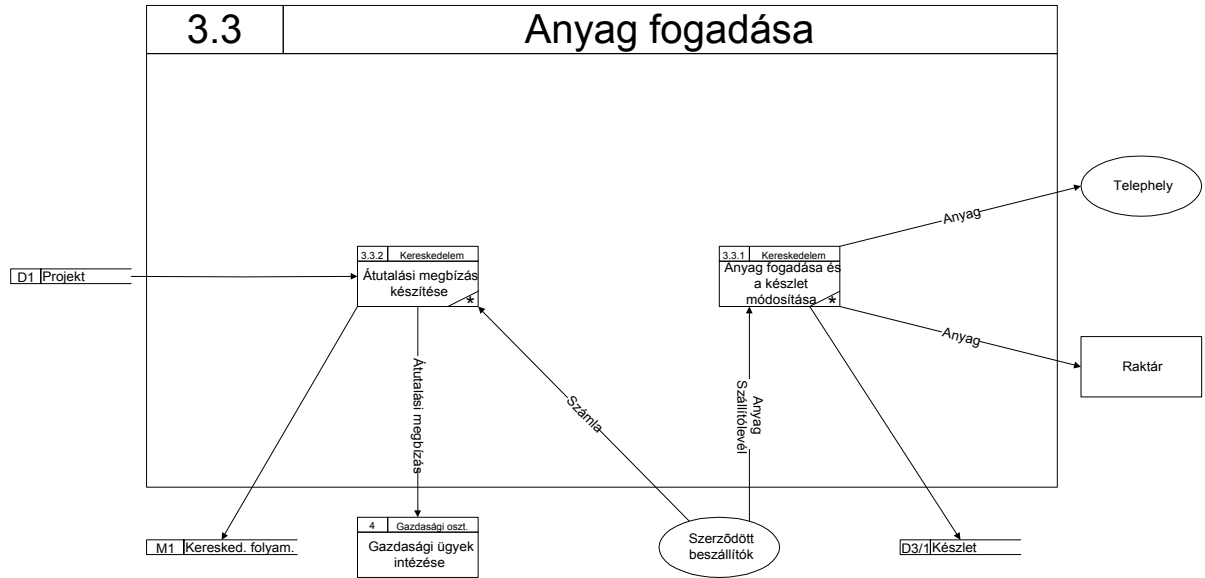
5.12. ábra

3 DFD
3.2 Proc.
(fizikai)



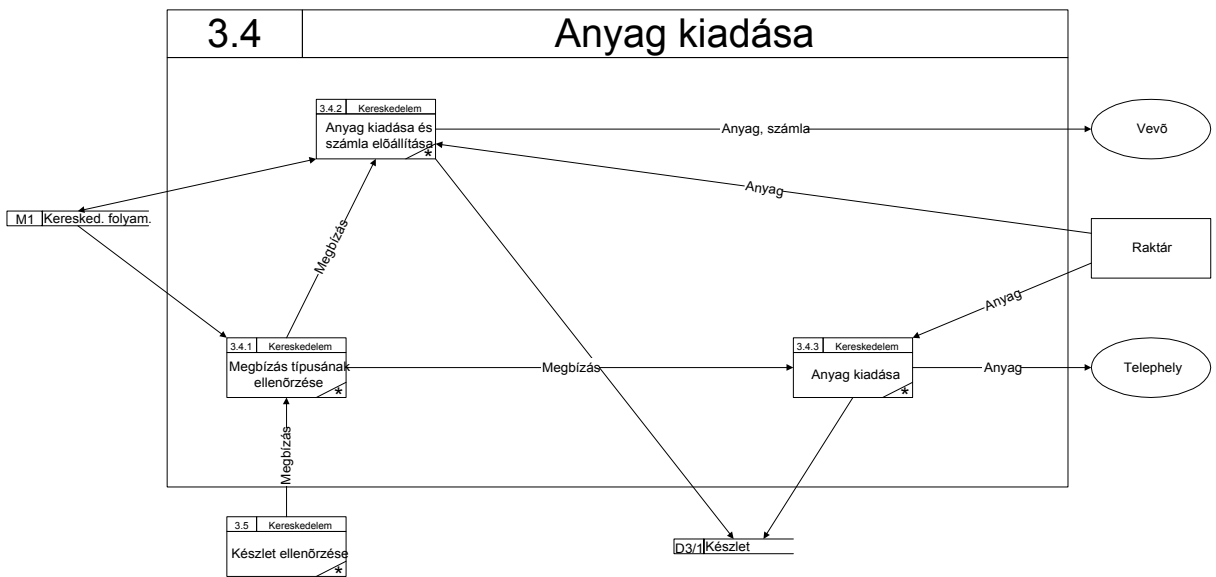
5.13. ábra

**3 DFD
3.3 Proc.
(fizikai)**



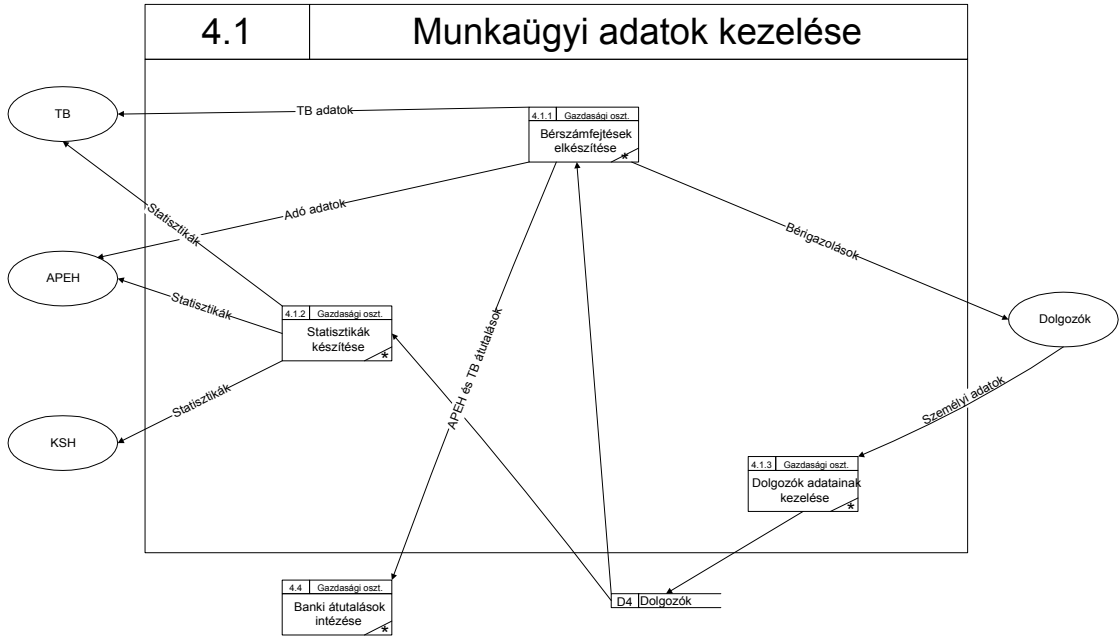
5.14. ábra

**3 DFD
3.4 Proc.
(fizikai)**



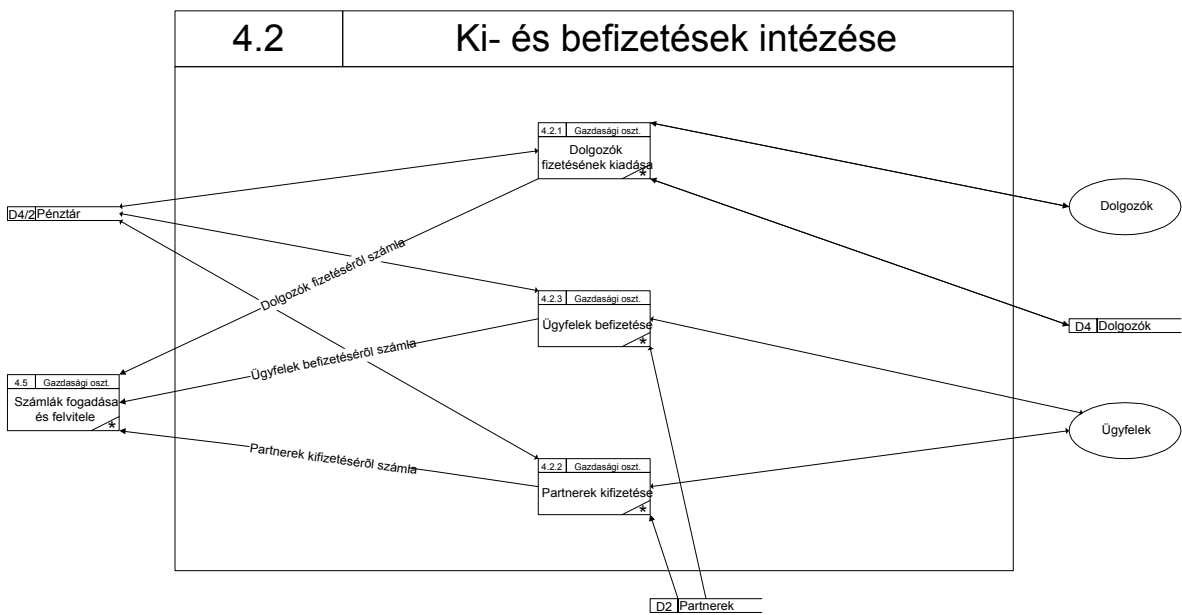
5.15. ábra

3 DFD
4.1 Proc.
(fizikai)

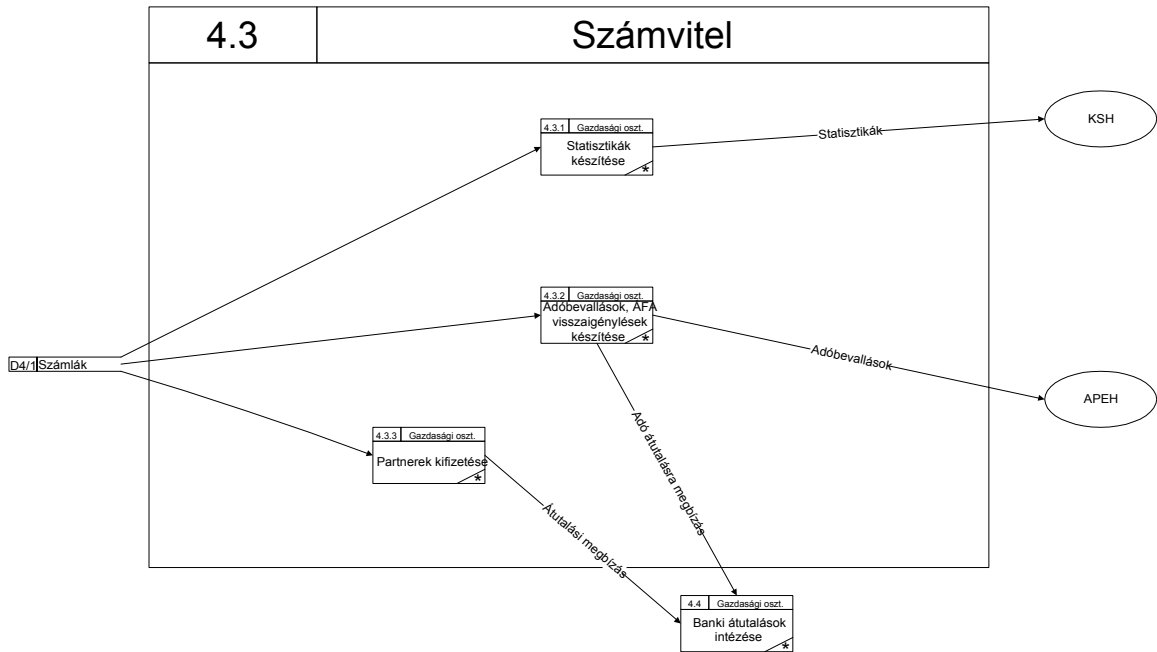


5.16. ábra

3 DFD
4.2 Proc.
(fizikai)



5.17. ábra



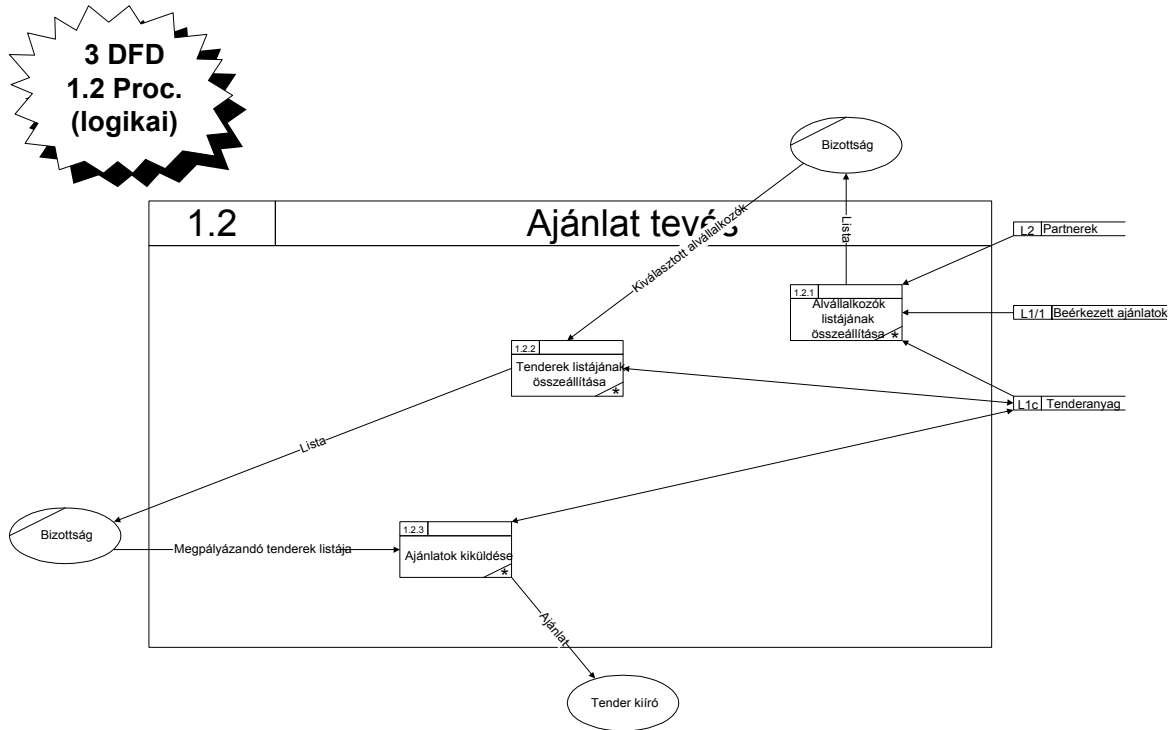
5.18. ábra

5.4. Logikai adatfolyam modell

A jelenlegi környezet folyamatainak fizikai vonatkozásait meg kellett szüntetni, az adattárolási kettősségeket fel kellett oldani, a folyamatokat pedig logikus szerkezetbe kellett rendezni. Itt történt meg a folyamatok összevetése az adatokkal, egy olyan megfeleltetést adva, amely kizárja, hogy a folyamatok által használt különböző adattárak ugyanazokra az adatokra vonatkozzanak. A cél az volt, hogy meghatározott szabályok alkalmazásával kiszűrjük a fizikai elemeket és a felesleges többszörözéseket a fizikai folyamatok modelljéből, kialakítva egy olyan logikai képet a működésről, amely valószínűleg az új rendszerben is érvényes lesz.

Az átalakítást a fizikai DFD-hierarchia legalsó szintjén kezdtük. Ennek az volt az oka, hogy ilyen módon egy alsó szintű diagram logikaivá alakításával előállítottuk a felette lévő szint egy folyamatát, tehát a legalsó szintek alakítása után már többé-kevésbé mechanikusan haladhattunk felfelé a hierarchiában. Úgy is lehet fogalmazni, hogy a felsőbb szinteket már nem átalakítottuk, hanem a már logikai első szintből állítottuk elő.

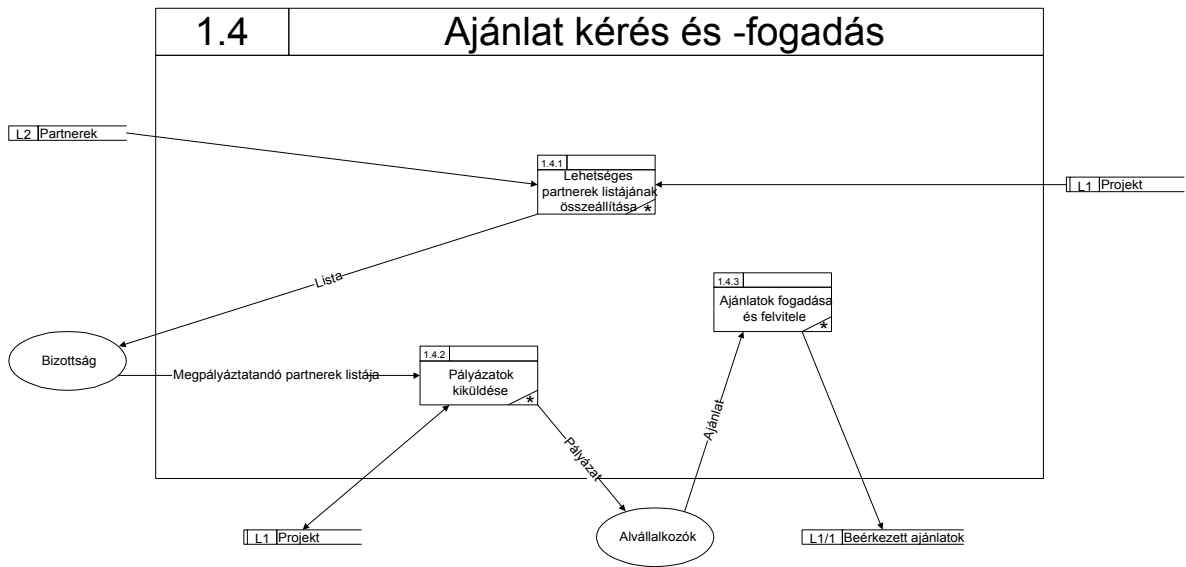
Mivel az *Ajánlat tevés* folyamata már egy nagyon kiforrott és jól bevált tevékenység a cégnél, a logikalizálás során nagyobb átalakításra nem szorult, csak a jelenlegi környezet fizikai vonatkozásait szüntettük meg (5.19. ábra).



5.19. ábra

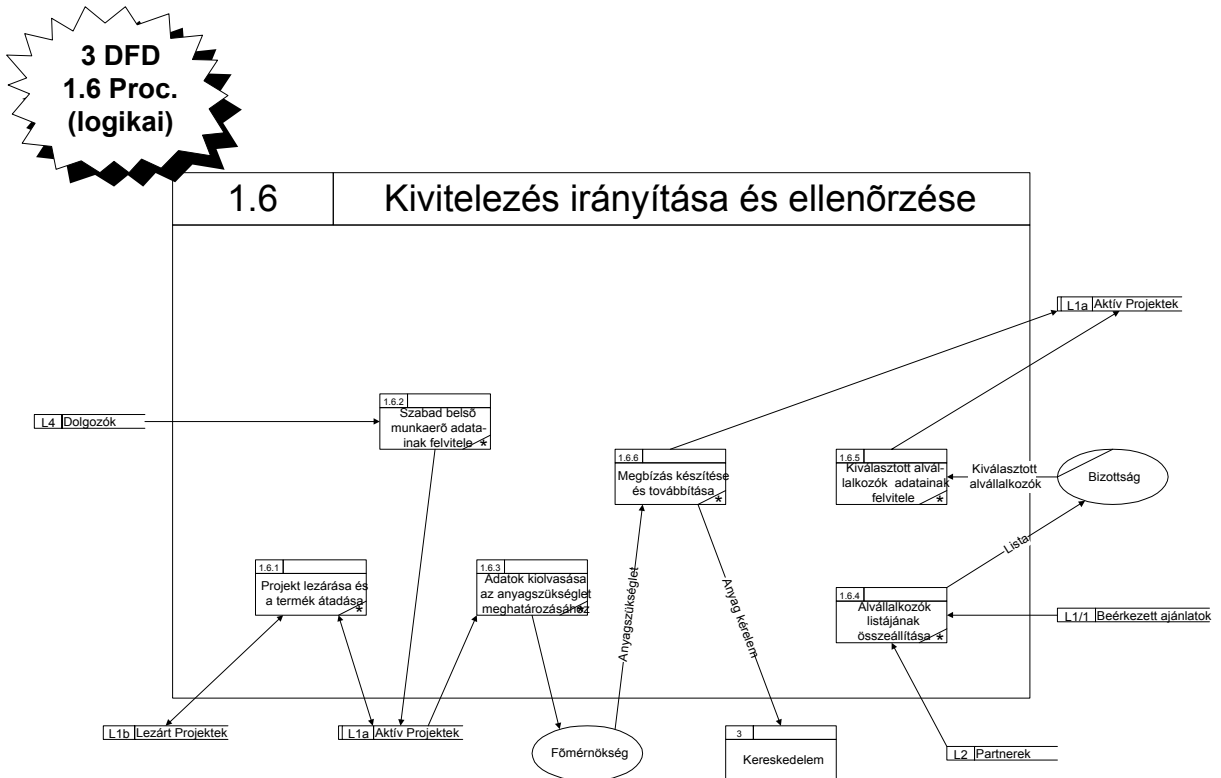
Az *Ajánlatkérés és -fogadás* folyamatánál már a *Tenderanyag* lokális manuális adattárát beolvasztottuk a *Projekt* immáron logikai adattárba. Ezt azért tehetjük meg, mert a *Tenderanyag* adattár csak az adott fizikai környezet miatt volt szükséges (5.20. ábra).

3 DFD
1.4 Proc.
(logikai)



5.20. ábra

A kivitelezés irányítása és ellenőrzése folyamat ábráján megszüntettük a *Tervek továbbítása* és az *Engedélyek felvitele* alfolyamatokat, mert kizárólag fizikai vonatkozásaik vannak (5.21. ábra).

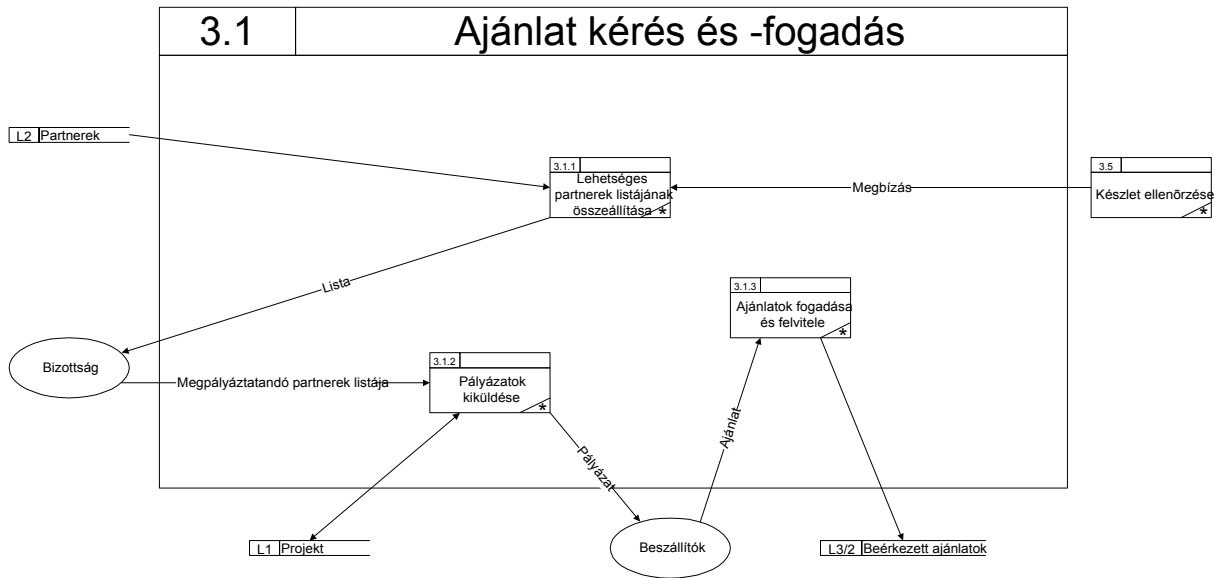


5.21. ábra

Mivel az *Új partner felvitele* folyamat több alfolyamatban is szerepel, felsőbb szintre helyeztük át, ezáltal megszüntettük az ismétlődést. (5.22., 5.30., 5.33. ábrák).

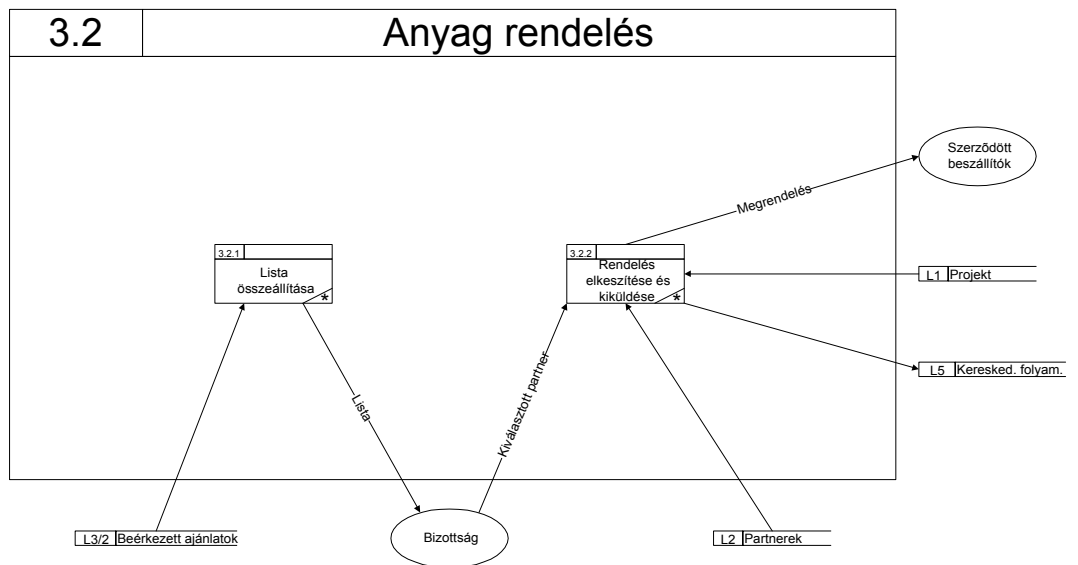
A többi harmadik szintű alfolyamatot nem részletezzük.

3 DFD
3.1 Proc.
(logikai)



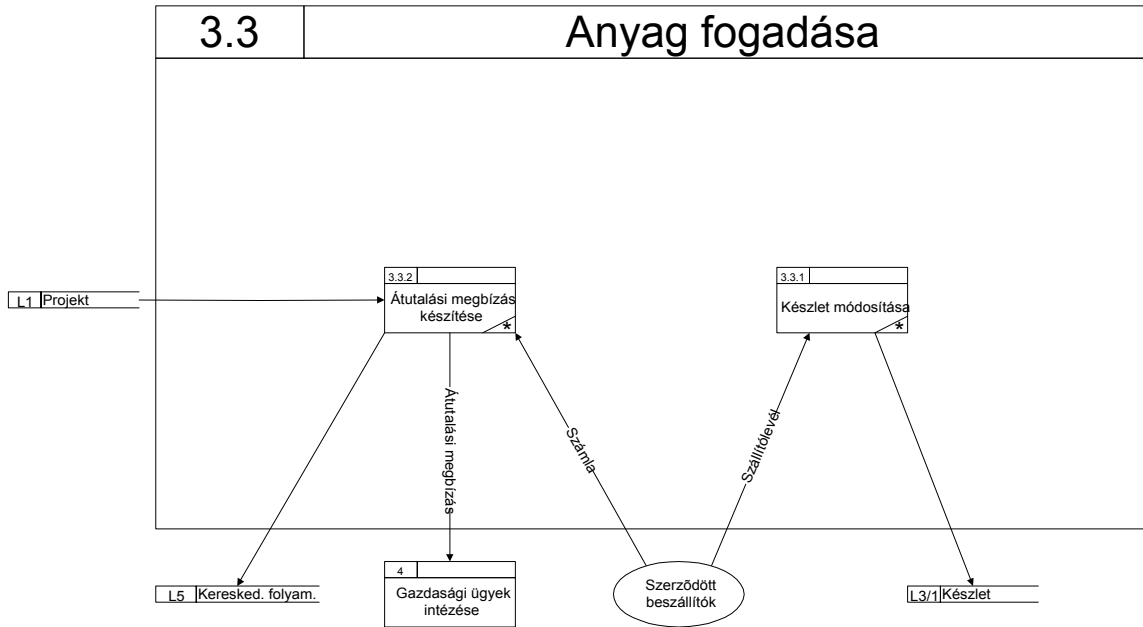
5.22. ábra

3 DFD
3.2 Proc.
(logikai)



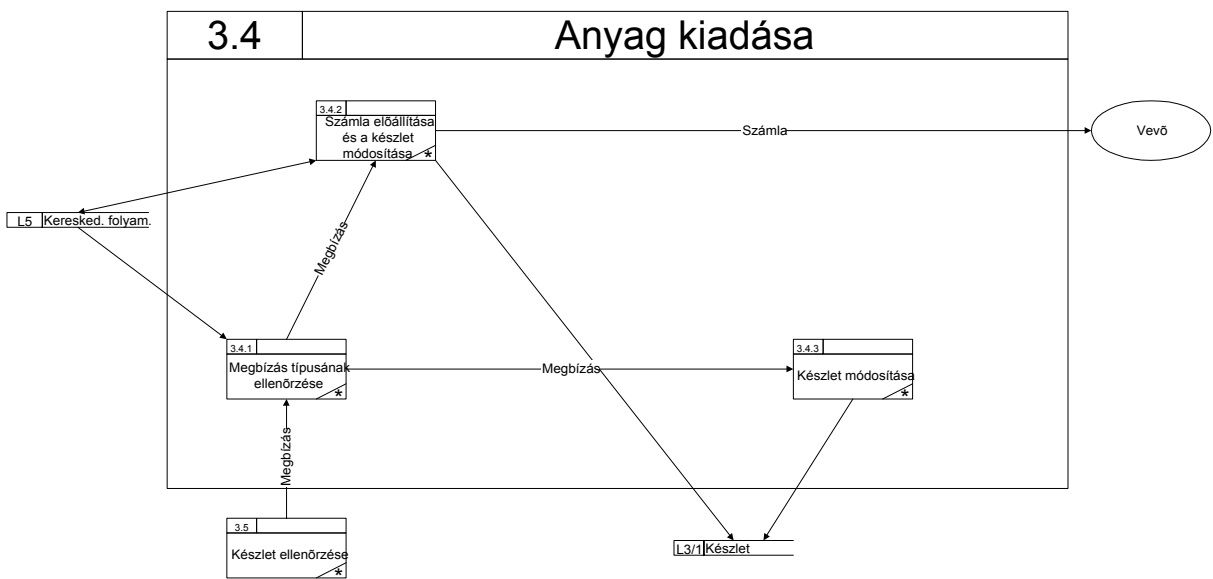
5.23. ábra

3 DFD
3.3 Proc.
(logikai)



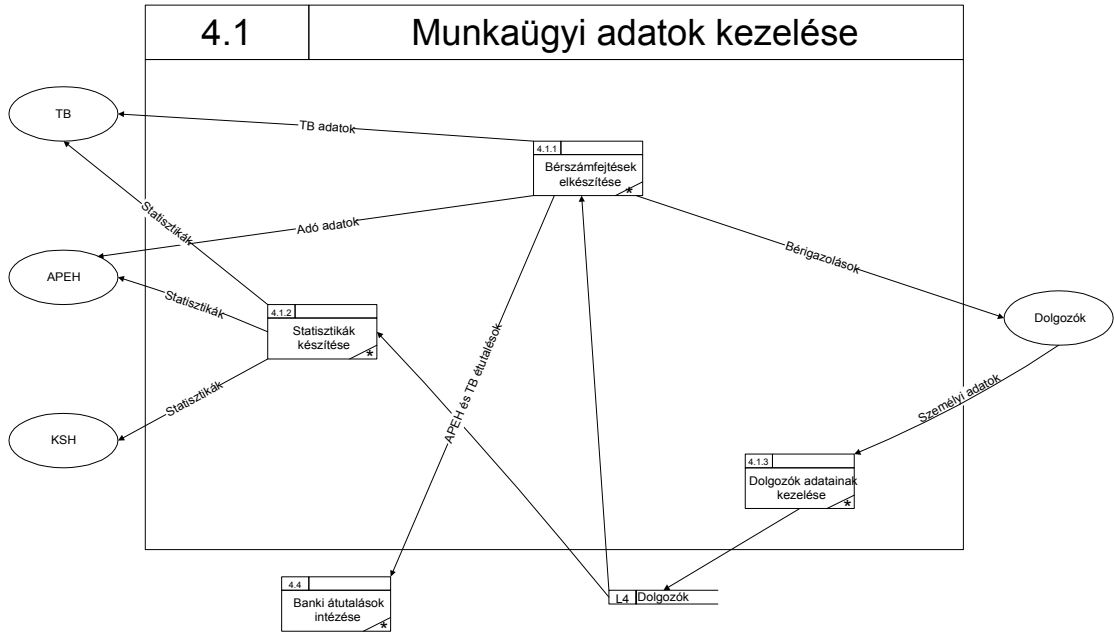
5.24. ábra

3 DFD
3.4 Proc.
(logikai)



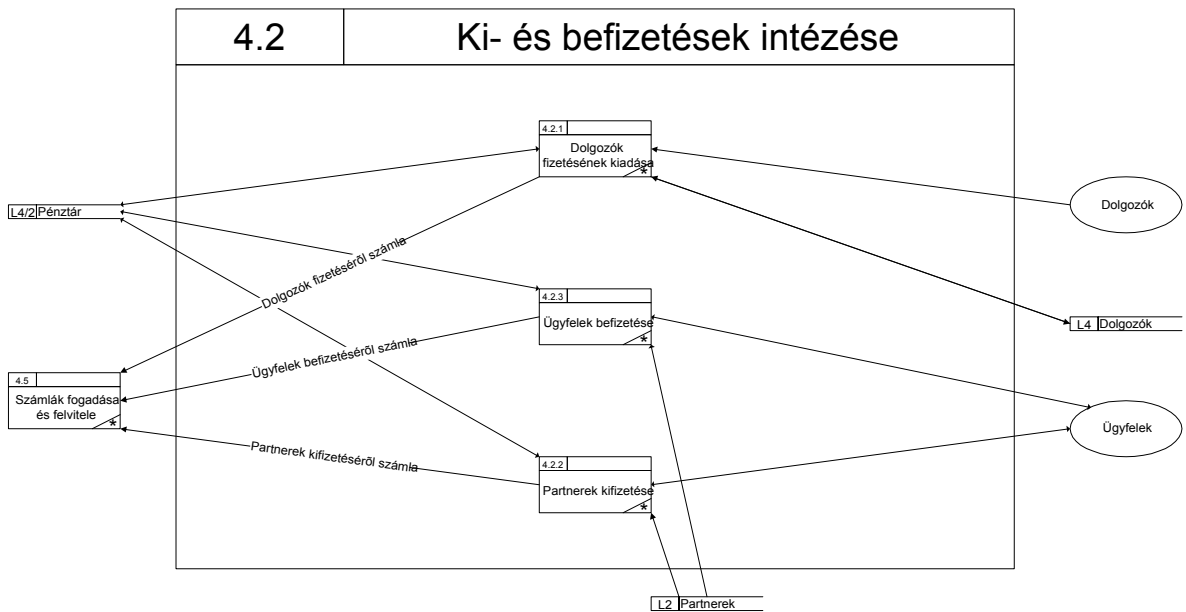
5.25. ábra

3 DFD
4.1 Proc.
(logikai)



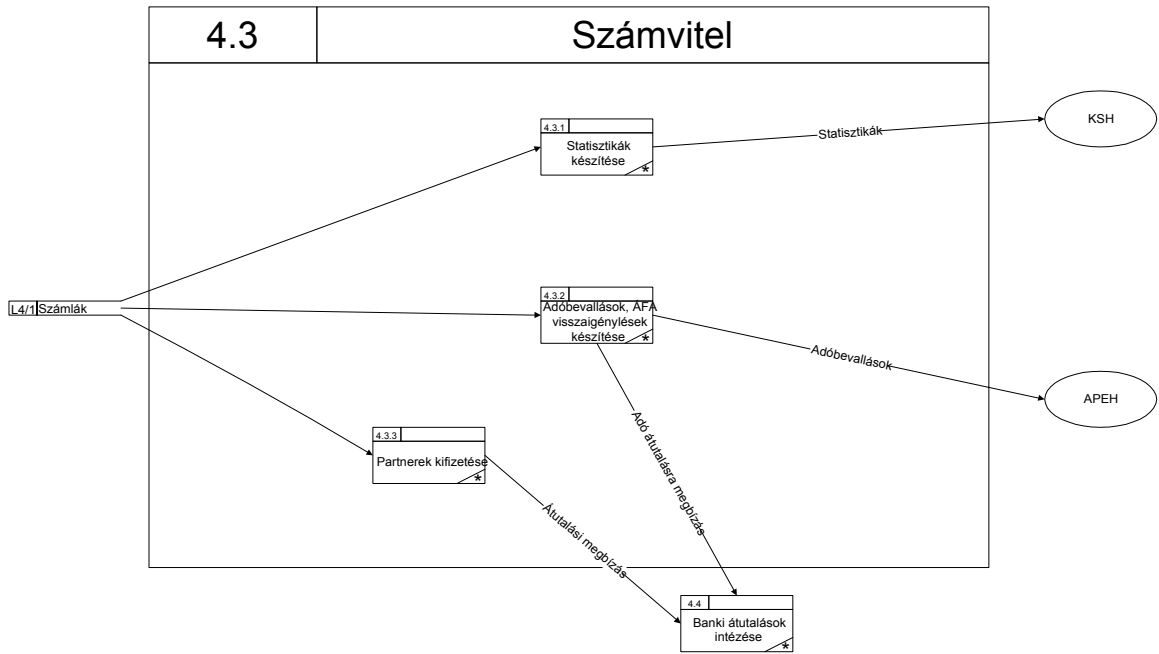
5.26. ábra

3 DFD
4.2 Proc.
(logikai)



5.27. ábra

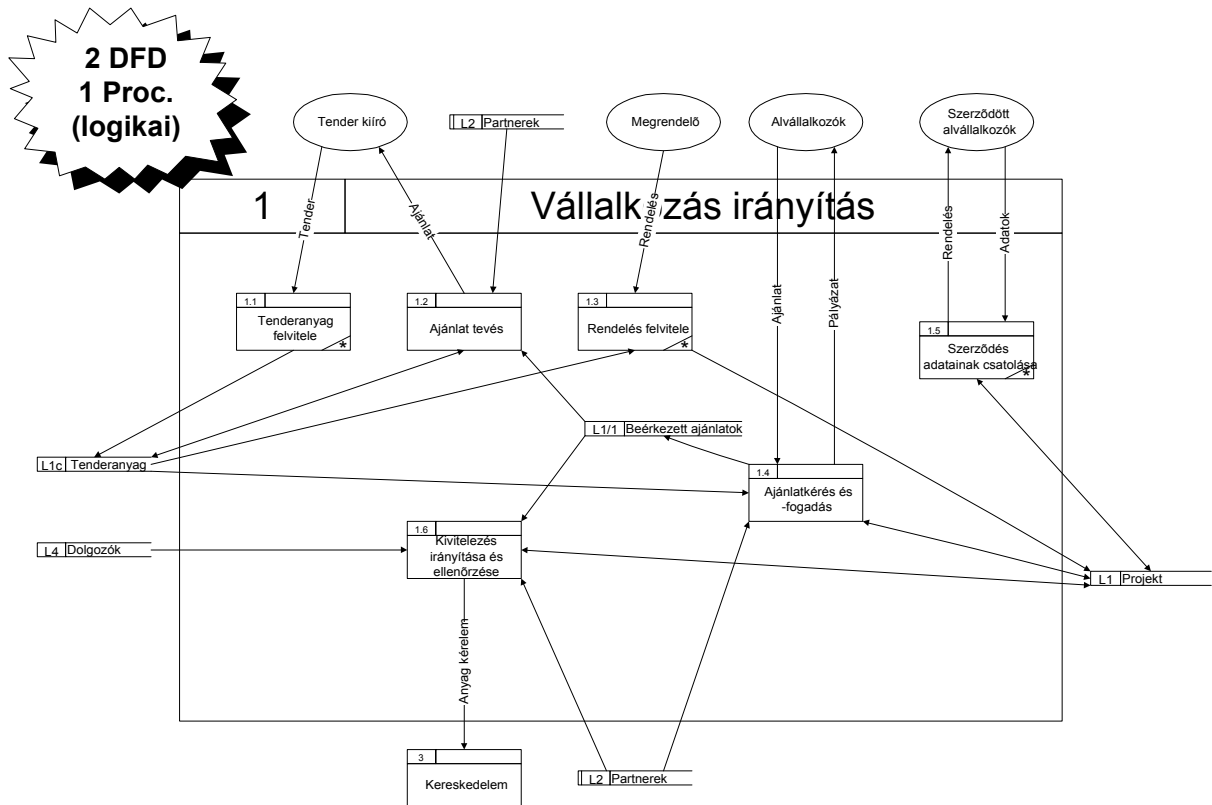
3 DFD
4.3 Proc.
(logikai)



5.28. ábra

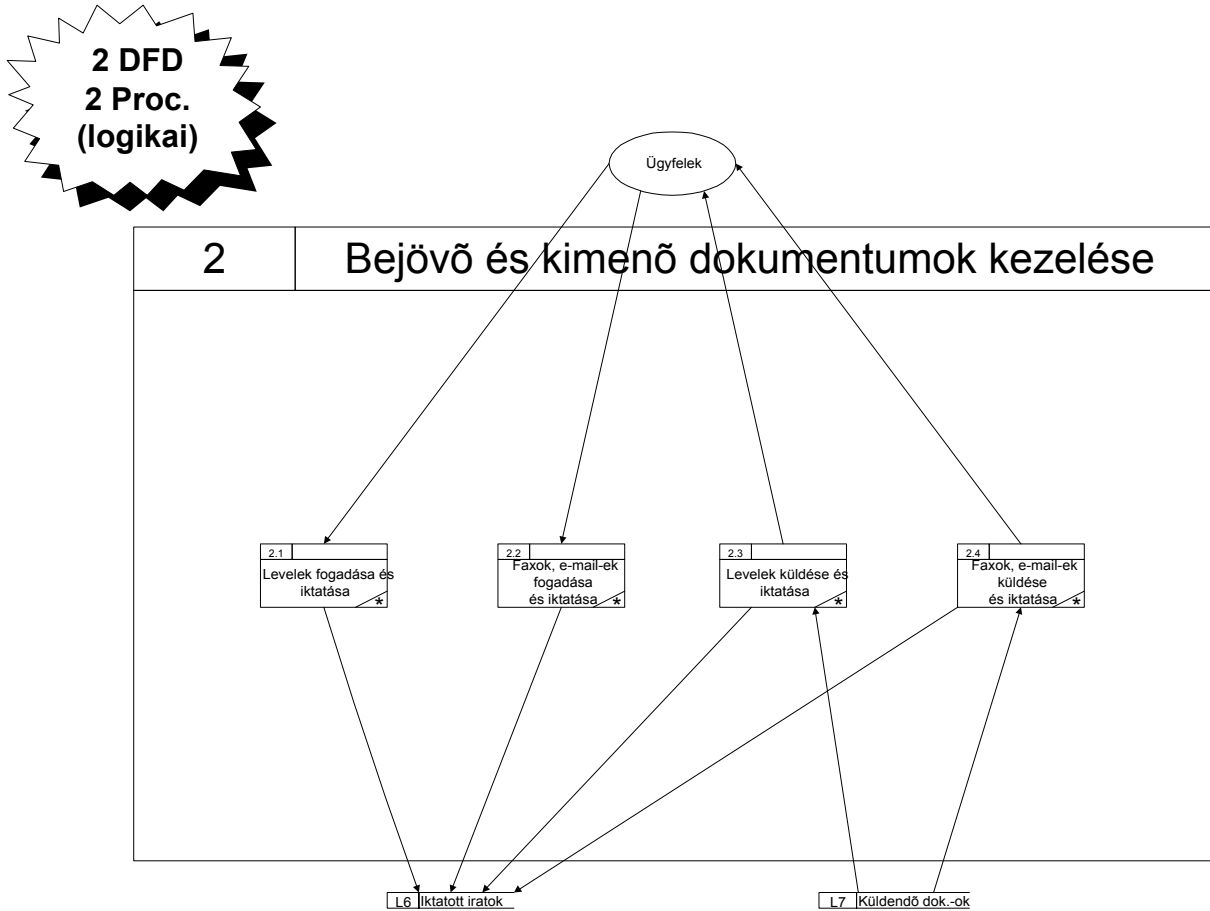
A 3. szintű DFD-k logikalizálás során bekövetkezett módosításait értelemszerűen alkalmazzuk a 2. szinten levő diagramokra. A fizikai vonatkozásokat mint korábban, itt is megszüntettük

A *Számlák felvitele* folyamatot töröltük a kettős adatfelvitel kiküszöbölése végett, a számlafelvétel így már csak a *Gazdasági ügyek intézése* folyamat része lett (5.29. ábra).



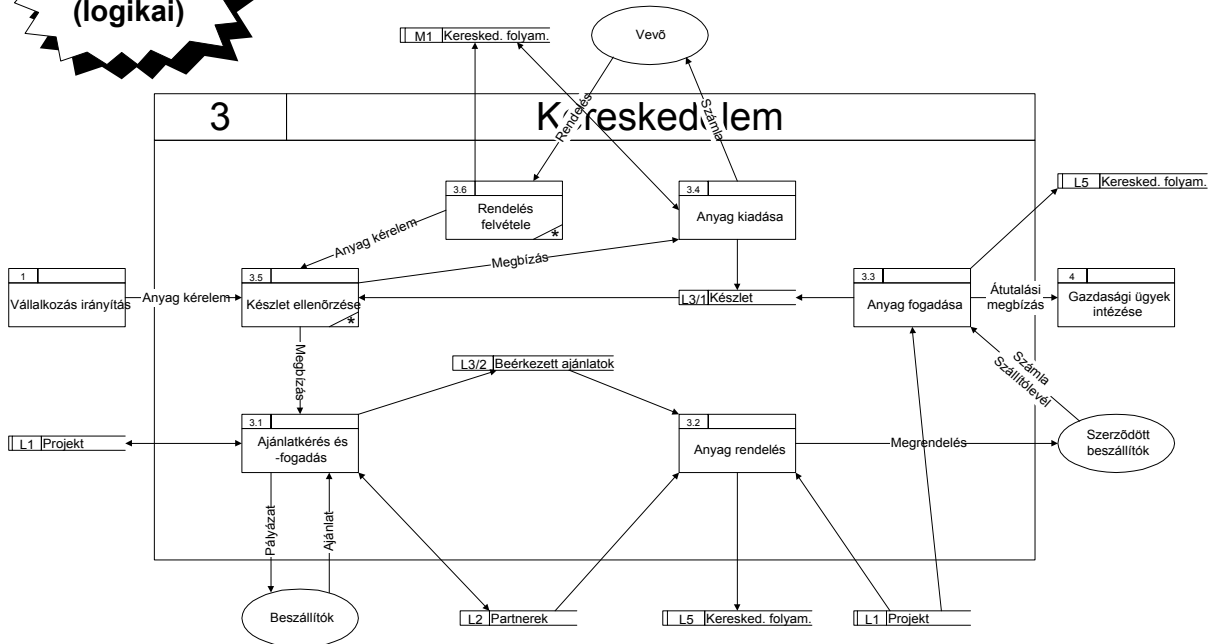
5.29. ábra

A *Bejövő és kimenő dokumentumok kezelése* folyamatnál a *Partnerek felvitele* folyamat felsőbb szintre helyezésén kívül különválasztottuk a hagyományos illetve elektronikus küldeményeket feldolgozó folyamatokat (5.30. ábra).



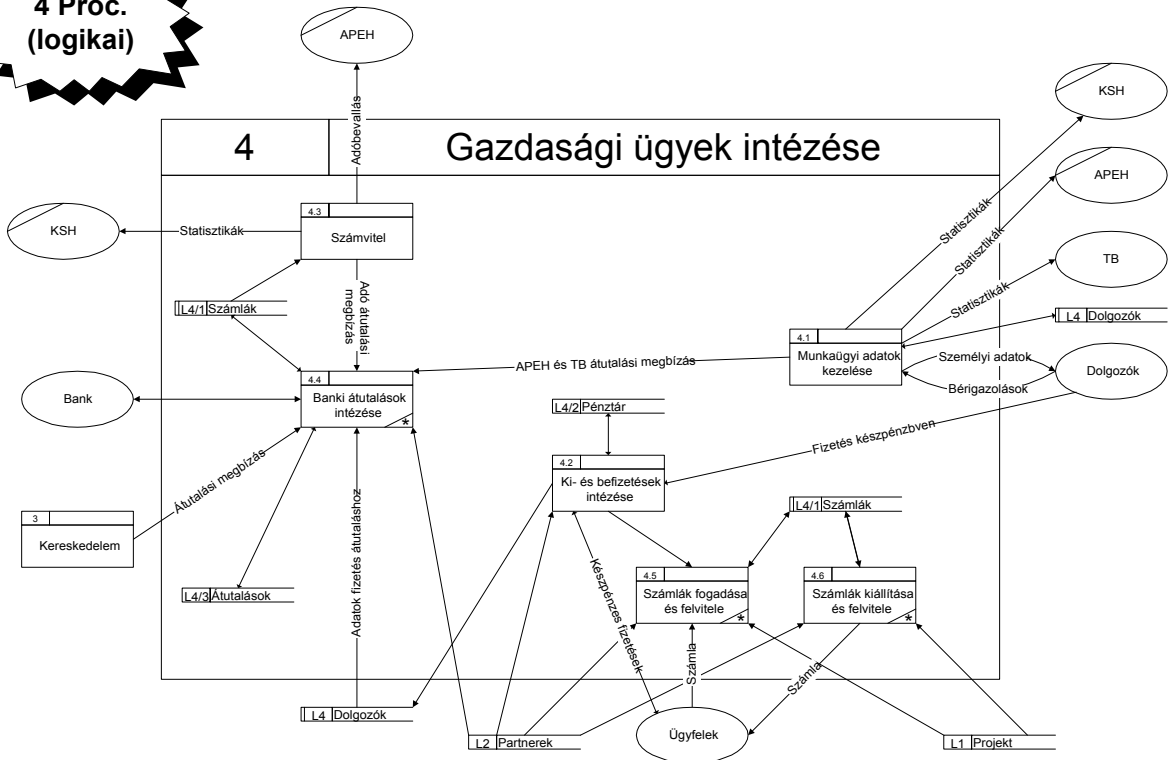
5.30. ábra

**2 DFD
3 Proc.
(logikai)**



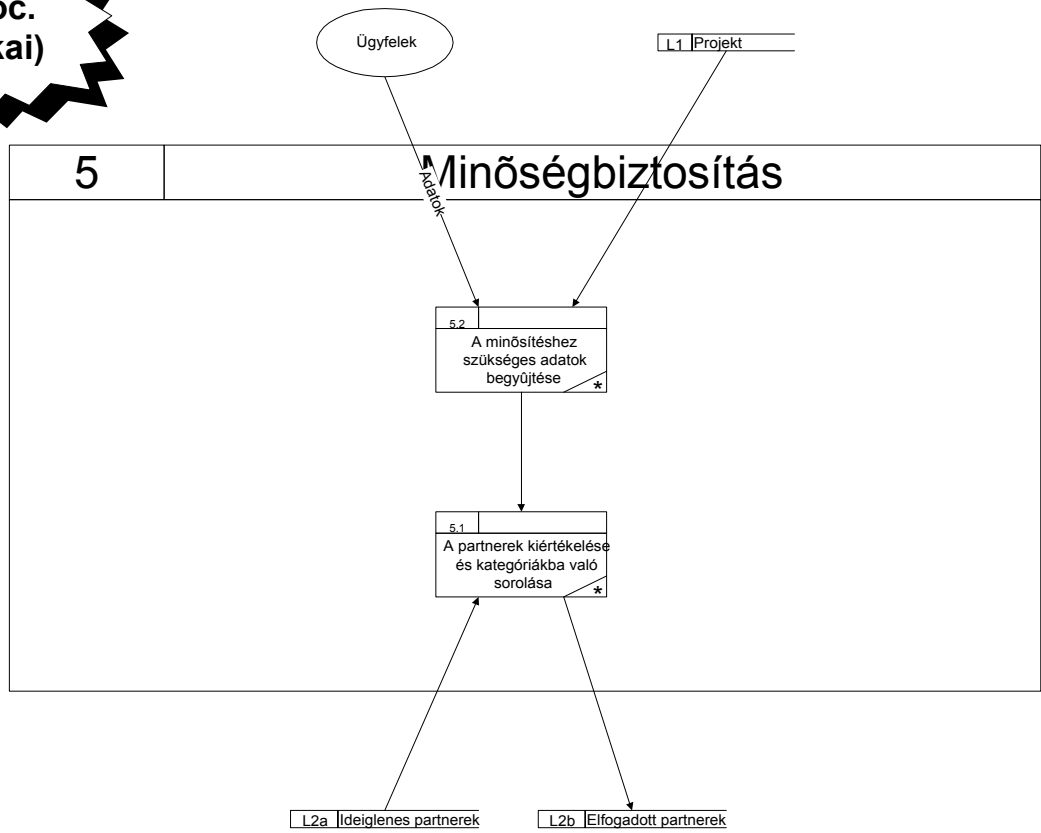
5.31. ábra

**2 DFD
4 Proc.
(logikai)**



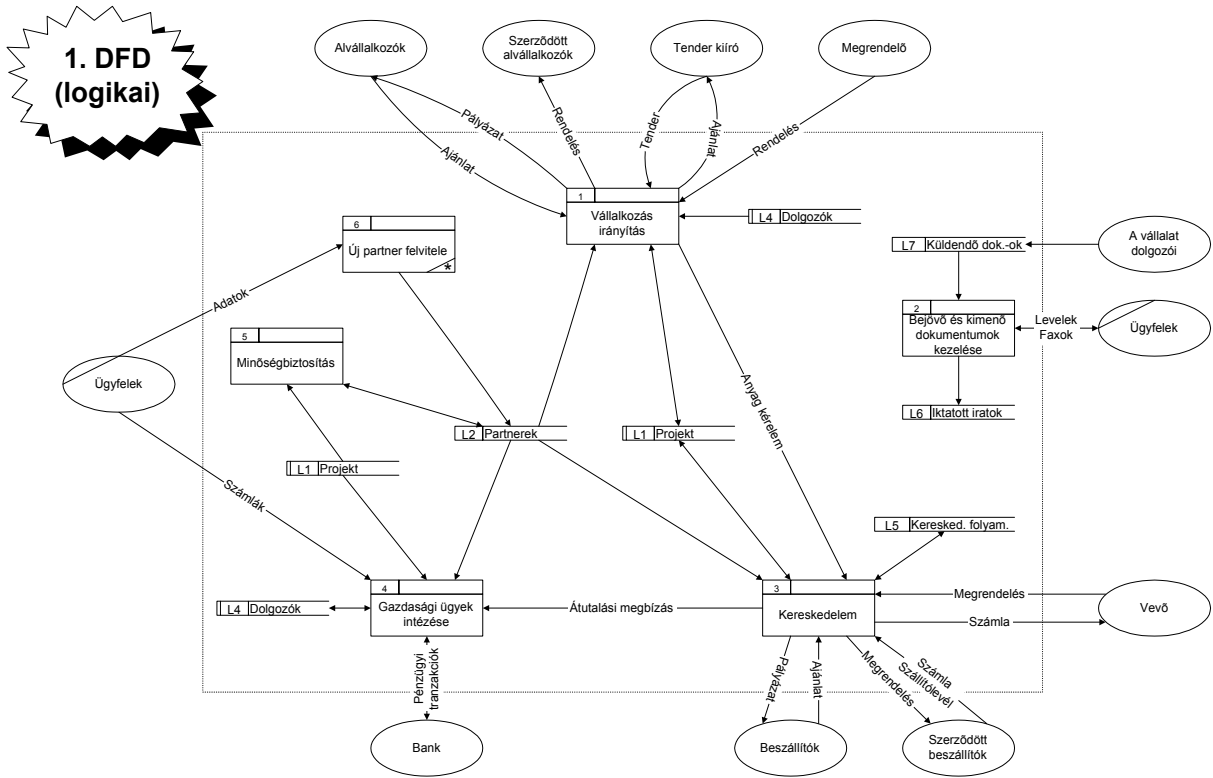
5.32. ábra

**2 DFD
5 Proc.
(logikai)**



5.33. ábra

A Top-level DFD-n megjelent az alsóbb szintekről kiiktatott *Új partner felvitele* folyamat. A felvitt új partnert azonban csak a *Minőségbiztosítás* folyamat minősítheti, ezzel betartva a cég által elnyert ISO 9001-es minősítés szabályait. A logikalizálás eredményeképpen a diagram letisztult és átláthatóbb lett (5.34. ábra).



5.34. ábra

6. Irodalom

- Keith Robinson - Graham Berrisford: *Object-oriented SSADM*. Prentice Hall, London, 1994
- Arató István - Schwarczenberger Istvánné Dr.: *Információs rendszerek szervezési módszertana*. ComputerBooks; Budapest, 1993
- Dr. Kovácsné Cohner Judit - Takács Tibor: *Ismerkedés az SSADM-mel*. ComputerBooks; Budapest, 1995
- Farkas A. - Kiss L.: *Az SSADM ismertetése*. KFKI, 1988
- Ed Downs - Peter Clue - Ian Coe: *Structured Systems Analysis and Design Method, Application and Context*. Prentice Hall, London, 1992
- MTA Információtechnológia Alapítvány: *SSADM Struktúrált rendszerelemzési és tervezési módszer*. Budapest, Miniszterelnöki Hivatal, 1993
- Sölvberg A. - Kung D. C.: *Information systems engineering. An introduction*. Springer-Verlag, Berlin, 1993
- Wataru Mayeda: *Alkalmazott gráfelmélet*. Műszaki könyvkiadó, Budapest 1976
- Sz. V. Jablonszkij - O. B. Lupanov: *Diszkrét matematika a számítástudományban*. Műszaki könyvkiadó, Budapest 1980
- Hajnal András - Hamburger Péter: *Halmazelmélet*. Tankönyvkiadó, Budapest 1983
- P. R. Halmos - L. E. Sigler: *Elemi Halmazelmélet*. Műszaki könyvkiadó, Budapest 1981
- Paul Gray: *Decision support and executive information systems*. Englewood Cliffs, NJ, 1994
- *Alaplap* számítógépes magazin, 1996. februári kiadás.
- Alb Péter, Ferenc Rudolf, Rajda Vilmos: *Információs rendszer elemzése és tervezése SSADM-mel és a DFD-k globális analízise*. OTDK dolgozat, Szeged, 1997