

# SSADM

Az SSADM (Structured System Analysis and Design Method) egy rendszerelemzési módszertan.

## A struktúrált módszertanok alapelvei:

**Elemzés felülről lefelé:** alrendszerre, funkciókra, folyamatokra bontás

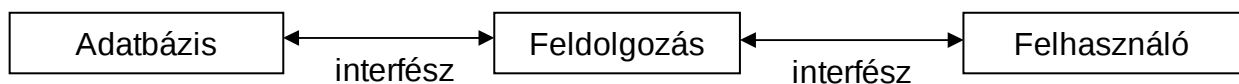
**Tervezés alulról felfelé:** hierarchikus építkezés alapelemekből, pontos, részletes terv.

**Logikai** (mi történik) és **fizikai** (hogyan történik) vizsgálatok szétválasztása. **Logikai:** a rendszer működésének belső logikája, viszonylag állandó. **Fizikai:** a tényleges megvalósítás adott eszközökön.

**Fokozatosság:** pl. egy új egyednél először csak a nevét és kapcsolatait határozzuk meg, az attribútumai csak később adjuk meg.

**Iterativitás:** amit egyszer már megterveztünk, az – akár többször is – módosításra szorulhat.

## Információs rendszerek komponensei:



**Adatok:** bemeneti (input), kimeneti (output), tárolt (adatbázis)

**Folyamatok (feldolgozások):** lekérdező, karbantartó. Részletezettség szerint: összetett folyamatok, elemi folyamatok, funkciók, eljárások

**Interfészek:** pl. karakteres, grafikus

## A rendszerszervezési munka menete

**Informatikai stratégia:** egy szervezeten belül több informatikai projekt lehet, ezek összehangolása a cég hosszú távú érdekeinek megfelelően.

**Megvalósíthatósági vizsgálat**

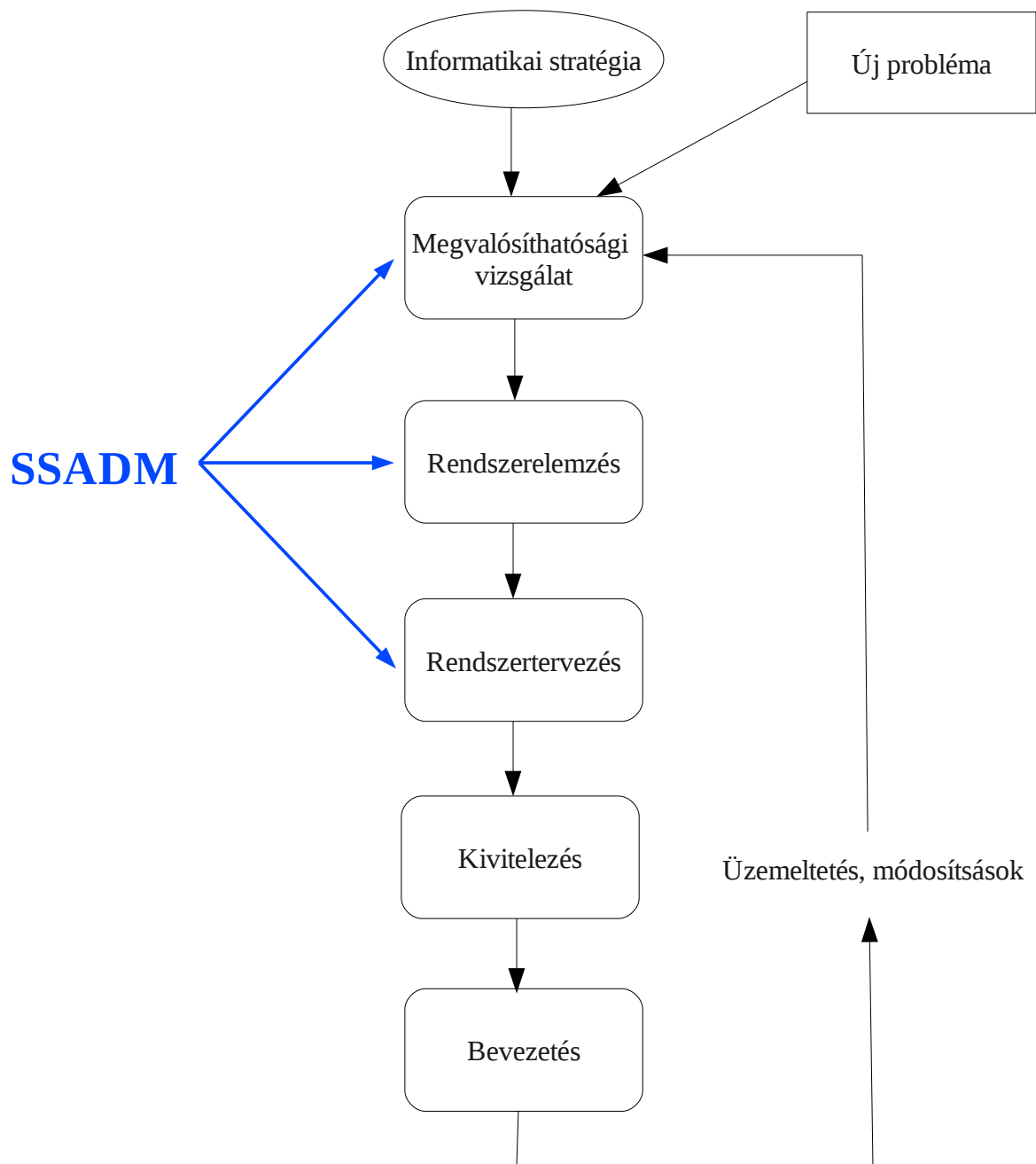
**Rendszerelemzés, rendszertervezés**

**Kivitelezés**

**Bevezetés**

**Üzemeltetés, módosítások**

Hosszabb idő, sokszori módosítás után a rendszer toldozottá válik, elavul, indokolttá válhat egy teljesen új rendszerrel való helyettesítése.



### **Az SSADM szerkezete:**

#### ***A rendszer három dimenziója, és a kapcsolódó technikák:***

**Adatok:** logikai adatmodell, relációs adatmodell, fizikai adattervezés

**Folyamatok:** adatfolyam modellezés, funkció meghatározás, I/O tervezés, dialógustervezés, menütervezés, logikai adatbázis folyamatok tervezése, fizikai folyamattervezés.

**Idő:** egyedtörténeti diagram, esemény-hatás diagram, logikai adatbázis folyamatok tervezése, fizikai folyamattervezés.

Az SSADM szerkezeti hierarchiájában vizsgáljuk a modulokat, a szakaszokat, az egyes lépéseket és a feladatokat.

### **Megvalósíthatóság-elemzés** (elhagyható)

0. Megvalósíthatóság eldöntése

### **Követelmény elemzés** (elemzési feladat)

1. Jelenlegi helyzet vizsgálata (adatok, folyamatok, követelmények)

2. Rendszerszervezési változat kiválasztása (alkalmatlan és drága változatok kiszűrése, az előnyös változatokat tarjuk csak meg)

### **Követelmény specifikáció** (elemzés és tervezés)

3. Követelmények meghatározása (EK-modell → top-down, normalizált logikai adatmodell → bottom-up )

- *Meglévő rendszer fizikai folyamatai* (az 1. szakaszban már elkészült)
- *Meglévő rendszer logikai folyamatai* (az 1. szakaszban már elkészült)
- *Tervezett rendszer logikai folyamatai*
- *Tervezett rendszer funkciói*

Eszköz: adatfolyam-modellezési technika.

↓

*feldolgozási specifikációk*: lekérdezési utak, karbantartást kiváltó események és hatásaik leírása.

Eszköz: eseményhatás-modellezési technika

### **Logikai rendszerspecifikáció** (tervezési feladat)

4. Rendszertechnikai változat kiválasztása: *Figyelembe veendő*: meglévő eszközök, hosszabb távú fejlesztési stratégia, kapacitás, költség, várható hatások. 3-6 változat kidolgozása javasolt, amely később 2-3 változatra szűkíthető.

5. Logikai rendszertervezés:

*Logikai adatmodell*: a korábbi szakaszokból örököljük.

*Feldolgozások* (ún. Jackson struktúrával írható le):

karbantartó (lekérdezést is tartalmazhat)

lekérdező

*Dialógusok tervezése*:

- dialógusszerkezet (Jackson struktúra)
- menüszerkezet

## **Fizikai rendszertervezés (tervezési feladat)**

### 6. Fizikai rendszertervezés:

Adattervek: logikai adatmodell átalakítása fizikai adattervvé.

Első közelítésű terv: egyedtípusokból relációsémák.

Finomított terv: idő-követelmények figyelembe vétele.

Feldolgozástervek: funkciók áttekintése, majd részletes specifikációk elkészítése.

Adattervezéstől független elemek

Fizikai adattervre épülő elemek

Interfész-tervek:

Adat ↔ feldolgozás interfészek

Felhasználói interfészek (képernyőtervek, nyomtatványok, listaformátumok) fizikai megtervezéséhez nem ad technikát az SSADM.

## **Az SSADM technikái:**

### **1. Diagramra épülő technikák:**

**Dokumentumáramlási diagram**

**Folyamatmodellezés (AFD)**

**Logikai adatmodellezés (Egyedmodell)**

**Egyed-esemény modellezés**

Elérési út modellezése

**I/O szerkezet meghatározása (Funkció meghat. része)**

**Dialógus tervezés**

Logikai feldolgozástervezés (lekérdezések és karbantartások tervezése)

### **2. Nem diagramra épülő technikák:**

**Követelménykatalógus**

Rendszerszervezési változatok kidolgozási módszere (rendszer határai, költség/haszon elemzés, hatások elemzése)

Rendszertехnikai változatok kidolgozási módszere (hardver-szoftver környezet, szervezetbeli és munkamódszerbeli változások, stb.))

**Funkció meghatározás**

**Relációs adatelemzés (normalizálás)**

Specifikáció prototipizálás (pl. gépen futó demó változat)

## **Követelménykatalógus:**

Készítése a projekt kezdetén indul el, és folyamatosan bővül.

**Funkcionális követelmények:** mit kell a rendszernek tudnia (nyilvántartandó adatok, lekérdezési igények, más rendszerekkel való kapcsolatok, stb.)

**Nem funkcionális követelmények:** hogyan, milyen minőségben kell tudnia a rendszernek (naprakésztség, válaszidők, tranzakciók száma, adatvédelem, stb.)

**Dokumentumáramlási diagram:**

A dokumentumáramlási ábra akkor hasznos, ha van egy jelenleg működő, főként kézi jellegű rendszer.

**Teendők:**

Soroljuk fel a főbb dokumentumokat illetve információ- áramlásokat.

Rajzoljuk meg a diagramot. **Csomópontok:** szervezeti egységek, **nyilak:** áramlás.

Jelöljük ki a rendszer határait.

**Folyamatmodellezés:**

**Kialakulásának története:** korábban szöveges leírást vagy folyamatábrát alkalmaztak.

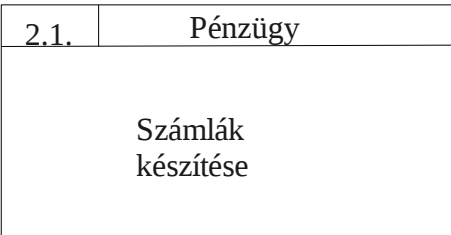

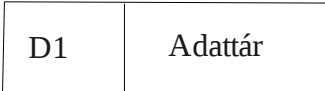
**Eszköze:** **AFD** = Adatfolyam diagram (= Adatáramlási diagram), angolul **DFD** = Data-flow diagram.

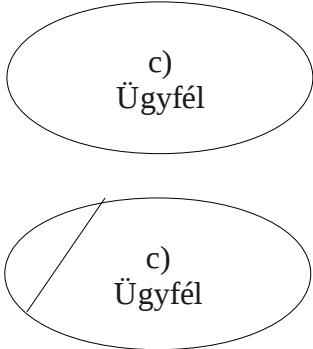
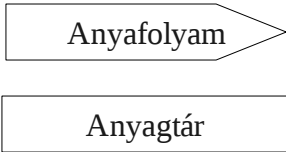
**Lényegében egy speciális gráf, ahol**

**Szögpontok:** folyamat, adattár, környezeti elem, anyagtároló

**Élek:** adatfolyam, anyagfolyam

**Az adatfolyam diagram elemei:**

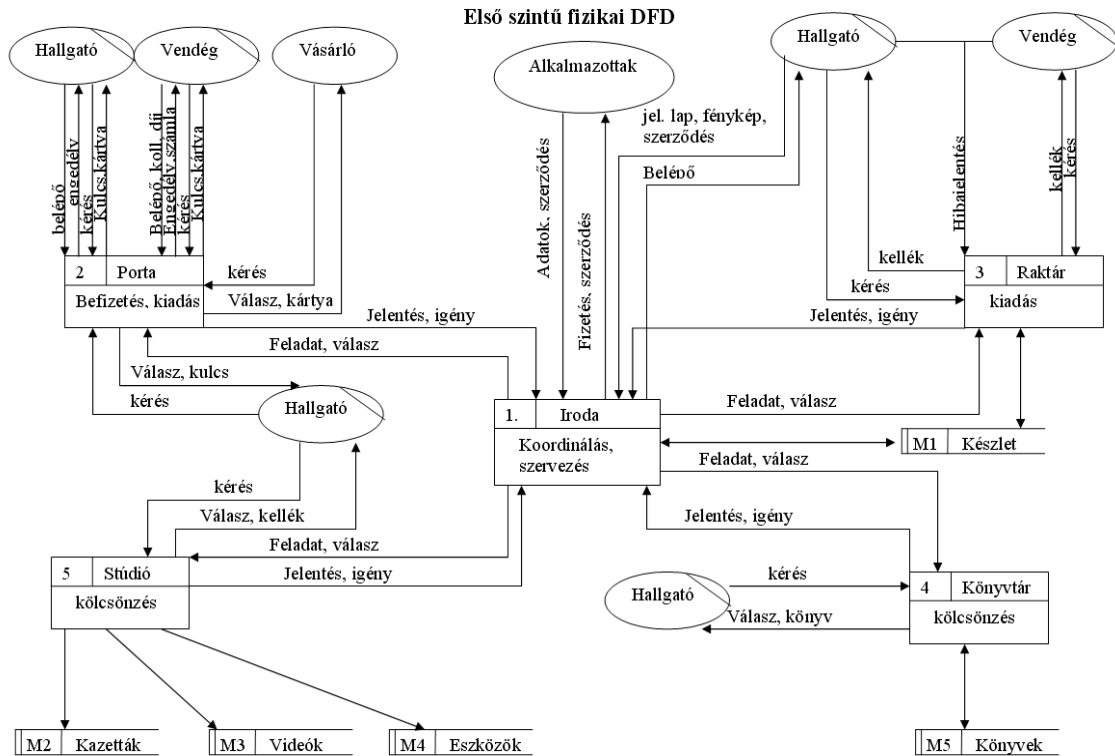
<p><b>Folyamat:</b> információ-átalakítás. <i>Tartalma:</i> Azonosító (sorszám), Szervezeti egység (pl. pénzügy) vagy beosztás (pl. raktáros) neve, Folyamat megnevezése. Az AFD döntési jelképet nem tartalmaz, a döntést is folyamatnak tekintjük.</p>	
<p><b>Adatfolyam.</b> <i>Tartalma:</i> Adat megnevezése (nyilvánvaló esetekben elhagyható)</p>	
<p><b>Adattár:</b> fájl, kartoték stb. <i>Típusai:</i> törzsadattár, átmeneti adattár. Karbantartást csak az adattár felé mutató nyíllal jelöljük (valójában az adatot előbb ki kell olvasni, majd módosítva visszaírni).</p>	

<p><i>Tartalma:</i> Azonosító (betű + szám). <i>Betűjelek:</i> M=manuális, D = digitális, L = logikai (logikai AFD-ben), T = temporális. Megnevezés. Áttekinthetőség érdekében több példányban is szerepelhet, ezt a bal szélén kettős vonal jelzi.</p>	<table border="1" data-bbox="1110 226 1442 315"> <tr> <td data-bbox="1110 226 1233 315">D1</td> <td data-bbox="1233 226 1442 315">Adattár</td> </tr> </table>	D1	Adattár
D1	Adattár		
<p><b>Környezeti elem:</b> a vizsgált rendszer határain kívül eső objektum (adatforrás vagy adatigénylő). Pl. partnervállalat, vállalat másik szervezeti egysége, adatszolgáltatást igénylő hivatal (pl. KSH). <i>Tartalma:</i> Megnevezés, Azonosító (szükség esetén). Áttekinthetőség érdekében több példányban is szerepelhet, ezt ferde vonal jelzi.</p>			
<p><b>Anyagfolyam.</b> Tartalma: Megnevezés. <b>Anyagtárolás.</b> Tartalma: Megnevezés.</p>			

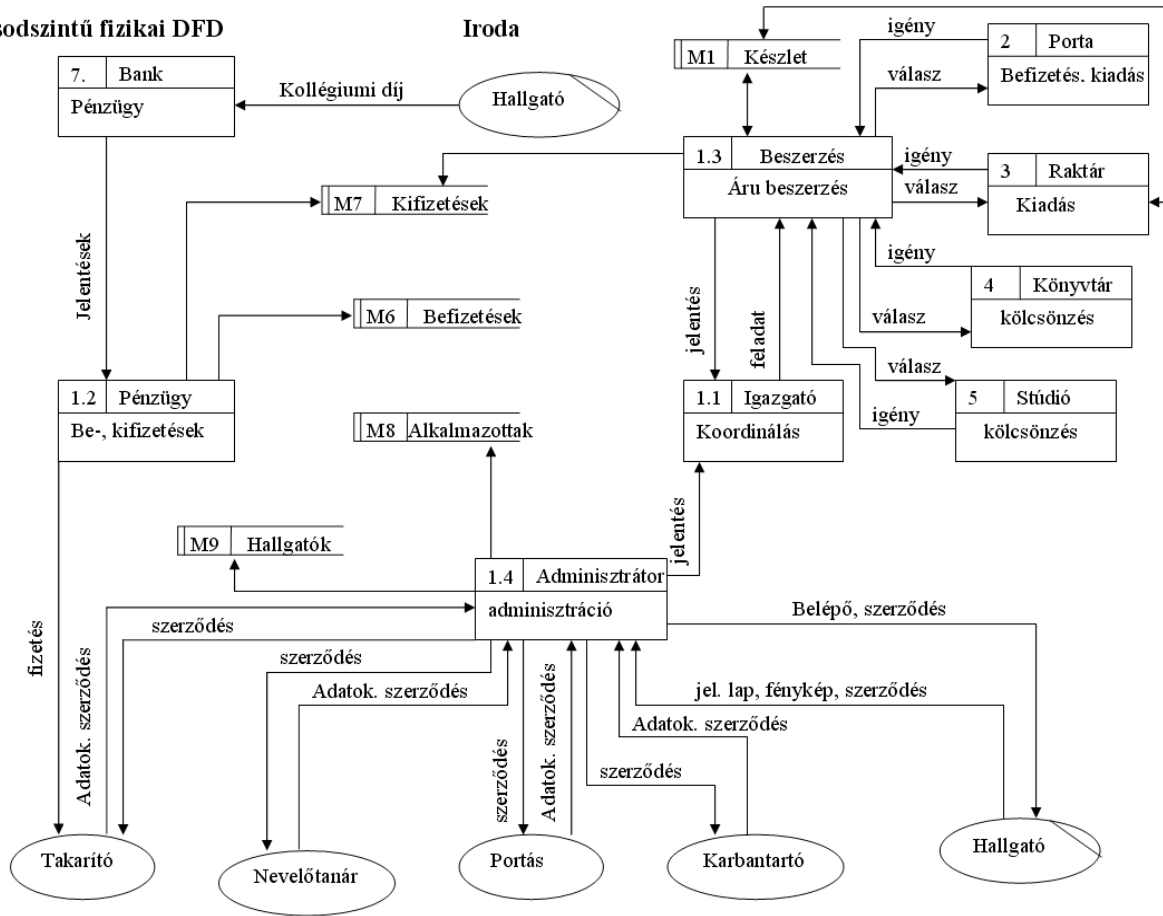
Megengedett kapcsolatok:

	<b>Környezeti elem</b>	<b>Folyamat</b>	<b>Adattár</b>	<b>Anyag tároló</b>
<b>Környezeti elem</b>	CSAK külső adat-ill. anyááramlás	Igen	Nem	Nem
<b>Folyamat</b>	Igen	Igen	Igen	Csak anyagfolyam
<b>Adattár</b>	Nem	Igen	Nem	Nem
<b>Anyag tároló</b>	Nem	CSAK anyagfolyam	Nem	Nem

Példa:

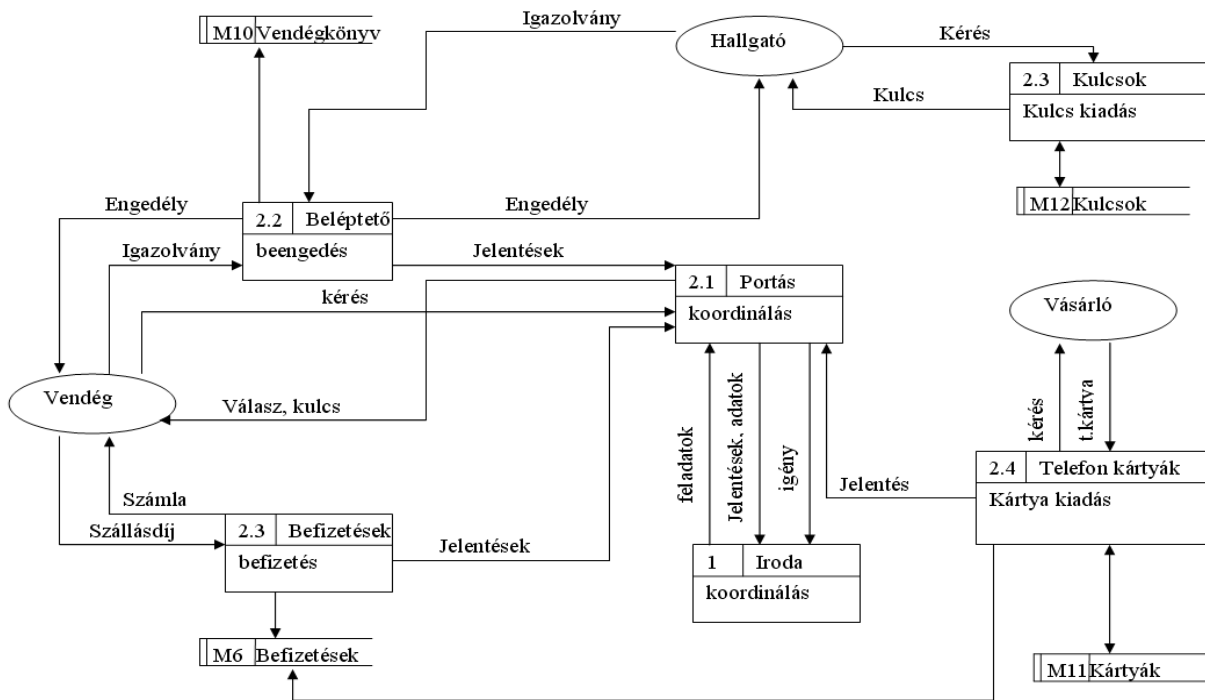


**Másodsztintű fizikai DFD**



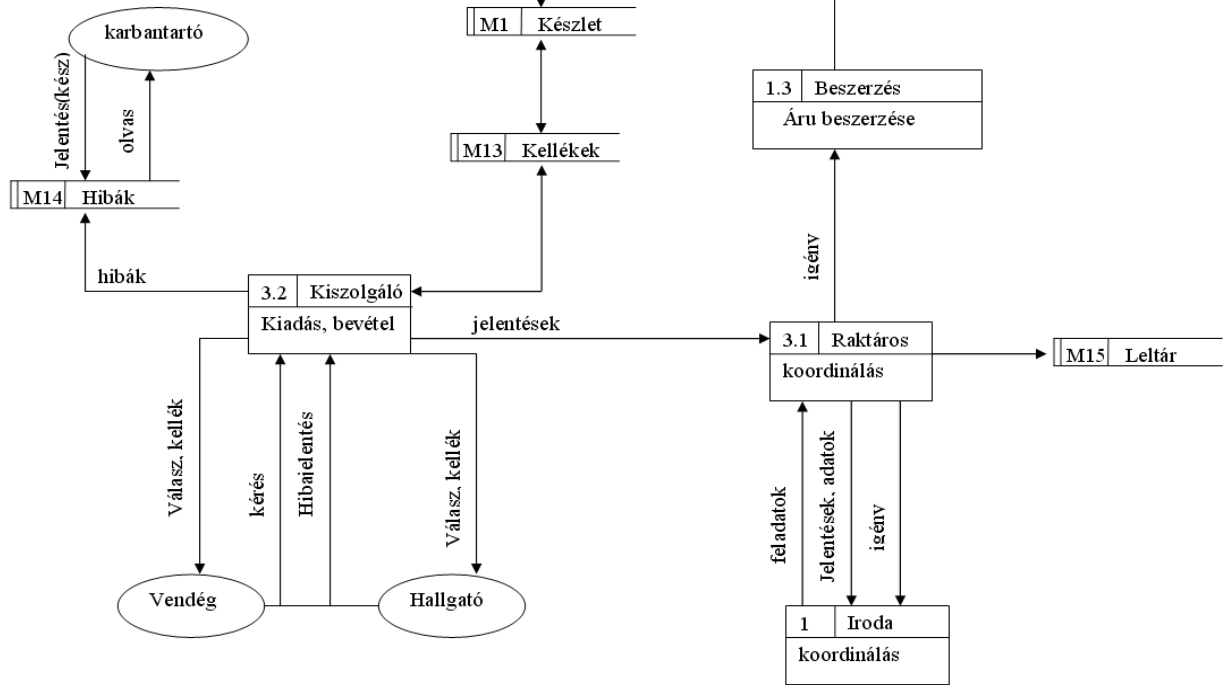
Másodszintű fizikai DFD

Porta



Másodszintű fizikai DFD

Raktár





### Logikai DFD:

Folyamatnál a szervezeti egység neve üresen maradhat.

Megnevezésekben kerüljük a fizikai megvalósításra való utalást (pl. "felvezetés kartonokra").

Anyagfolyam nem szerepelhet.

### A Fizikai → logikai DFD átalakítási szabályai:

A fizikai AFD-hierarchia legalsó szintjén kezdjük.

A felsőbb szinteket általában már nem, vagy alig kell átalakítani.

**Adattárak logikaivá alakítása.** Törölendők azok az (átmeneti) adattárak, amelyek csak az adott fizikai környezetben voltak szükségesek.

**Folyamatok logikaivá alakítása.** Törölhetők azok a folyamatok, amelyek pusztán az adatok újraszervezését végzik.

### DFD-k alkalmazása:

**Fizikai** és **logikai** szintű leírásra egyaránt alkalmas:

- Felméréskor (1. szakasz): először fizikai AFD, azután logikai AFD
- Különböző lehetséges megoldások közül való választáskor (2. szakasz): logikai AFD-k
- Új rendszer tervezésekor (3. szakasz): a tervezett rendszer logikai AFD-je
- Új rendszer fizikai tervezésekor (6. szakasz): a tervezett rendszer fizikai AFD-je

**Megjegyzés:** Szükség esetén az AFD egyes elemeinek szöveges leírása és mellékelendő.

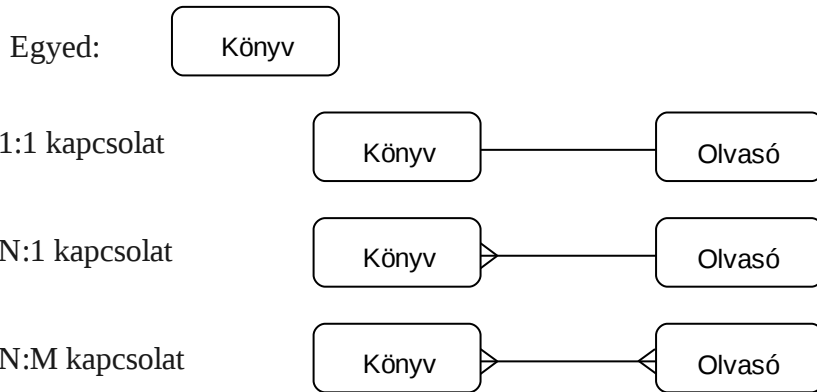
### Logikai adatmodellezés:

Az adatok belső logikai szerkezetének leírása, függetlenül a feldolgozási követelményektől, tárolóeszközök lehetőségeitől, stb.

**Egyed-kapcsolat modell** (EK), röviden **egyedmodell**

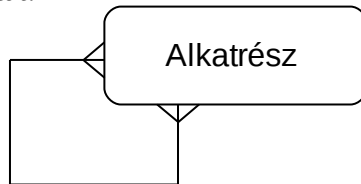
Később továbbléphetünk **egyed-tulajdonság-kapcsolat modellre** (ETK).

## Jelölésrendszer:





Nincs sokágú kapcsolat!  
Kapcsolatnak nem lehet attribútuma!

Rekurzív kapcsolatot:

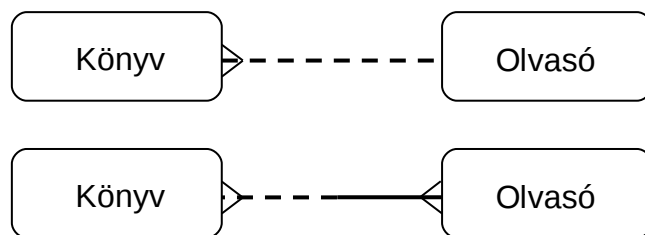


Kapcsolat jellege:

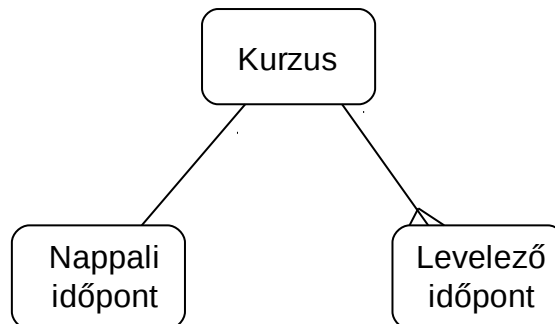
Egy egyed **kötelezően (teljesen)** vesz részt a kapcsolatban, ha minden egyedpéldány részt vesz legalább egy kapcsolatban  folytonos vonal.

Ellenkező esetben az egyed **esetlegesen (részlegesen)** vesz részt a kapcsolatban  szaggatott vonal.

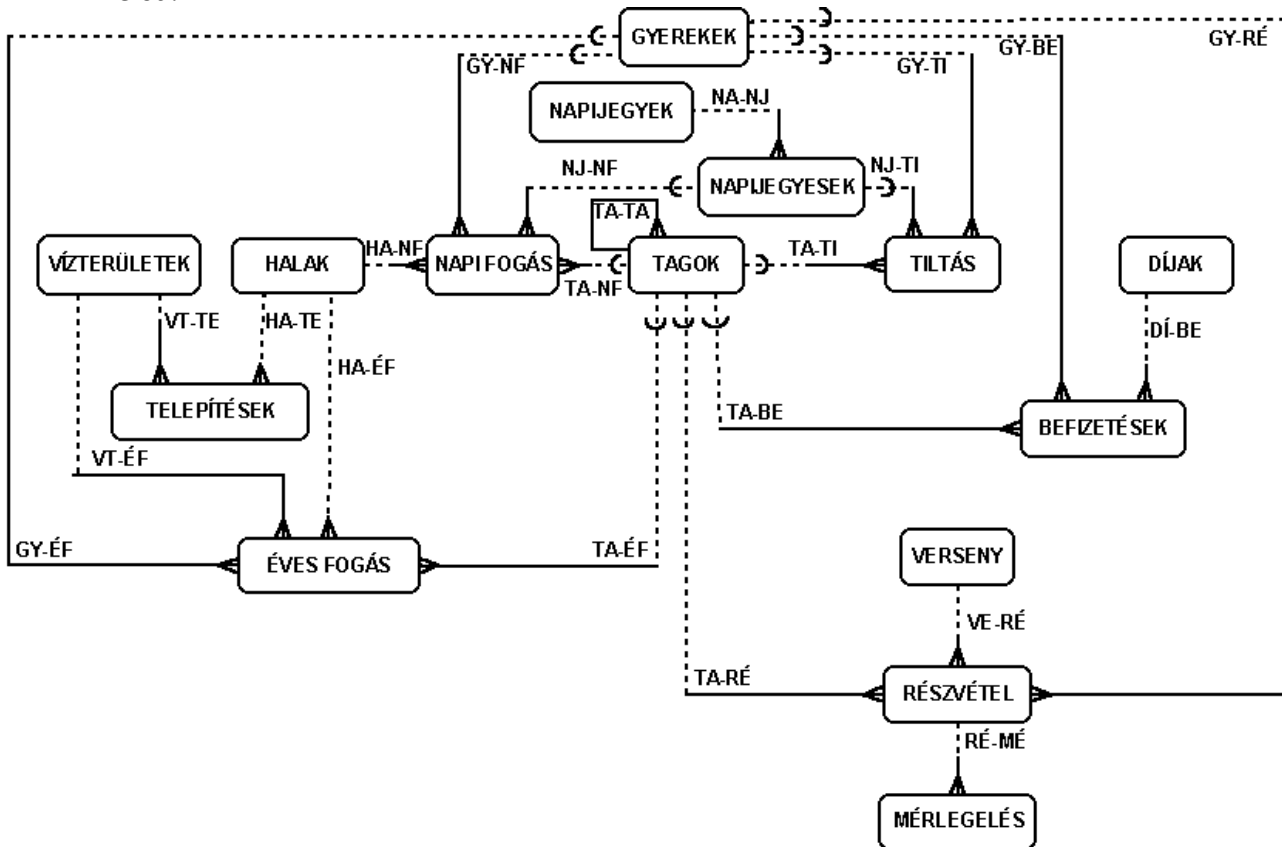
Ha a jelleggel nem akarunk foglalkozni, mindenütt folytonos vonal alkalmazható.



Kizáró kapcsolat:



Példa:



A logikai adatmodellezés gyakorlata:

**Általában minden kapcsolatot 1:N-re hozunk:**

N:M-nél új egyedet veszünk fel,

1:1-nél a két egyedet összeolvasztjuk (ha az célszerű)

**Elnevezések:**

Minden egyed különböző névvel szerepeljen.

Kapcsolat neve képezhető az egyedek neveiből, pl.

Könyv-Olvásó kapcsolatnál KÖ-OL.

Egyedmodellek alkalmazása:

Jelenlegi egyedmodell (1. szakasz)

Tervezett egyedmodell (3. szakasz)

Relációs adatelemzésnél (5. szakasz, lásd később)

**Megjegyzés:** Szükség esetén az egyedek (és kapcsolatok) szöveges leírása és mellékelendő.

### Funkciók meghatározása:

Az SSADM 3. szakaszában végzendő, nem diagram alapú technika.

**Funkció:** a felhasználó szempontjából egy egységet képező folyamat.

Általában a legalsó szintű AFD egy folyamatának felel meg, de kivételesen lehet folyamat része, vagy több folyamat együttese is (például ha közvetlen folyamat–folyamat adatfolyam van).

Az AFD rendszerszervezői, a funkció felhasználói szemléletű.

### Funkciók típusai:

#### ***Feldolgozás típusa szerint:***

lekérdezés,

karbantartás

#### ***Megvalósítás módja szerint:***

on-line,

off-line

mindkettő

#### ***A kezdeményező szerint:***

Felhasználói (környezeti elemből kiinduló esemény indítja el),

Rendszerfunkció (a funkció végrehajtása „belülről” indul, pl. meghatározott időnként ellenőrizni kell a kölcsönzési idők lejártát).

### A funkciók megállapítására szolgáló űrlap fontosabb rovatai:

*Funkció neve és azonosítója*

*Funkció típusa:* lásd fent

*Felhasználói szerepkörök:* mely felhasználó-csoportoknak lesz jogosultsága

*Funkcióleírás:* input, output, feldolgozás leírása

*Hibakezelés:* teljességében csak a fizikai tervezés során tölthető ki.

*AFD eljárás(ok)*

*I/O leírások:* a rendszer határait átlépő adatfolyamokat kell megadni.

*I/O szerkezetek:* I/O szerkezeti diagramok megadása.

*Követelménykatalógusra hivatkozás:* mely felhasználói igény kielégítésére kellett létrehozni a funkciót.

*Tömegszerűség:* időegység alatt hányszor kerül sor a funkció végrehajtására.

*Lekérdezések:* visszakeresési útra való hivatkozások.

**Közös feldolgozás:** ha a funkció olyan feldolgozást tartalmaz, amelyet a rendszerben máshol is alkalmazunk, akkor erre lehet itt hivatkozni.

**Dialógusnevek:** a funkcióhoz kapcsolódó dialógusokra hivatkozás.

**Szolgáltatás szintjére vonatkozó körülmények:** pl. a válaszdőre vonatkozóan: Leírás = „válaszdő”, Célérték = 3 sec, Tűrés = 10 sec.

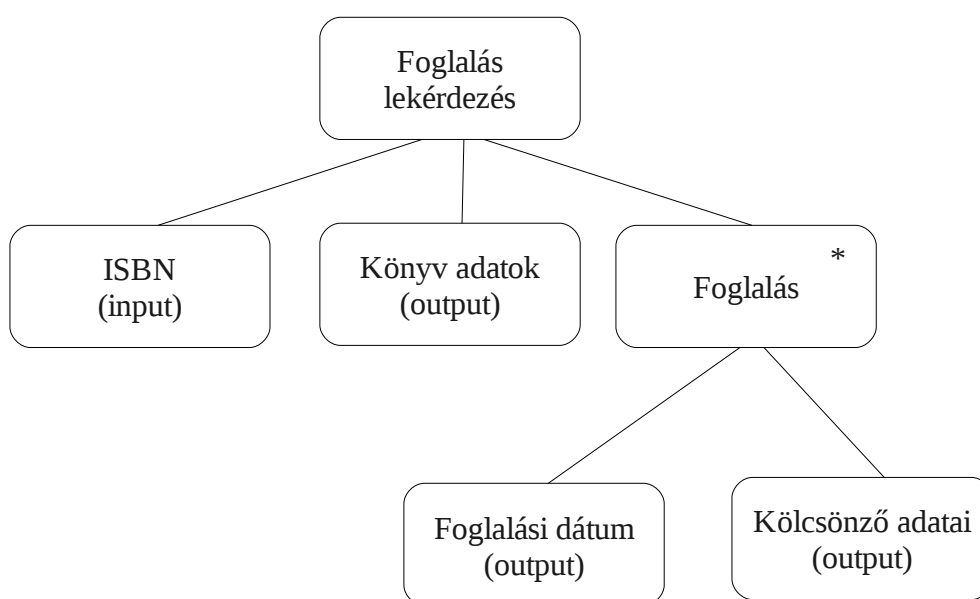
### I/O szerkezeti diagramok:

A funkció meghatározás része.

#### **Megadás Jackson-diagrammal:**

Téglalapok és összekötő vonalak

Ismétlődés jelzése csillaggal



### Relációs adatelemzés

**Logikai adatmodellezés:** felülről lefelé (top-down)

**Relációs adatelemzés:** alulról felfelé (bottom-up). A funkció meghatározás után, a fejlesztésnek abban a szakaszában végezzük, amikor a tulajdonságtípusokról már elég részletes információnk van.

**A két módszer eredményét összehasonlítjuk,** és addig finomítjuk az elemzéseket, amíg az eredmények meg nem egyeznek.

#### A relációs adatelemzés lépései:

1. **Tulajdonságtípusok** összegyűjtése.
2. **Relációsémák** felírása a tulajdonságtípusokból (esetleg I/O szerkezetek alapján). Normalizálatlanok lehetnek, még 1NF sem kötelező.
3. **Normalizálás** 3NF-ig (tovább menni csak ritkán kell).

4. **Konszolidálás:** azonos kulcsú sémák összevonása.
5. **Eredmény:** relációs adatbázis séma.

<p><b>Normalizálatlan</b></p> <p><b>Dolgozó</b>  <u>adószám</u>  név  lakcím  munkahelynév  munkahelycím  beosztás</p>	<p><b>1NF</b></p> <p><b>Dolgozó</b>  <u>adószám</u>  név  lakcím</p> <p><b>Dolgozik</b>  *<u>adószám</u>  <u>munkahelynév</u>  munkahelycím  beosztás</p>	<p><b>2NF</b></p> <p><b>Dolgozó</b>  <u>adószám</u>  név  lakcím</p> <p><b>Dolgozik</b>  *<u>adószám</u>  *<u>munkahelynév</u>  beosztás</p> <p><b>Munkahely</b>  <u>munkahelynév</u>  munkahelycím</p>
--	---	---

Logikai összehasonlítás adatmodellel:

**1. módszer:** A relációs adatelemzés eredményéből írunk fel egyedmodellt:

Relációnév → egyed, külső kulcs → kapcsolat.

A kapott egyedmodell összehasonlítása a logikai adatmodellezésnél kapott egyedmodellel.

**2. módszer:** A logikai adatmodellezés eredményéből írunk fel relációs modellt:

EK modell → relációs adatbázis séma

A kapott adatbázis séma összehasonlítása a relációs adatelemzés eredményével.

Fizikai adatbázis specifikáció:

Táblázat		
Attribútum 1	Adattípus, hossz	megjegyzés
...	...	...
Attribútum n	Adattípus, hossz	megjegyzés

## Egyed-esemény mátrix

### **Egyed-esemény mátrix:**

Függőlegesen az egyedek felsorolása.

Vízszintesen az események felsorolása.

### **A mátrix bejegyzései:**

L = létrehozás

O = olvasás

M = módosítás

T = törlés

(Más megközelítés: R=read, W=write, M=modify, D=delete)

<b>Események</b>	Kezelési napló bővítése	Kezelési terv bővítése	Új kezelési módszerek felvétele	Új tárgyi eszköz felvétele	Tárgyi eszköz adatok módosítása	Új megbízás rögzítése	Megbízás adatainak módosítása	Új orvos felvétele	Orvos adatainak módosítása	Orvos archiválása	Új beteg felvétele	Beteg adatainak módosítása	Új alkalmazott felvétele	Alkalmazott adatainak módosítása	Alkalmazott archiválása
<b>Egyedek</b>															
Alkalmazott													L	M	M
Árlista															
Beteg											L	M			
Betegség			L									T			
Kép															
Kezelési_Mód			L												
Kezelési_Napló	L		M												
Kezelési_Terv		L													
Könyvelés				L											
Külső vállalkozó															
Megbízás				T	L	M									
Orvos								L	M	M					
Rendelő										T					
Számla	M														
TE	M			L	M										

## Dialógus tervezés

**Cél:** képernyőtervek és menüszerkezetek elkészítése.

**Kiindulás:** a funkciómeghatározásnál a felhasználó által kezdeményezett on-line funkciókat tekintjük.

**Lépések:**

**Felhasználói szerepek:** hasonló tevékenységi körű és jogosultságú felhasználói csoportok behatárolása.

**Szerep-funkció mátrix készítése**

**Dialógusok megtervezése.**

Szerep-funkció mátrix:

Ügyállapot megváltoztatása					
Ügyirat kísérelő készítése					
Ügycsoportba sorolás					
Iktatás					
Szigonálás					
Ügyintéző				X	
Titkár					
Főoszt. vez.	X		X		X
Oszályvez.	X				
Póstabontó		X			

Menütervezés:

A szerep-funkció mátrixból indulunk ki és hierarchikus csoportosítást végzünk.

